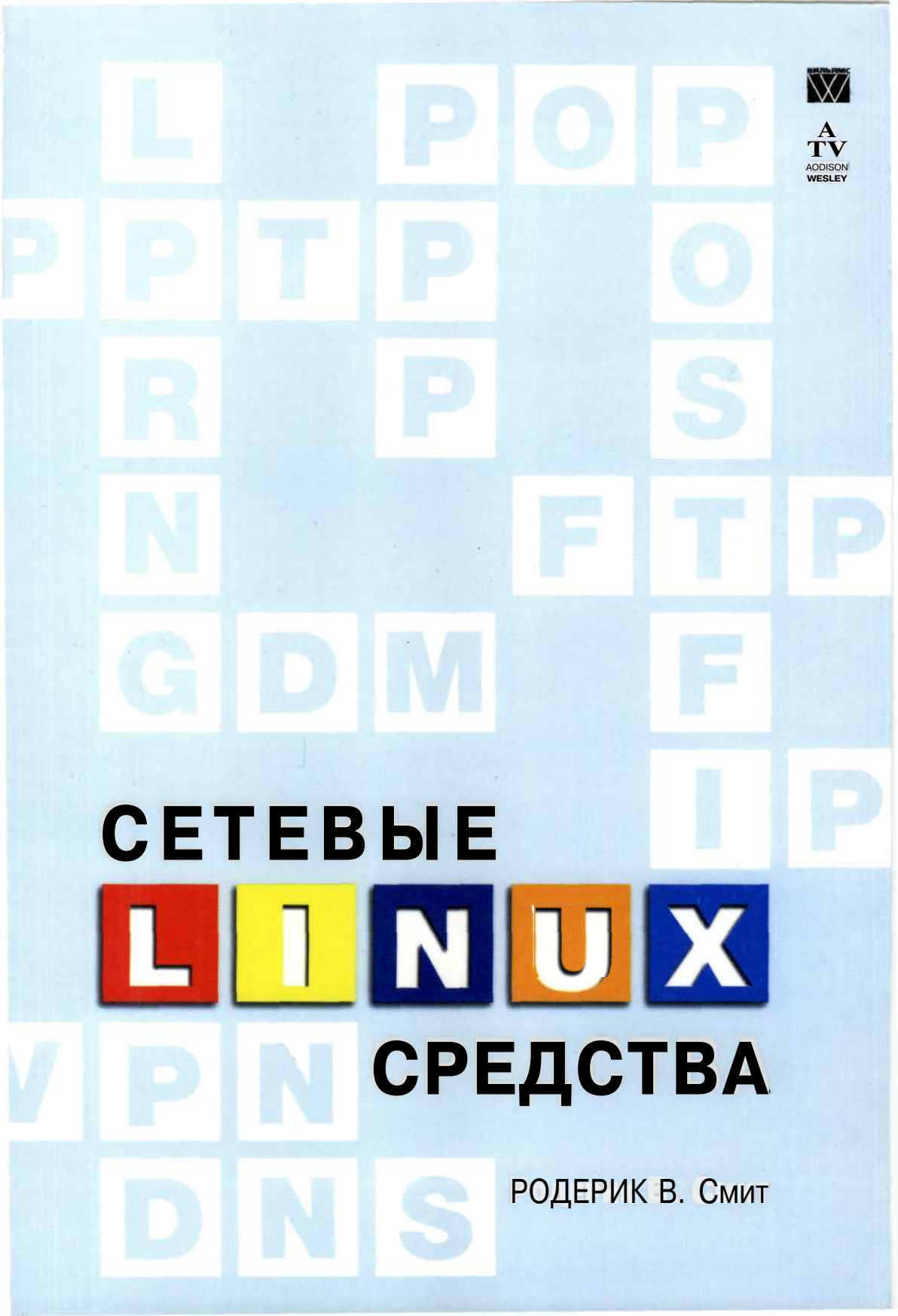

A
TV
ADDISON
WESLEY
СЕТЕВЫЕ
LINUX
СРЕДСТВА
РОДЕРИК В. Смит

Сетевые средства Linux

Advanced Linux Networking

Roderick W. Smith

A
TT
ADDISON-WESLEY

Boston • San Francisco • New York • Toronto • Montreal
London • Munich • Paris • Madrid • Capetown • Sydney
Tokyo • Singapore • Mexico City

Сетевые средства Linux

Родерик В. Смит

Москва • Санкт-Петербург • Киев
2003

ББК 32.973.26-018.2.75
С50
УДК 681.3.07

Издательский дом "Вильяме"

Зав. редакцией *С. Н. Тригуб*

Перевод с английского и редакция *В. В. Вейтмана*

По общим вопросам обращайтесь в Издательский дом "Вильяме" по адресу:
info@williamspublishing.com, http://www.williamspublishing.com

Смит, Родерик, В.

С50 Сетевые средства Linux. : Пер. с англ. — М. : Издательский дом "Вильяме", 2003. — 672 с. : ил. — Парал. тит. англ.

ISBN 5-8459-0426-9 (рус.)

В этой книге описаны принципы действия и область применения многих серверов, выполняющихся в системе Linux. Здесь рассматриваются DHCP-сервер, серверы Samba и NFS, серверы печати, NTP-сервер, средства удаленной регистрации и система X Window. Не забыты и средства, традиционно используемые для обеспечения работы Internet-служб: серверы DNS, SMTP, HTTP и FTP. Большое внимание уделено вопросам безопасности сети. В данной книге нашли отражения также средства удаленного администрирования — инструменты Linuxconf, Webmin и SWAT.

Данная книга несомненно окажется полезной как начинающим, так и опытным системным администраторам.

ББК 32.973.26-018.2.75

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства Addison-Wesley Publishing Company, Inc.

Authorized translation from the English language edition published by Pearson Education, Inc., Copyright © 2002 All rights reserved. No part of this book may be reproduced, stored in retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise without either the prior written permission o the Publisher.

Russian language edition published by Williams Publishing House according to the Agreement with R&I Enterprises International, Copyright © 2003

ISBN 5-8459-0426-9 (рус.)
ISBN 0-201-77423-2 (англ.)

© Издательский дом "Вильяме", 2003
© Pearson Education, Inc., 2002

Оглавление

Содержание

Отзывы о книге *Сетевые средства Linux*

Появилась прекрасная книга по Linux, осталось воспользоваться ею. Не упустите свой шанс.

Александр Стенцин, Help Net Security, `www.net-security.org`

Если вы стремитесь в полной мере использовать сетевые возможности Linux — эта книга для вас. Я настоятельно рекомендую прочитать ее.

Майкл Дж. Джордан, Linux Online

Выхода подобной книги давно ожидали читатели. Менее чем на 700 страницах автор смог изложить суть самых различных вопросов, связанных с работой Linux. Автор является высококвалифицированным специалистом в своей области и щедро делится своими знаниями с читателями.

Роджер Бертон, West, DiverseBooks . com

Посвящается жертвам
11 сентября 2001 года.
Надеюсь, что добро все же
одержит победу над злом.

Введение

Компьютерные сети изменили нашу жизнь. Они были почти незаметны в 1970-х и даже в 1980-х. Однако в начале 1990-х годов что-то произошло. Возможно, это было появление World Wide Web и графических Web-броузеров, благодаря которым Internet пришла во многие семьи. Возможно, число сетевых соединений превысило какой-то критический предел. Может быть, этот предел превысило количество сетевых программ. Как бы то ни было, сейчас о сетях знают все. А самое главное, что каждый знает о существовании Internet.

Internet объединяет миллионы компьютеров, на многих из которых выполняются серверы — программы, принимающие запросы от клиентов и обрабатывающие их. Благодаря тому что протоколы, на которых базируется Internet, допускают межплатформенное взаимодействие, в обмене данными могут участвовать клиенты и серверы, выполняющиеся на различных компьютерах и в разных операционных средах. В последние годы одной из самых популярных операционных систем стала Linux. Установленная на недорогом компьютере x86, система Linux обеспечивает эффективную работу серверов, поддерживающих узлы небольшого и среднего размеров. С увеличением производительности компьютеров появляется возможность выполнения в среде Linux серверов, обрабатывающих большие объемы данных. В результате от системного администратора часто требуется умение настраивать систему Linux и серверы, выполняющиеся в ее среде.

На каких же серверах следует остановить свой выбор? Существуют сотни, если не тысячи, серверных программ. В большинстве книг, посвященных системе Linux, основное внимание уделяется нескольким популярным серверам: HTTP-серверу (обычно это Apache), серверам удаленной регистрации, таким как Telnet и SSH, файловым серверам, примерами которых являются NFS и Samba, и некоторым другим типам серверов. В данной книге рассматриваются самые различные серверы. Разнообразие рассматриваемых вопросов не дает возможности подробно изучить работу и особенности настройки каждого из серверов, но все же приведенной информации достаточно, чтобы обеспечить выполнение соответствующих программ. Помимо наиболее популярных серверов, в настоящей книге также рассматриваются средства, которым обычно уделяется мало внимания, но которые, тем не менее, чрезвычайно важны для нормального функционирования сети. Так, например, здесь есть главы, посвященные DHCP-серверу, временному серверу и системе Kerberos. В данной книге не излагаются основы функционирования сетей. Считается, что читатель уже имеет представление о сетевых средствах и собирается повысить свою квалификацию.

Главы, в которых описываются сложные серверы, такие как Apache и Samba, не содержат исчерпывающего их описания. За общей информацией об этих инструментах следует рассмотрение их расширенных функций, ориентированное на администратора, имеющего некоторый опыт работы. Новичкам, прежде чем изучать эти главы, желательно прочитать книги, содержащие вводный курс администрирования данных серверов.

На кого рассчитана эта книга

Данная книга содержит расширенные сведения о сетевых средствах Linux и ориентирована на специалистов, которые уже работали с сетями и системой Linux. Первые главы содержат сведения о настройке низкоуровневых сетевых средств Linux. Чтобы материал этих глав был понятен, надо иметь общее представление о Linux, или, по крайней мере, о UNIX, и знать терминологию, применяющуюся при описании сетевых средств. Если

же вы незнакомы с системой Linux, вам имеет смысл прочитать вводные материалы, например, книгу Марселя Гагне (Marcel Gagne) *Linux System Administration: A User s Guide* (Addison-Wesley, 2002) или нашу с Вики Стенфилдом (Vicki Stanfield) книгу *Linux System Administration* (Sybex, 2001).

Если вы хотите больше узнать о таких серверах, как Apache и Samba, но не собираетесь покупать книгу, полностью посвященную одному продукту, либо если вы хотите получить сведения о небольших, но очень важных серверах, таких как `xntpd` и `xfs`, эта книга — для вас. В ней вы также найдете многочисленные практические советы, например, о том, как запустить сервер и завершить работу с ним, как создать резервную копию информации в сети, как ограничить сферу действий сервера поддеревом `chroot`, как построить брандмауэр, и т. д. Эти сведения позволят вам по-новому взглянуть на процесс администрирования вашей сети и, возможно, повысить производительность и надежность ее работы.

При написании данной книги я ориентировался на администраторов сетей небольшого и среднего размеров. В такой сети могут присутствовать компьютеры под управлением UNIX, Windows, MacOS и других операционных систем, и, конечно же, к ней подключена хотя бы одна машина Linux. В большинстве глав описываются общие принципы работы того или иного инструмента, а затем приводится информация о его использовании. Эту книгу можно использовать как справочное пособие. Если вы хотите иметь книгу, в которой описаны самые разнообразные сетевые средства Linux, то вы держите ее в руках.

Версии Linux

Одна из причин возникновения проблем при администрировании Linux состоит в том, что Linux нельзя рассматривать как единую операционную систему; это скорее набор систем, созданных на базе одного ядра. Разновидности Linux называют *версиями* или *дистрибутивными пакетами*. В состав дистрибутивного пакета входят ядро, инсталляционная программа, ориентированная на данную версию Linux, набор утилит, специальных инструментов, пользовательских программ и т. д. Кроме того, любой дистрибутивный пакет включает сценарии; некоторые из них предназначены для запуска серверов, другие — для настройки компонентов системы. В разных дистрибутивных пакетах содержатся разные версии ядра и различные наборы инструментальных средств. При инсталляции некоторых дистрибутивных пакетов часто устанавливаются специальные серверы, например почтовый сервер, в роли которого, в зависимости от версии Linux, выступают программы `sendmail`, Exim или Postfix. Характерные черты каждой версии Linux накладывают свой отпечаток на работу с ней и особенно на ее администрирование.

Во многих книгах игнорируются различия между разными версиями Linux. Основное внимание в них уделяется одному дистрибутивному пакету, а остальные упоминаются лишь время от времени. При написании этой книги я ставил перед собой цель более или менее подробно описать все популярные версии. В частности, здесь рассматриваются особенности Caldera OpenLinux 3.1, Debian GNU/Linux 2.2, Mandrake 8.1, Red Hat 7.2, Slackware 7.0, SuSE 7.3 и TurboLinux 7.0. У меня не было возможности детально рассмотреть каждую систему, я старался привлечь внимание читателя к главным различиям между ними. Так, например, в книге вы найдете сведения о том, какие средства используются в каждой версии Linux для запуска серверов, какие программы обеспечивают FTP-взаимодействие, и т. д. В некоторых главах рассматриваются различные программы, реализующие конкретный тип сервера. Сделано это для того, чтобы вы могли составить

представление о различиях в конфигурации, устанавливаемой по умолчанию в разных дистрибутивных пакетах.

Структура книги

Книга состоит из четырех частей, в каждой из которых содержится от четырех до тринадцати глав. Серверы, использующиеся для обслуживания локальных пользователей, и серверы, применяемые для организации взаимодействия по Internet, описаны в разных частях. Однако следует заметить, что некоторые серверы могут применяться в различных целях. Ниже кратко описано содержание каждой части.

Часть I. Эта часть короче других; она содержит лишь четыре главы. В ней описана настройка ядра системы, приведены общие сведения о конфигурации средств TCPJIP, стеках протоколов и сценариях запуска.

Часть II. В этой части рассматриваются серверы, к которым, вероятнее всего, будут обращаться лишь компьютеры вашей локальной сети. В ней обсуждаются сервер DHCP, система Kerberos, серверы Samba и NFS, организация печати с помощью LPD, временные серверы, почтовые серверы POP и IMAP, серверы новостей, серверы удаленной регистрации, система X Window и сервер VNC, сервер шрифтов, сервер удаленного администрирования и средства резервного копирования.

Часть III. Эта часть посвящена рассмотрению серверов, используемых для работы в Internet. В ней представлены сведения о серверах DNS, почтовых серверах, поддерживающих протокол SMTP, Web- и FTP-серверах.

Часть IV. В этой части обсуждаются вопросы сетевой безопасности. Здесь рассматриваются общие вопросы защиты, средства для создания поддерева `chroot`, настройка специальных функций маршрутизатора, создание брандмауэров с помощью `iptables`, средства NAT и настройка VPN.

Соглашения, принятые в книге

Чтобы упростить восприятие излагаемого материала, в книге приняты следующие соглашения.

Основной текст отображается обычным пропорциональным шрифтом.

Курсивом представлены термины, встречающиеся впервые. Кроме того, курсив используется для отображения текстовых описаний, заменяющих значения опций, поля записей и другие фрагменты кода.

Моноширинным `шрифтом` выделены имена файлов и узлов сети, фрагменты программного кода, содержимое конфигурационных файлов, команды, введенные с клавиатуры, и текст, выводимый на экран при выполнении программы.

Если в тексте встречается команда, вводимая с клавиатуры, в начале строки отображается приглашение для ввода. Символ # свидетельствует о том, что команду задает пользователь `root` (иногда встречаются исключения из данного правила). Если в роли приглашения для ввода выступает символ $, это означает, что команду вводит обычный пользователь. Некоторые команды могут занимать несколько строк. Признаком того, что продолжение команды находится на следующей строке, является символ \. Вы можете вводить команду точно так, как она приведена в книге, либо отказаться от использования обратной косой черты и задавать команду в одной строке.

В книге также встречаются специальные фрагменты текста, занимающие один или несколько абзацев. В них приводится информация, имеющая лишь косвенное отношение

к излагаемому материалу, либо, напротив, замечания, на которые следует обратить особое внимание. Эти фрагменты выделяются следующим образом.

Так оформляются сведения, которые не имеют непосредственного отношения к вопросам, рассматриваемым в тексте, но могут быть полезны для читателя. Это может быть, например, информация об особенностях работы ранних версий программ.

СОВЕТ

Данные, выделенные таким образом, помогут вам решить задачу неочевидным способом. Здесь может быть указана, например, ссылка на программный продукт, редко упоминаемый в других источниках.

ВНИМАНИЕ

Так оформляется предупреждение об опасности, связанной с теми или иными действиями. Это может быть, например, информация о программах, которые при некорректном использовании способны повредить систему, рекомендации воздержаться от действий, не соответствующих политике провайдера, или сведения об особенностях конфигурации, которые могут быть использованы для незаконного проникновения в систему.

Врезка

Врезка во многом похожа на фрагмент "На заметку", но объем ее значительно больше, она занимает как минимум два абзаца. В ней приводятся сведения, которые трудно включить естественным образом в текст раздела, но которые, тем не менее, интересны, связаны с текущим материалом и могут быть важны для читателя.

При обсуждении вопросов сетевого взаимодействия часто бывает необходимо указать IP-адреса компьютеров. В большинстве случаев я использую адреса, выделенные для внутренних сетей (192.168.0.0-192.168.255.255, 172.16.0.0-172.31.255.255 и 10.0.0.0-10.255.255.255). *Я* поступаю так для того, чтобы читатель, неудачно повторивший пример из книги, не мог нанести вред реальному компьютеру, подключенному к Internet.

Контактная информация

Если во время чтения книги у вас возникнут вопросы, вы можете обратиться ко мне по адресу `rodsmith@rodsbooks.com`. Я также поддерживаю Web-страницу, посвященную данной книге, которая находится по адресу `http://www.rodsbooks.com/adv-net/`.

Благодарности

Я хочу выразить благодарность редактору Кэрен Геттмен (Karen Gettman) за кропотливую работу, выполненную при подготовке данной книги к печати. В этом ей помогала координатор проекта Эмили Фрей (Emily Frey), которая сделала ряд ценных замечаний относительно текста книги. Ни одна техническая книга не может увидеть свет без помощи консультантов, которые помогают убедиться, что при изложении материала автор не погрешил против истины. Консультантами при подготовке этой книги к печати были

Кэрел Белоун (Karel Baloun), Эми Фонг (Amy Fong), Говард Ли Хекнесс (Howard Lee Harkness), Гарольд Хоук (Harold Hauck), Эрик Х. Херрин (Eric H. Herrin II), Дэвид Кинг (David King), Роб Колстэд (Rob Kolstad), Мэтью Миллер (Matthew Miller), Айэн Редферн (Ian Redfern) и Элекси Зинин (Alexy Zinin). После их работы над текстом в книге не могло остаться ни одной ошибки, если же такая будет обнаружена, то она, несомненно, была допущена лично мною. Я хочу также поблагодарить Дэвида Кинга (David King) за участие в многочисленных и плодотворных дискуссиях о сетевых средствах Linux. И, наконец, я выражаю благодарность моему агенту Нейлу Селкинду (Neil Salkind) из Studio B и Майклу Слоутеру (Michael Slaughter) из Addison-Wesley, стараниями которых эта книга попала в руки читателей.

ЧАСТЬ I

Низкоуровневая конфигурация системы

Глава 1

Настройка сетевых средств ядра

"Все дороги ведут в Рим" — гласит пословица. Нечто подобное можно сказать и о сетевых средствах Linux; в этом случае в роли Рима выступает ядро операционной системы. Рано или поздно весь сетевой трафик будет обработан ядром. Различные компьютеры и разные сети отличаются друг от друга, поэтому для ядра Linux предусмотрен ряд опций, изменяя значения которых вы можете оптимизировать систему с учетом задач, которые ей придется выполнять. Установку некоторых опций можно производить в процессе загрузки системы, передавая ядру необходимые параметры, либо впоследствии, после окончания загрузки. Подобные опции будут рассмотрены в последующих главах. Существуют также характеристики, для изменения которых надо перекомпилировать ядро системы.

В данной главе описаны опции, которые задают конфигурацию ядра. Сначала здесь рассматривается процедура настройки ядра. Затем обсуждаются опции, которые определяют свойства TCP/IP и других сетевых протоколов, а также сетевых фильтров. Далее речь пойдет о драйверах Linux, предназначенных для поддержки различных сетевых устройств. В конце главы кратко описывается процесс компиляции ядра системы.

В этой главе компиляция ядра рассматривается лишь в общих чертах, основное внимание уделяется опциям, определяющим характеристики сетевых средств системы. Если вы хотите получить более подробную информацию о конфигурации ядра системы, обратитесь к документу *Linux Kernel HOWTO* (`http://www.linuxdoc.org/HOWTO/Kernel-HOWTO.html`) либо к соответствующим книгам, представляющим собой введение в операционную систему Linux.

Конфигурация ядра

Для того чтобы установить опции, определяющие процесс компиляции ядра, необходимо иметь в наличии исходный код ядра. Исходный код входит в состав всех дистрибутивных пакетов, но при установке системы можно либо разрешить, либо запретить копирование исходного кода на жесткий диск компьютера. Следует заметить, что в некоторых случаях исходный код, поставляемый в составе дистрибутивного пакета, может быть из-

менен по сравнению со стандартным кодом ядра (так, например, в состав кода могут быть включены специальные драйверы). Целесообразно вначале инсталлировать стандартное ядро, а затем, по мере необходимости, установить дополнительные модули (не исключено, что для выполнения ваших задач никакие дополнения не потребуются). Список основных узлов, содержащих архивы Linux, находится по адресу `http://www.kernel.org`. В частности, там вы найдете ссылку на `ftp://sunsite.unc.edu` и адреса других узлов, содержащих последние варианты исходного кода ядра Linux. (Конечно, вы можете работать с исходным кодом ядра, который входит в состав дистрибутивного пакета, но, как было сказано выше, в нем могут быть установлены дополнительные модули. Если в процессе работы возникнут проблемы, то устранить их будет легче, если у вас инсталлировано стандартное ядро.)

Номер версии ядра системы состоит из трех чисел, разделенных точками. Если второе число четное (например, 2.4.17), то ядро называется *стабильным,* или *рабочим.* Нечетное второе число в номере версии (например, 2.5.2) указывает на то, что ядро находится *в процессе разработки.* Стабильное ядро обеспечивает более высокую надежность. Используя ядро, находящееся в процессе разработки, вы получаете возможность ознакомиться с новыми техническими решениями. Чаще всего в ядре с нечетным вторым числом номера версии используются новые драйверы, реализованы новые варианты интерфейса или применяются другие подобные новшества. Устанавливая систему для практического использования, желательно использовать ядро с четным вторым числом номера версии. Исключением является ситуация, когда необходимый вам драйвер присутствует только в версии с нечетным вторым числом. В этом случае можно также использовать *обратный перенос* (back-port) драйвера в одну из предыдущих стабильных версий.

Обычно исходный код ядра содержится в каталоге **`/usr/src/linux`** либо в одном из подкаталогов `/usr/src` (при этом в имени каталога присутствует номер версии ядра, например **`/usr/src/linux-2.4.17`**). В последнем случае желательно создать ссылку `/usr/src/linux`, указывающую на каталог с исходным кодом ядра. Если вы поступите так, то обеспечите нормальную работу программ, которые предполагают, что исходный код ядра содержится в каталоге `/usr/src/linux`. Таким образом, удобно работать с несколькими версиями исходного кода ядра, а если надо перейти от одной версии к другой, достаточно лишь изменить символьную ссылку.

Разархивировав исходный код ядра в каталог `/usr/src/linux`, надо сделать это каталог рабочим в используемой вами оболочке. После этого можно задать одну из описанных ниже команд конфигурирования ядра.

- make `config`. Данное средство конфигурирования является базовым. При этом у вас поочередно будут запрашиваться значения опций ядра. Отвечать на вопросы утомительно и при этом легко допустить ошибку. В случае ошибки придется начать всю процедуру сначала. Данная команда в настоящее время используется крайне редко.

- make menuconfig. Это средство конфигурирования предоставляет меню, позволяющее просматривать опции и задавать новые значения. Меню отображается в алфавитно-цифровом режиме. В этом случае изменить придется только те оп-

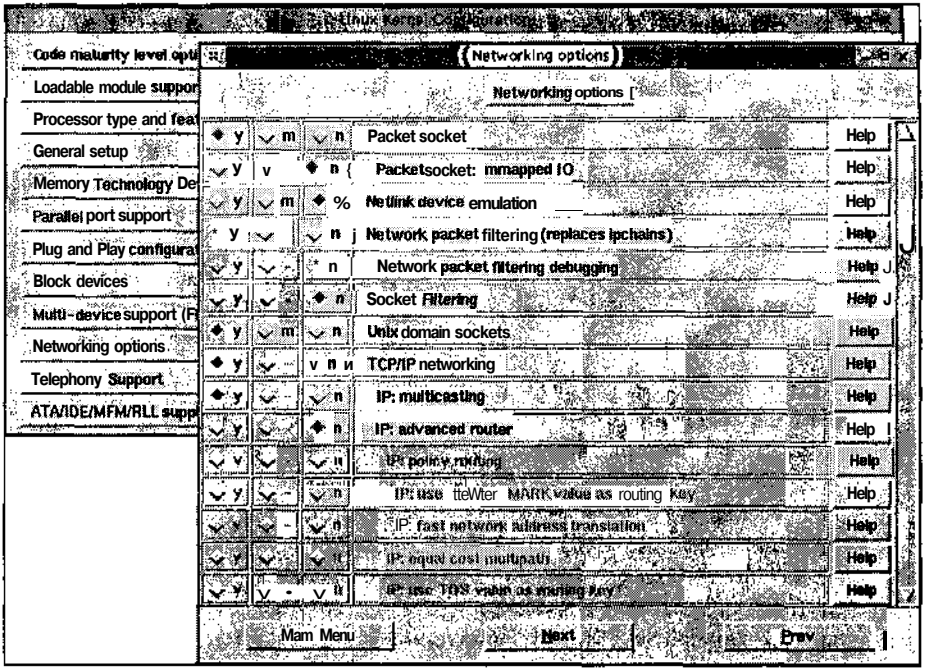


Рис. 1.1. Опции ядра Linux объединяются в категории и подкатегории, для каждой из которых предусмотрено отдельное меню

ции, значения которых не устраивают вас. При работе в текстовом режиме данное средство применяется чаще других.

- make xconfig. Данный способ установки конфигурации аналогичен make menuconfig, за исключением того, что меню отображается средствами графического интерфейса. В этом случае выбор опций и установку их значений можно выполнять с помощью мыши. Это средство установки конфигурации применяется при работе в среде X Window (X Window иногда называют X).

Все три способа позволяют работать с одними и теми же опциями. Опции объединены в несколько категорий; некоторые из категорий содержат подкатегории. Если вы используете make menuconfig или make xconfig, то для каждой категории отображается отдельное меню (пример работы с окном, отображаемым по команде make xconfig, показан на рис. 1.1). При настройке сетевых средств в основном используются категории Networking Options и Network Device Support, которые подробно рассматриваются в двух последующих разделах.

Для большинства опций предусмотрены переключатели. Примерами таких переключателей могут служить Y, M и N, показанные на рис. 1.1. Y и N указывают на присутствие или отсутствие опции в составе ядра, а M (сокращение от modular compilation — модульная компиляция) указывает на то, что соответствующие средства должны быть скомпилированы как отдельный модуль, которые можно загружать и выгружать независимо от других компонентов ядра. Более подробно о настройке опций рассказывается ниже.

Данная глава посвящена опциям версии 2.4.x ядра Linux, в частности, материал главы ориентирован на ядро 2.4.17. Опции, относящиеся к сетевым средствам, модифицировались раньше и, по-видимому, будут изменяться и в будущем. В версиях 2.2.x ядра опции в основном совпадают; различаются они лишь в деталях. В состав разрабатываемого ядра 2.5.x включено инструментальное средство CML2, предназначенное для настройки. Дополнительную информацию об этом инструменте можно получить по адресу `http://tuxedo.org/~esr/cml2/`.

Поддержка сетевых протоколов

Меню Networking Options содержит опции, влияющие на работу сетевых протоколов. Вы можете включить или исключить средства поддержки стека протоколов либо отдельных протоколов (в основном данные опции касаются семейства протоколов TCP/IP). Опции из этого меню позволяют также оптимизировать ядро для выполнения конкретных функций, например маршрутизации или фильтрации пакетов.

Опции для работы с пакетами и гнездами

Низкоуровневые сетевые средства Linux позволяют программам передавать и принимать фрагменты данных, называемые *пакетами,* посредством специальных структур, которые называются *гнездами* (socket). В большинстве случаев обмен данными через гнездо осуществляется по тому же принципу, что и обмен данными с файлом. Стек сетевых протоколов обеспечивает передачу информации по адресу назначения, где происходит ее интерпретация.

В некоторых случаях желательно и даже необходимо изменить принцип обработки данных; иногда приходится расширять стандартный набор операций над пакетами. Сделать это позволяют специальные опции, рассмотрению наиболее важных из них посвящены разделы данной главы. Некоторые из опций кратко описаны ниже.

- Packet Socket. Эта опция позволяет приложениям непосредственно обращаться к требуемому протоколу, минуя некоторые уровни стека протоколов. Для большинства программ такая возможность не нужна; ее используют лишь инструментальные средства сетевой диагностики и Специальные утилиты, действующие на нижнем уровне. В качестве примера подобных программ можно привести утилиту `tcpdump`, которая выводит информацию о пакетах TCP и IP. Данная опция не обязательна. Она несколько увеличивает размер ядра и дает возможность злоумышленникам воспользоваться утилитами сетевой диагностики. С другой стороны, отключив данную опцию, вы не сможете воспользоваться целым рядом утилит.

- Packet Socket: Mapped IO. Если данная подопция Packet Socket включена, производительность инструментальных средств, использующих низкоуровневые соединения, повышается.

- Unix Domain Sockets. Некоторые важные программы Linux используют сетевые протоколы для обмена данными даже в том случае, если они выполняются на одном и том же компьютере. В качестве примеров можно привести средство протоколирования `syslogd` и программы, выполняющиеся в среде X Window (X-программы используют сетевой протокол для взаимодействия с X-сервером, выполняющим

отображение данных). Опция Unix Domain Sockets допускает взаимодействие в пределах одной системы даже в тех случаях, когда на компьютере не установлено сетевое оборудование. Даже если средства поддержки сетевого обмена присутствуют, опция Unix Domain Sockets обеспечивает более высокую скорость обмена по сравнению с обычными TCP-гнездами. Обычно данная опция устанавливается; без нее обходятся лишь системы, предназначенные для выполнения на специализированных устройствах.

По умолчанию все три указанные опции устанавливаются. При необходимости вы можете запретить Packet Socket.

Опции сетевой фильтрации

Опции сетевой фильтрации блокируют или преобразуют пакеты, поступающие на компьютер или покидающие его. Данные опции используются при создании брандмауэров и выполнении IP-маскировки (подробно эти вопросы будут обсуждаться в главе 25). Брандмауэры блокируют нежелательные обращения к компьютеру или сети, а IP-маскировка позволяет организовать работу в Internet пользователей всей локальной сети при наличии одного IP-адреса. Опции ядра системы, предназначенные для фильтрации, перечислены ниже.

- Socket Filtering. В обычных условиях ядро направляет все пакеты, полученные через некоторое гнездо, программе, которая создала это гнездо. Опция Socket Filtering позволяет указать ядру на то, что принятые пакеты должны быть сначала переданы небольшой программе (которая называется фильтром). Эта программа способна блокировать некоторые из пакетов. Как правило, программы могут работать без данной опции. Исключение составляют последние варианты серверов DHCP и клиентов DHCP. Если в вашей сети используются средства DHCP (Dynamic Host Configuration Protocol — протокол динамической конфигурации узла), данная опция должна быть установлена.

- Network Packet Filtering. Данная опция является наиболее важным средством фильтрации, так как именно она делает возможной работу брандмауэра и IP-маскировку. Обычно опция Network Packet Filtering устанавливается; при этом становится доступной опция Network Packet Filtering Debugging, которую можно использовать для решения возникающих проблем. Кроме того, становится также доступным подменю IP: Netfilter Configuration. В этом подменю отображаются описанные ниже опции.

- Connection Tracking. Эта опция обеспечивает более высокую степень контроля над сетевыми соединениями, чем это возможно в обычных условиях. Как правило, маршрутизаторы ограничиваются пересылкой информационных пакетов между сетевыми интерфейсами. Если опция Connection Tracking активна, система запоминает IP-адрес источника, IP-адрес назначения и порты для дальнейшего использования. Эта возможность необходима для реализации IP-маскировки. В других случаях опцию Connection Tracking можно отключить. Если данная опция установлена, доступны опции поддержки FTP, что позволяет обеспечить работу данного протокола при наличии IP-маскировки.

- IP Tables Support. Данная опция включает поддержку ядром утилиты iptables, используемой для реализации брундмауэтов и осуществления IP-маскировки (эти вопросы будут подробно обсуждаться в главе 25). При установленной опции IP Tables Support становятся доступны подопции, позволяющие настроить средства поддержки iptables для выполнения конкретных задач. Многие из этих подопции задают соответствие ядра определенному типу, и их имена имеют вид *Tun* Match Support. Из них очень важна опция Connection S*ate Match Support, которая позволяет осуществлять *проверку пакетов с учетом состояния* (stateful packet inspection). Эта операция применяется в брундмауэрах и подробно рассматривается в главе 25. Также важны опции Packet Filtering, Full NAT и LOG Target Support и их подопции. Установив данные опции, вы можете использовать ваш компьютер как брандмауэр или осуществлять IP-маскировку. Для независимой рабочей станции или сервера опцию Full NAT можно не указывать.

- ipchains (2.2-Style) Support. В некоторых случаях бывает необходимо обеспечить работу сценариев брандмауэра, ориентированных на использование утилиты ipchains (эта утилита применялась при работе с версиями ядра 2.2.x). Поддержку ipchains можно включить в том случае, если средства IP Tables Support не были скомпилированы непосредственно в ядро системы. (Средства iptables и ipchains выполняют приблизительно одинаковые действия, но они не совместимы друг с другом.) Если вы создаете брандмауэр с нуля, можете смело отключить поддержку ipchains.

- ipfwadm (2.0-Style) Support. При работе с версиями 2.0.x ядра для создания брандмауэров использовалось инструментальное средство ipfwadm. Чтобы использовать сценарии брандмауэра, ориентированные на ipfwadm, надо установить данную опцию. Следует помнить, что средства поддержки ipfwadm не совместимы ни с iptables, ни с ipchains. Если вы не используете ipfwadm-сценарии либо твердо решили преобразовать их для работы с iptables, можете отказаться от установки данной опции.

По мере перехода от версий 2.0.x к версиям 2.4.x ядра Linux средства поддержки фильтрации пакетов становились все сложнее. В ядре 2.4.x предусмотрены многие дополнительные возможности; создавая брандмауэр, важно активизировать те опции, которые необходимы для решения конкретной задачи. Если вы сомневаетесь, нужна ли та или иная опция из меню IP: Netfilter Configuration, рекомендую вам установить ее. В этом случае объем ядра несколько возрастет, но вы получите возможность использовать различные правила брандмауэра.

ВНИМАНИЕ Вам может показаться, что использовать правила брандмауэра на машине под управлением Linux не обязательно, особенно если она находится в сети, которая защищена выделенным брандмауэром. К сожалению, в системе защиты многих сетей есть недостатки, поэтому дополнительные меры предосторожности не помешают. Возможно, вам потребуется установить на своем компьютере дополнительный простой брандмауэр.

Опции маршрутизации TCP/IP

Маршрутизатор — это компьютер, который непосредственно передает данные из одной сети в другую. Маршрутизаторы также часто называют *шлюзами*. Так, например, маршрутизатор может понадобиться для связи сети, принадлежащей отделу большой корпорации, с корпоративной сетью. Корпорация, в свою очередь, использует маршрутизатор для обеспечения связи своей сети с Internet. Рассмотрению опций маршрутизации посвящена глава 24. Сейчас вам достаточно знать лишь то, что для ядра Linux предусмотрен ряд опций, являющихся подопциями IP: Advanced Router.

Опции поддержки IPv6

Работа Internet обеспечивается за счет протоколов семейства TCP/IP, в частности, для передачи пакетов используется протокол IP (IPv4). К сожалению, на сегодняшний день уже невозможно игнорировать тот факт, что версия IPv4 устарела. Для представления IP-адреса в IPv4 используется 32-разрядное число, т. е. общее число адресов равно 2^{32}, или 4294967296. Вследствие неэффективности механизма распределения адресов реальное их количество оказывается намного меньшим. В результате возникла проблема нехватки IP-адресов. Кроме того, недостатки в защите IPv4 позволяют злоумышленникам вмешиваться в сеансы сетевого взаимодействия. На момент написания данной книги, т. е. в 2002 г., с проблемами, связанными с использованием IPv4, еще можно мириться, но их придется решить в течение ближайшего десятилетия.

В настоящее время разрабатывается версия IPv6, призванная заменить IPv4. В IPv6 поддерживаются 128-разрядные IP-адреса. Общее число IP-адресов равно 2^{128}, или 3,4 x 10^{38} — приблизительно $2{,}2 \times 10^{18}$ адресов на квадратный миллиметр поверхности Земли. IPv6 также обеспечивает дополнительные средства защиты. В настоящее время число сетей, в которых используется IPv6, очень мало. Если ваш компьютер подключен к такой сети или если вы собираетесь в качестве эксперимента организовать обмен данными во внутренней сети предприятия посредством IPv6, вам надо активизировать средства поддержки IPv6, установив для этого опцию IPv6 Protocol (Experimental) в меню Networking Options. После установки данной опции вам станут доступны дополнительные опции, объединенные в подменю IPv6: Netfiler Configuration. В этом подменю также находятся описанные ранее опции фильтрации, но они ориентированы на работу с протоколом IPv6.

Чтобы активизировать средства поддержки IPv6, надо установить значения Yes опции Prompt for Development или Incomplete Code/Drivers в меню Code Maturity Level Options. То же самое надо сделать при работе с любыми "экспериментальными" драйверами. Со временем эксперименты с IPv6 закончатся, и опция, включающая поддержку IPv6, будет относиться к числу основных опций. Пока это не произошло, при работе с IPv6, как и при использовании других "экспериментальных" средств, следует соблюдать осторожность.

Опции **QoS**

Предположим, что компьютер под управлением Linux действует как маршрутизатор в сети с напряженным трафиком или выполняет роль сервера и обрабатывает при этом большой объем данных. При этом может возникнуть ситуация, когда система будет в течение некоторого времени получать большее число пакетов, чем она может обработать. Очевидно, что в этом случае необходимы специальные средства планировки, которые

устанавливали бы очередность передачи пакетов. Как правило, в системе Linux используется стратегия FIFO (first in/first out — "первый пришел — первый вышел"), согласно которой пакет, предназначенный для передачи, находится в очереди до тех пор, пока не будут переданы все пакеты, поставленные в очередь раньше него. Но в некоторых случаях необходимо предоставить пакетам определенного типа некоторые преимущества. Это могут быть пакеты, адресованные в конкретную сеть, или пакеты, которые содержат информацию, соответствующую определенному протоколу. Так, например, пакеты, содержащие информацию реального времени, например данные Internet-телефонии, целесообразно передавать вне очереди. Назначать приоритеты пакетам позволяют опции QoS (quality of service — качество сервиса). Эти опции доступны посредством подменю QoS and/or Fair Queueing меню Networking Options.

Для того чтобы реализовать систему QoS, необходимо выбрать опцию QoS and/or Fair Queueing в одноименном меню. В результате автоматически устанавливается ряд опций этого меню. Другие опции задаются отдельно. Основными из них являются опции планирования передачи пакетов и организации очереди, такие как CBQ Packet Scheduler и SFQ Queue. Эти опции позволяют ядру выполнять более сложную обработку пакетов по сравнению с традиционно используемым принципом FIFO. Опции QoS Support и Packet Classifier API, а также их подопции позволяют использовать Differentiated Services и Resource Reservation Protocol. При этом появляется возможность обмена QoS-приоритетами с другими маршрутизаторами. Если все маршрутизаторы на пути от одного узла к другому поддерживают совместимые между собой протоколы QoS, скорость передачи важных данных может быть увеличена за счет задержки информации, время доставки которой некритично.

Если система не выполняет функции маршрутизатора, опции QoS в ней, как правило, не используются. Если же вы создаете маршрутизатор, а в особенности, если он планируется для использования в сети с интенсивным обменом данными, желательно установить эти опции. Активизировав одну опцию, целесообразно активизировать и все остальные, в противном случае система не будет обладать должной гибкостью. Так, например, если вы не установите опцию U32 Classifier, то не сможете задавать приоритеты исходя из адресов назначения пакетов.

На практике использование средств QoS предполагает применение расширенных средств маршрутизации, таких как ip и tc. Об этих инструментах речь пойдет в главе 24, однако они слишком сложны, чтобы привести их исчерпывающее описание в рамках одной главы. Дополнительную информацию об ip и о tc можно найти в документах *iproute2 + tc Notes* (`http://snafu.freedom.org/linux2.2/iproute-notes.html`) и *Differentiated Services on Linux* (`http://diffserv.sourceforge.net`).

Поддержка протоколов высокого уровня

В ядре Linux предусмотрена поддержка нескольких протоколов высокого уровня. Благодаря этому коды, отвечающие за работу с этими протоколами, выполняются намного быстрее, чем соответствующие коды обычных пользовательских программ. Кроме того, поддержка высокоуровневых протоколов в ядре обеспечивает более тесную интеграцию этих протоколов с остальными компонентами операционной системы. Например, включение в состав ядра средств поддержки NFS позволяет монтировать удаленные ресурсы и использовать их так же, как и компоненты локальной файловой системы. В версиях

2.4.x ядра реализована поддержка трех важных высокоуровневых протоколов: HTTP, NFS и SBM/CIFS.

В приведенном перечне содержатся не все протоколы, поддерживаемые ядром. Кроме указанных выше, ядро Linux позволяет работать и с другими протоколами, в частности, с различными протоколами, обеспечивающими разделение сетевых ресурсов.

Ускорение HTTP-обмена

Работа World Wide Web в основном базируется на использовании протокола HTTP (Hypertext Transfer Protocol — протокол передачи гипертекстовой информации). По сути, в ядре Linux реализован простой сервер HTTP, который включается при установке опции Kernel HTTPd Acceleration. Для настройки и активизации этого сервера в псевдофайлы, находящиеся в каталоге `/proc/sys/net/khttpd`, записываются специальные значения. Вопросы работы со встроенным сервером HTTP подробно рассматриваются в главе 20.

Реализовать сервер HTTP в составе ядра оказалось сравнительно не сложно, так как передача клиенту статических Web-страниц (документов, содержимое которых не изменяется при различных обращениях клиентов) мало отличается от копирования файлов с диска на удаленные компьютеры. Ядро может выполнять эту операцию гораздо эффективнее, чем пользовательские программы. Для обслуживания запросов, связанных с предоставлением динамических Web-страниц, а также запросов, предполагающих сложную обработку статических документов, ядро обращается к обычному Web-серверу, например Apache. При этом нет необходимости в специальных настройках Apache; этот сервер попросту "не видит" запросов на получение статических Web-страниц.

Опции для работы с NFS

NFS (Network Filesystem — сетевая файловая система), разработанная Sun, предназначена для организации совместного использования файлов несколькими компьютерами. Благодаря NFS обращение к удаленным файлам осуществляется так же, как и обращение к файлам на локальной машине. Поддержка NFS реализована в ядре Linux. Подробно вопросы работы с NFS рассматриваются в главе 8. Для того чтобы иметь возможность монтировать каталоги, экспортируемые другими компьютерами, надо установить опции, отвечающие за поддержку NFS. Опции, включающие средства клиента и сервера NFS, содержатся в подменю Network File Systems меню File Systems (а не в меню Networking Options, как это можно было бы ожидать). Опции для работы с NFS перечислены ниже.

- NFS File System Support. Эта опция включает базовые средства поддержки клиента NFS (т. е. средства, позволяющие монтировать удаленные каталоги NFS и пользоваться ими так же, как и фрагментами локальной файловой системы).
- Provide NFSv3 Client Support. За время своего развития система NFS претерпела многочисленные изменения. Последней была разработана версия 3 (NFSv3). Поддержку этой версии необходимо задавать явно, так как стандартные средства, включаемые с помощью NFS File System Support, не обеспечивают надежную работу NFSv3. Для поддержки NFSv3 опция NFS File System Support также должна быть установлена.

- Root File System on NFS. Чтобы эта опция стала доступной, надо установить опцию IP: Kernel Level Autoconfiguration в меню Networking Options. Опция Root File System on NFS позволяет монтировать внешний каталог как корневую файловую систему Linux. Эта возможность обычно используется только на бездисковых рабочих станциях.

- NFS Server Support. Чтобы компьютер под управлением Linux мог работать как сервер NFS (т. е. мог предоставлять свои каталоги другим компьютерам), желательно установить данную опцию. В большинстве случаев она позволяет ускорить работу сервера NFS.

- Provide NFSv3 Server Support. Если вы хотите обеспечить работу сервера NFS, ориентированного на использование средств ядра, установите данную опцию. Как и в случае NFSv3-клиента, для поддержки NFSv3 должны также быть включены базовые средства NFS.

Чаще всего средства NFS используются в Linux и различных версиях UNIX. Обмен файлами с прочими системами реализуется другими средствами, одно из которых рассматривается ниже.

Опции для работы с SMB/CIFS

NFS — не единственный протокол, обеспечивающий разделение файлов. В Macintosh для этой цели используется AppleTalk; большой популярностью пользуются также протоколы IPX/SPX, разработанные Novell. В системе Linux, помимо NFS, часто применяется система Samba, которая реализует протокол SMB (Server Message Block — блок сообщений сервера). Эти средства известны также под названием CIFS (Common Internet Filesystem — общая межсетевая файловая система). Подробно настройка и использование Samba рассматриваются в главе 7.

Samba предоставляет средства, необходимые для того, чтобы система Linux функционировала как SMB/CIFS сервер; при этом специальная настройка ядра не требуется. Если вы хотите монтировать в Linux разделяемые каталоги SMB/CIFS, вам надо установить опцию SMB File System Support, которая действует подобно опции NFS File System Support. Две подопции (Use a Default NLS и Default Remote NLS Option) обеспечивают преобразование имен файлов на основе NLS (National Language Support — поддержка национальных языков). Эти опции полезны при использовании алфавитов, отличных от латинского, например кириллицы, а также символов с дополнительными элементами.

Чтобы обеспечить работу Linux как клиента SMB/CIFS, не обязательно устанавливать опции ядра SMB/CIFS. Это также позволяет сделать программа `smbclient`. Данная программа не монтирует ресурсы; для передачи разделяемых файлов предоставляется интерфейс, подобный интерфейсу клиентской программы FTP.

Поддержка альтернативных сетевых протоколов

Несмотря на то что семейство протоколов TCP/IP, обеспечивающее работу глобальной сети Internet, пользуется широкой популярностью, в системе Linux также реализованы другие стеки протоколов. Опции, включающие поддержку различных сетевых протоко-

лов, представлены в меню Networking Options. Многие из опций, содержащихся в этом меню, на самом деле являются подопциями TCP/IP Networking. За ними следуют опции, соответствующие другим протоколам; некоторые из них перечислены ниже.

- Asynchronous Transfer Mode (ATM). Этот набор экспериментальных опций предназначен для поддержки аппаратных средств и протоколов ATM. Опции ATM в равной мере относятся как к аппаратным средствам, так и к сетевым протоколам, но в версии ядра 2.4.x они, как и другие опции поддержки сетевых протоколов, сосредоточены в меню Networking Options.

- The IPX Protocol. Семейство протоколов IPX (Internetwork Packet Exchange — межсетевой обмен пакетами), разработанное Novell, применяется во многих сетях, в частности в Netware. Для работы с этими протоколами вам потребуется дополнительное программное обеспечение, например Mars_nwe (дополнительную информацию о нем вы можете получить по адресу `http://www.redhat.com/support/docs/tips/Netware/netware.html`). Опция NCP File System Support, расположенная в подменю Network File Systems меню File Systems, дает возможность монтировать тома Netware, подобно тому, как опции NFS и SMB/CIFS позволяют монтировать фрагменты файловой системы Windows.

- AppleTalk Protocol Support. Компания Apple разработала стек протоколов AppleTalk, предназначенный для разделения файлов и принтеров на компьютерах Macintosh. Для поддержки AppleTalk в Linux используются средства ядра, а также пакет Netatalk (`http://netatalk, sourceforge.net/`).

- DECnet Support. Корпорация DEC (Digital Equipment Corporation) разработала для своих компьютеров сетевую технологию под названием DECnet. Система Linux включает средства поддержки DECnet, но для работы с данным стеком протоколов необходимо установить специальный программный пакет. Дополнительную информацию об использовании DECnet можно получить, обратившись по адресу `http://linux-decnet.sourceforge.net`.

В системе Linux также предусмотрены опции поддержки менее популярных протоколов, например Acorn Econet. В большинстве случаев при установке системы достаточно включить поддержку TCP/IP и одного-двух дополнительных стеков протоколов. Вследствие бурного развития Internet производители, ранее использовавшие собственные стеки протоколов, модернизировали свои инструментальные средства для работы с TCP/IP. Так, например, несмотря на то, что Apple в течение длительного времени применяла AppleTalk, средства разделения файлов на компьютерах Macintosh сейчас используют как AppleTalk, так и TCP/IP.

В ядре Linux отсутствуют средства поддержки стека протоколов NetBEUI. Для работы с разделяемыми файлами Windows в настоящее время с успехом применяются средства SMB/CIFS.

В главе 3 детально рассматриваются стеки сетевых протоколов и их использование.

Опции для работы с аппаратными средствами

В меню Network Device Support содержатся опции, определяющие взаимодействие системы с различными сетевыми устройствами. Самые важные из этих опций управляют использованием драйверов сетевых карт. Несмотря на то что в настоящее время наиболее распространены Ethernet-карты, в меню Network Device Support содержатся опции для работы и с другими устройствами, предназначенными для создания локальных сетей. Кроме того, Linux предоставляет опции, включающие драйверы устройств дальней связи и драйверы беспроводных устройств. Устройства PC Card (применяющиеся в портативных компьютерах) описаны в отдельном подменю, которое входит в состав меню Network Device Support. Вы также можете выбрать устройства, позволяющие устанавливать соединения по телефонным линиям через модемы, и другие аппаратные средства.

Для того чтобы получить доступ к описанным выше опциям, надо установить опцию Network Device Support, которая находится в начале меню Network Device Support. Если вы не сделаете это, опции в данном меню будет не доступны.

Устройства Ethernet

На момент написания данной книги, т. е. в 2002 г., подавляющее большинство локальных сетей строились на базе Ethernet. Беспроводные технологии также пользуются определенной популярностью, но сети, созданные на их основе, проигрывают сетям Ethernet в скорости обмена. Одной из проблем, усложняющих процесс администрирования операционных систем, является тот факт, что число разновидностей Ethernet-карт исчисляется сотнями и даже тысячами.

К счастью, при построении Ethernet-карт применяется ограниченный набор микросхем, поэтому для поддержки подавляющего большинства таких устройств достаточно шестидесяти драйверов. Опции, управляющие использованием этих драйверов, сосредоточены в двух подменю: Ethernet (10 or 100Mbit) и Ethernet (1000 Mbit). Большинство опций находится в первом меню. Соответствующие им драйверы, как следует из названия подменю, ориентированы на устройства, обеспечивающие скорость обмена 10 и 100 Мбод. На момент написания этой книги наибольшей популярностью пользуются Ethernet-платы 100 Мбод (100-мегабитовая Ethernet), а в некоторых случаях применяются более новые платы 1000 Мбод (или гигабитовая Ethernet). Разрабатываются также Ethernet-карты 10 Гбод.

Ethernet-сети различаются не только по скорости обмена данными, но и по типу кабеля. Для соединения устройств применяются коаксиальные кабели (в некоторых типах 10 мегабитовых Ethernet-сетей), витые пары (во всех 100-мегабитовых Ethernet-сетях, а также в некоторых типах 10-мегабитовых и гигабитовых Ethernet-сетей) и волоконно-оптические кабели (в некоторых типах гигабитовых Ethernet-сетей). Витые пары обеспечивают соединение на расстоянии до 100 метров (обычно такое соединение устанавливается между компьютером и концентратором либо коммутатором). Волоконно-оптические соединения допускают обмен данными на расстоянии до 5 километров.

Структура меню Ethernet (10 or lOOMbit) далека от совершенства. В начале меню перечислены сетевые карты 3Com, SMC, Racal-Interlan и некоторых других производителей. За ними расположены устройства ISA (Industry Standard Architecture), затем следуют устройства EISA (Extended ISA), VLB (VESA Local Bus) и PCI (Peripheral Component In-

terconnect). Завершается список группой параллельных Ethernet-адаптеров. В результате для поиска требуемого устройства приходится просматривать две-три группы опций.

Существуют также Ethernet-устройства, драйверы которых не устанавливаются посредством опций меню Network Device Support или его подменю. Так, например, для устройств PC Card используются специальные драйверы, а адаптеры USB — Ethernet описаны в меню USB Support. Для того чтобы использовать устройство USB, вам надо, в зависимости от контроллера, присутствующего на материнской плате, установить либо опцию UHCI Support, либо OHCI Support, а также включить опцию, которая соответствует требуемому драйверу, например USB ADMtek Pegasus-Based Ethernet Device Support.

Альтернативные средства для создания локальных сетей

Несмотря на свою популярность, Ethernet — отнюдь не единственное устройство, позволяющее создать локальную сеть. В ядре Linux предусмотрена поддержка различных типов сетей. Число драйверов для устройств, отличных от Ethernet, невелико, но это не означает, что средства для организации соответствующих сетей разработаны недостаточно хорошо. Ниже приведены некоторые опции, присутствующие в меню Network Device Support.

- Token Ring. В течение многих лет технология Token Ring, разработанная IBM, была главным конкурентом Ethernet, однако начиная с 1990 г. преимущество Ethernet стало очевидным. Большинство карт Token Ring поддерживают скорость обмена до 16 Мбод, но в настоящее время появились модели 100 Мбод. Максимальное расстояние между устройствами в сети Token Ring составляет 150-300 метров. Средства поддержки устройств Token Ring сосредоточены в подменю Token Ring Devices меню Network Device Support.

- LocalTalk. Для компьютеров Macintosh компания Apple разработала сетевую технологию, включающую протоколы как аппаратного (LocalTalk), так и программного (AppleTalk) уровня. Для взаимодействия с сетями LocalTalk были разработаны устройства x86; некоторые из них поддерживает система Linux. Соответствующие опции находятся в меню AppleTalk Devices. Как ни странно, версия Linux, разработанная для Macintosh, не поддерживает LocalTalk. На момент написания данной книги максимальная скорость обмена данными в сети LocalTalk составляла 2 Мбод.

- ARCnet. Это сетевая технология, которая в основном используется в специальных целях, например, для подключения охранных устройств или для сбора результатов научных экспериментов. Устройства ARCnet обеспечивают скорость обмена от 19 Кбод до 10 Мбод. Соединение устройств осуществляется с помощью коаксиального кабеля, витой пары или волоконно-оптического кабеля. Опции поддержки ARCnet находятся в подменю ARCnet Devices. Помимо драйвера устройства, вам необходимо включить драйвер, предназначенный для поддержки формата ARCnet-пакетов (RFC 1051 или RFC 1201).

- FDDI и CDDI. FDDI (Fiber Distributed Data Interface — волоконно-оптический интерфейс распределенных данных) и CDDI (Copper Distributed Data Interface — "медный"

интерфейс распределенных данных) предназначены для создания сетей со скоростью обмена информацией порядка 100 Мбод. Преимущество FDDI перед 10 мегабитовой Ethernet состоит в том, что данная технология обеспечивает связь на расстоянии до 2 километров. Следует заметить, что гигабитовая Ethernet с передачей данных по волоконно-оптическому кабелю обеспечивает дальность до 5 километров. Для того чтобы опции ядра 2.4.17, предназначенные для поддержки FDDI/CDDI, стали доступны, надо установить опцию FDDI Driver Support.

- HIPPI. HIPPI (High Performance Parallel Interface — высокопроизводительный параллельный интерфейс) обеспечивает скорость обмена данными 800 или 1600 Кбод. При соединении с помощью витой пары максимальная дальность составляет до 25 метров, многорежимное волоконно-оптическое соединение обеспечивает дальность до 300, а однорежимное волоконно-оптическое соединение — до 10 километров. Ядро 2.4.17 поддерживает единственное устройство HIPPI Essential RoadRunner. Заметьте, что драйвер данного устройства считается экспериментальным.

- Fiber Channel. Данный тип сетевого интерфейса поддерживает как волоконно-оптическое соединение, так и соединение с помощью обычного кабеля и обеспечивает скорость передачи данных 133-1062 Мбод. При использовании волоконно-оптического кабеля максимальная дальность составляет до 10 километров. Ядро 2.4.17 поддерживает единственное устройство Fiber Channel Interphase 5526 Tachyon.

Некоторые из описанных выше сетевых сред, например Token Ring, используются для создания локальных сетей, компоненты которых размещаются в одном здании либо в нескольких зданиях, расположенных рядом. Другие, например FDDI и HIPPI, чаще применяются для организации соединения между компьютерами, расположенными на большом расстоянии, например, находящимися в различных помещениях на территории университетского городка. Поддержка этих технологий системой Linux означает, что компьютер под управлением Linux может выступать в роли маршрутизатора, связывающего между собой различные типы сетей.

Далее в этой книге мы будем предполагать, что локальная сеть, к которой подключены компьютеры под управлением Linux, создана на базе технологии Ethernet. Если при решении конкретной задачи вам потребуется организовать поддержку других сетей, то единственное, что вам придется для этого сделать, — это изменить имя сетевого интерфейса. Устройствам Ethernet имена присваиваются следующим образом. Первое устройство имеет имя eth0, второе — `eth1` и т. д. Аналогично именуются другие устройства, например, первому устройству Token Ring присваивается имя `tr0`, а второму устройству FDDI — имя `fddi1`.

Устройства с широкой полосой пропускания и устройства, обеспечивающие связь на большой дальности

Термин "устройства с широкой полосой пропускания" имеет несколько значений. Во-первых, этот термин обозначает устройства, позволяющие одновременно передавать различные типы информации, например, видео, аудио и обычные цифровые данные. Во-вторых, устройствами с широкой полосой пропускания называют устройства с большой

пропускной способностью, позволяющие увеличить скорость передачи данных по коммутируемым линиям (например, реализовать скорость обмена до 200 Кбод). Конечно, величина 200 Кбод выглядит более чем скромно по сравнению со скоростями, которые обеспечивает технология Ethernet, но все же 200 Кбод — это значительно больше, чем скорость 56 Кбод, которой позволяют добиться обычные телефонные линии.

Устройства с широкой полосой пропускания часто применяются в небольших компаниях для связи с серверами Internet-провайдеров или для организации взаимодействия компьютеров, расположенных далеко друг от друга. В большинстве случаев посредством устройств с широкой полосой пропускания пользователи подключают свои компьютеры к Internet. Этим данные устройства отличаются от других сетевых устройств, которые связывают несколько компьютеров в одну локальную сеть. Часто, подключая компьютеры пользователей через устройства с широкой полосой пропускания, провайдеры накладывают ограничения на их действия. Так, например, провайдер может запретить пользователю устанавливать на своем компьютере серверы.

На момент написания данной книги наиболее популярными из устройств с широкой полосой пропускания были DSL (Digital Subscribe Line — цифровая линия подписки) и кабельные модемы. Существует несколько разновидностей DSL, например ADSL (Asymmetric DSL — асимметричная DSL) и SDSL (Single-Line, или Symmetric DSL — одиночная линия, или асимметричная DSL). В процессе работы **DSL-устройства** передают высокочастотные сигналы по обычным телефонным линиям. Кабельные модемы пересылают данные по кабельным телевизионным сетям и используют полосу частот свободного телевизионного калана. Некоторые из устройств с широкой полосой пропускания передают данные через спутниковые системы, по радиоканалам и по волоконно-оптическим кабелям.

Большинство соединений с широкой полосой пропускания используют специальные внешние модемы, а подключение к локальным компьютерам производится через Ethernet-порты. Поэтому, для того, чтобы работа через такое соединение стала возможной, необходим Ethernet-адаптер, кроме того, надо обеспечить поддержку этого адаптера с помощью стандартного драйвера Linux. Сам по себе широкополосный модем может работать без специального драйвера, но некоторые провайдеры используют PPPoE (Point-to-Point Protocol over Ethernet — протокол межузлового взаимодействия через Ethernet). В этом случае необходимо обеспечить поддержку данного протокола в системе Linux, для чего надо установить опцию PPP over Ethernet в меню Network Device Support (эта опция считается экспериментальной). Для того чтобы опция PPP over Ethernet была доступна, необходимо включить опцию PPP Support. Средства PPPoE понадобятся также в том случае, если вы собираетесь запускать на вашем компьютере пакет Roaring Penguin PPPoE (этот пакет находится по адресу `http://www.roaringpenguin.com/pppoe/`).

Некоторые широкополосные модемы используют вместо Ethernet интерфейс USB. В ядре 2.4.17 поддержка данных устройств не предусмотрена, но Alcatel предоставляет Linux-драйверы для своего модема Speed Touch USB DSL (`http://www.alcatel.com/consumer/dsl/supuser.htm`). Информацию о других **USB-продуктах** можно найти по адресу `http://www.linux-usb.org`.

Ряд широкополосных модемов, особенно предназначенных для установления низкоуровневых ADSL-соединений, поставляются вместе с внутренними **PCI-картами**. Лишь немногие из этих устройств поддерживаются в системе Linux. В частности, в ядре 2.4.17 предусмотрена поддержка General Instruments Surfboard 1000 и некоторых односторонних модемов. Односторонними (one-way) называются модемы, которые могут только прини-

мать данные. При работе с такими модемами для передачи данных используются обычные модемы, взаимодействующие по телефонным линиям. В настоящее время односторонние модемы встречаются очень редко. Драйверы для модема Diamond 1MM DSL можно найти по адресу `http://www.rodsbooks.com/network/network-dsl.html`, однако они являются модернизацией существующих Ethernet-драйверов и при использовании их в 2.4.x и более старших версиях ядра могут возникать проблемы.

Компьютеры, расположенные на большом расстоянии друг от друга, часто соединяют с помощью *выделенных линий,* которые часто называют *арендованными линиями.* Роль выделенной линии чаще всего выполняет обычная телефонная линия, арендованная у телефонной компании. По такой линии не передается тональный сигнал, все меры по организации взаимодействия модемов должны принять специалисты, занимающиеся созданием сети. Сети, созданные на базе выделенных линий, обычно называют *региональными сетями,* или WAN (Wide-Area Network). При организации региональных сетей часто используются специальные внешние устройства, называемые *WAN-маршрутизаторами.* Для создания региональных сетей могут также применяться специальные интерфейсные карты. В системе Linux предусмотрена поддержка таких устройств; соответствующие опции находятся в подменю Wan Interfaces меню Network Device Support. Как и при работе со многими другими подменю, для того, чтобы получить доступ к опциям, которые соответствуют конкретным устройствам, необходимо включить опцию Wan Interfaces Support, расположенную в начале данного подменю.

Беспроводные устройства

В течение последних лет сети, созданные на базе беспроводных устройств, становятся все более популярными. Беспроводные технологии позволяют компьютерам обмениваться данными, даже если они не подключены к сети с помощью сетевых кабелей. Беспроводные сети очень удобно устанавливать, если прокладка кабеля по каким-либо причинам затруднена или в том случае, если пользователь, работающий на портативном компьютере, вынужден часто перемещаться и не имеет возможности подключиться к сетевому кабелю.

К сожалению, на момент написания данной книги беспроводные сети по многим характеристикам уступают Ethernet-сетям. Беспроводные сети гораздо дороже Ethernet-сетей, скорость передачи данных относительно мала, а разработка стандартов, регламентирующих структуру и работу таких сетей, еще продолжается. В настоящее время основными стандартами являются 802.11 и 802.11b. Стандарт 802.11 определяет скорость обмена 2 Мбод со снижением до 1 Мбод. *Снижением* называется повторный обмен инициализационными параметрами в случае, если уровень сигнала слишком низкий или если помехи искажают сигнал. Стандарт 802.11b определяет скорость 11 Мбод со снижением до 5,5, 2 и 1 Мбод. Существует также беспроводная технология Bluetooth, которая поддерживает скорость обмена до 1 Мбод. Основное направление развития беспроводных сетей связано с увеличением скорости передачи данных. Планируется разработать беспроводные версии ATM со скоростью обмена до 155 Мбод.

Как правило, беспроводные локальные сети создаются на базе беспроводных устройств PC Card, используемых в портативных компьютерах. В одних случаях эти устройства могут непосредственно обмениваться данными друг с другом, в других случаях для взаимодействия необходима базовая станция, которая может выполнять роль шлюза к сети, созданной традиционными средствами. С помощью базовой станции можно также

организовать подключение к Internet. Существуют беспроводные ISA и PCI-карты, которые позволяют подключать к беспроводным сетям настольные компьютеры и рабочие станции. Для поддержки устройств PC Card, ISA и PCI в системе Linux необходимо установить соответствующие драйверы; для обеспечения работы базовой станции никакое специальное программное обеспечение не требуется.

Для включения средств поддержки беспроводных устройств в ядре Linux предусмотрены опции, которые содержатся в меню Wireless LAN (Non-Hamradio). Опции в данном меню расположены по типам устройств, а не по технологиям, используемым в них (например, 802.11b или Bluetooth). Кроме того, существуют пакеты Wireless Extensions и Wireless Tools, которые упрощают управление беспроводными сетями, созданными с помощью средств Linux. Информацию об этих пакетах можно найти по адресу `http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html`. Здесь же расположены дополнительные ссылки на документы, имеющие отношение к беспроводным сетям.

Устройства PC Card

Большинство портативных компьютеров имеют как минимум одно гнездо PC Card. (Часто в документации по системе Linux для обозначения устройств PC Card используется старый термин PCMCIA. Устройства PC Card можно подключать и удалять в процессе работы компьютера. Поскольку при разработке системы Linux предполагалось, что сетевые интерфейсы не должны исчезать без предупреждения, для работы с устройствами PC Card создан специальный пакет Card Services. При подключении или удалении устройств PC Card инструменты Card Services соответственно запускают или останавливают компоненты ядра, имеющие отношение к этим устройствам. Дополнительную информацию о Card Services можно получить, обратившись по адресу `http://pcmcia-cs.sourceforge.net`.

В ядре 2.4.17 опции поддержки многих устройств PC Card находятся в меню PCMCIA Network Device Support. Некоторые из опций, соответствующих беспроводным устройствам, находятся в меню Wireless LAN (Non-Hamradio). После того как вы включите соответствующую опцию и сконфигурируете карту, она будет работать как обычное устройство ISA или PCI. Так, например, Ethernet PC Card распознается системой как устройство eth0, а для его настройки используются стандартные инструменты, которые будут рассмотрены в главе 2.

Версии ядра, предшествующие 2.4.x, требуют для поддержки устройств PC Card специальный пакет драйверов. Следует также заметить, что многие устройства PC Card до сих пор не поддерживаются стандартным ядром Linux. Упомянутый выше пакет драйверов входит в набор Card Services. Если вы используете ядро 2.4.x, для работы с PC Card вам вряд ли придется устанавливать специальные драйверы; такие драйверы могут потребоваться для модемов, SCSI-адаптеров и других устройств.

Устройства для связи по коммутируемым линиям

Для установления связи по коммутируемой линии чаще всего используются обычные модемы. Чтобы обмен данными был возможен, необходимо также включить средства поддержки протокола PPP (Point-to-Point Protocol — протокол межузлового взаимодействия). Соединение по коммутируемой линии устанавливается из командной строки либо с помощью инструмента, предоставляющего графический пользовательский интерфейс. Подробно вопросы установления соединения будут рассматриваться в главе 2.

Для того чтобы активизировать средства поддержки PPP, надо установить опцию PPP (Point-to-Point Protocol) Support в меню Network Device Support. Если вы включите эту опцию, станут доступны некоторые дополнительные опции, например PPP Support for Async Serial Ports и PPP Deflate Compression. Эти опции не всегда необходимы, но в некоторых случаях они могут повысить эффективность обмена информацией за счет сжатия данных, передаваемых по линии связи. Если вы собираетесь использовать средства ядра PPPoE для работы через DSL-соединение, вам придется установить экспериментальную опцию PPP over Ethernet. Для некоторых дополнительных PPPoE-пакетов эта опция не нужна.

Протокол PPP часто используется при передаче информации через соединения, устанавливаемые без участия модема, например, при обмене данными между компьютерами, подключенными через последовательные порты. Следует заметить, что такое подключение применяется чрезвычайно редко, так как Ethernet-карты недорогие и обеспечивают гораздо более эффективное взаимодействие по сети. Соединение через последовательные порты обычно устанавливают на короткое время, когда нет смысла использовать сетевые карты.

PPP — не единственный протокол, посредством которого может осуществляться связь по коммутируемой линии. В ядре Linux поддерживается также протокол SLIP (Serial Line Internet Protocol — протокол Internet для обмена по последовательной линии), который выполняет практически те же функции, что и PPP. Особенности протокола SLIP таковы, что он плохо подходит для взаимодействия с Internet-провайдером, поэтому вам вряд ли придется использовать его. SLIP используется некоторыми инструментами Linux для выполнения действий на локальном компьютере. Например, утилита, поддерживающая *dial-on-demand*, т. е. устанавливающая PPP-соединение при обнаружении сетевой активности, использует SLIP для выявления попыток обращения к компьютерам за пределами локальной сети.

Помимо PPP и SLIP, для организации обмена данными между компьютерами может использоваться протокол PLIP (Parallel Line Internet Protocol — протокол Internet для обмена по параллельной линии). Как нетрудно догадаться, этот протокол применяется тогда, когда компьютеры соединены друг с другом через параллельный порт (порт принтера). Параллельный порт позволяет гораздо быстрее передавать данные, чем последовательный порт RS-232; несмотря на это, соединение через параллельный порт также применяется редко, поскольку Ethernet обеспечивает более высокое быстродействие. Для того чтобы использовать протокол PLIP, надо установить опцию PLIP (Parallel Port) Support в меню Network Device Support; при этом предварительно следует активизировать опцию Parallel Port Support в одноименном меню, а при работе на x86 надо также выбрать опцию PC-Style Hardware. Информацию об организации сети PLIP можно получить в документе *PLIP Mini-HOWTO* (`http://www.linuxdoc.org/HOWTO/mini/PLIP.html`). Если вы не можете воспользоваться кабелем Turbo Laplink, то найдете в этом документе рекомендации по изготовлению кабеля для соединения компьютеров.

Компиляция и установка ядра

До сих пор мы рассматривали опции ядра, имеющие отношение к сетевым протоколам и аппаратным средствам, используемым для соединения вашего компьютера с сетью. Компиляция ядра непосредственно не связана с обеспечением сетевого взаимодействия,

однако значение этой задачи нельзя недооценивать. Чтобы добавить или удалить некоторые сетевые средства, необходимо перекомпилировать ядро. В данном разделе рассматриваются основные вопросы компиляции и установки ядра системы.

ВНИМАНИЕ Если вы установили необходимые вам опции сетевого взаимодействия, это совсем не означает, что вы полностью выполнили настройку ядра. Существуют также опции, предназначенные для управления контроллерами EIDE, адаптерами SCSI, файловой системой на диске, а также многие другие опции. Несмотря на то что эти опции не рассматриваются в данной книге, они чрезвычайно важны для обеспечения нормальной работы операционной системы. Если вы неправильно установите значения соответствующих опций, система не будет загружаться либо будет работать некорректно (например, скорость обмена данными с диском может стать недопустимо низкой). Опции ядра подробно обсуждаются в документе *Linux Kernel HOWTO*, который находится по адресу `http://www.linuxdoc.org/HOWTO/Kernel-HOWTO.html` (и на многих других серверах). Вопросы настройки ядра также рассматриваются в ряде книг по Linux.

Драйверы, встроенные в ядро, и драйверы, реализуемые в виде модулей

Как вы уже знаете, при настройке ядра можно включить или отключить некоторые свойства ядра, например, вы можете разрешить или запретить использование конкретного Ethernet-адаптера. На рис. 1.1 видно, что существуют опции, значения которых можно выбирать более чем из двух возможных вариантов. В качестве примера рассмотрим опцию Packet Socket. Для этой опции могут быть заданы значения Y, M и N. Значение Y (Yes) указывает на то, что средства, соответствующие данной опции, должны быть включены в основной файл ядра, а значение N (No) запрещает использование этих средств. Значение M задает некоторое "промежуточное" решение. Если вы выберете значение M, то соответствующие средства будут скомпилированы, но не войдут в основной файл ядра. Вместо этого фрагмент кода будет реализован как отдельный модуль ядра, который по мере надобности загружается или удаляется из памяти. Для опций, являющихся подопциями других опций (например, Packet Socket: Mmapped IO, показанной на рис. 1.1), значение M обычно не предусмотрено. Решение о включении их в основной файл ядра или реализации в виде отдельного модуля принимается в зависимости от значения родительской опции.

Средства, включенные в основной файл ядра, доступны с момента загрузки системы и до окончания ее работы. Ситуация, при которой фрагмент кода будет удален из памяти, возникнуть не может. Существуют опции, для которых реализующий их программный код должен быть включен в основной файл ядра. Так, например, файловая система, в которой содержится корневой каталог системы, должна быть доступна с момента загрузки, поэтому соответствующий драйвер должен быть включен непосредственно в ядро. Если вы установили рассмотренную ранее опцию Root File System on NFS, вам придется скомпилировать средства поддержки сетевых устройств и включить их в ядро.

На первый взгляд может показаться, что все средства, которые будут использоваться при работе системы, следует включить в основной файл ядра, однако такой подход имеет серьезный недостаток: при этом увеличивается объем оперативной памяти, занимаемой

ядром. Кроме того, размер файла ядра на диске также увеличивается, что может создавать трудности при загрузке системы. Поэтому в системе Linux предусмотрена возможность компилировать средства, соответствующие большей части опций, в виде модулей. Это позволяет работать с ядром небольшого размера и в то же время обеспечивает поддержку большого количества разнообразных устройств. В частности, в виде модулей могут быть скомпилированы средства для работы с большинством сетевых устройств, поэтому драйверы, включаемые в состав дистрибутивных пакетов, обычно подготавливаются именно в таком виде.

Если администрирование компьютера под управлением Linux является вашей обязанностью, то именно вам предстоит решить, следует ли использовать драйверы сетевых карт, реализованные в виде модулей, или их следует включить в состав ядра. Если вы включите драйвер сетевой карты непосредственно в ядро системы, вам не придется обеспечивать при настройке системы, чтобы перед началом сетевого обмена загружался требуемый модуль. (На самом деле система, поставляемая как дистрибутивный пакет, изначально сконфигурирована так, что данная задача решается корректно.) С другой стороны, если вы администрируете большое количество компьютеров, на которых установлена система Linux, то, возможно, предпочтете создать ядро и набор модулей, которые будете устанавливать на различные компьютеры. В этом случае целесообразно реализовать драйверы в виде модулей.

Программные средства поддержки стека протоколов также могут быть непосредственно включены в ядро или скомпилированы как модули. (Исключением являются протоколы TCP/IP; их можно либо включить в основной файл ядра, либо не использовать вовсе; в виде модулей можно реализовать лишь средства, соответствующие некоторым подопциям опции, управляющей использованием данного стека протоколов.) Так, например, если вы знаете, что каталоги в файловых системах других компьютеров, предоставляемые средствами NFS, будут использоваться лишь непродолжительное время, целесообразно реализовать средства поддержки клиента NFS как отдельный модуль. Поступив таким образом, вы сэкономите часть оперативной памяти в течение времени, когда средства NFS не используются, но монтирование внешних каталогов будет осуществляться несколько дольше, чем это было бы в том случае, если бы фрагмент кода, реализующий клиент NFS, был включен непосредственно в ядро.

Как вы, наверное, поняли, однозначного ответа на вопрос, надо ли включать коды поддержки опций непосредственно в ядро или их следует компилировать как отдельные модули, не существует. Я рекомендую вам сначала выяснить, насколько часто соответствующие средства будут использоваться в процессе работы системы. Если они должны использоваться постоянно, включайте их в основной файл ядра; если же они будут задействованы лишь эпизодически, компилируйте их в виде отдельных модулей. Если размер ядра становится слишком велик и при его загрузке средствами LOADIN (DOS-утилита для загрузки Linux) возникают проблемы, следует отдать предпочтение модульной организации ядра. Возможность загрузки с помощью LOADIN желательно сохранить, так как это поможет справиться с некоторыми проблемами.

Компиляция ядра

После того как вы сконфигурировали ядро системы, выполнив make `xconf`ig или другую команду, приведенную в начале данной главы, вы должны скомпилировать ядро и установить его модули. Для этого необходимо выполнить следующие команды:

```
4 make dep
# make bzImage
# make modules
t make modules_install
```

Первая из этих команд выполняет подготовительную работу. Слово **dep** сокращенно обозначает dependency, соответственно при выполнении команды make dep анализируются выбранные вами опции и определяется, какие исходные файлы зависят от других. Если вы пропустите этот шаг, компиляция будет выполнена некорректно.

В результате выполнения второй команды строится основной файл ядра, который имеет имя **bzImage** и обычно размещается в каталоге **/usr/src/linux/arch/i386/boot**. Существуют различные варианты данной команды. Например, при создании ядра небольшого размера можно использовать команду make zImage (ядро в формате bzImage дает возможность загрузчику, например LILO, обрабатывать ядро большего размера, чем это позволяет **z**Image). Как **z**Image, так и bzImage представляют собой сжатые варианты ядра. Они являются стандартом для компьютеров x86, но на других платформах вам придется вместо make bzImage вызвать команду make **vmlinux**. В результате выполнения данной команды строится несжатое ядро. Каталог, в который помещается основной файл ядра, может отличаться от приведенного выше. Если вы работаете на компьютере, отличном от x86, вместо каталога **i386** будет использован каталог, имя которого соответствует текущей платформе. Так, например, на PowerPC этот каталог имеет имя ppc.

Команда make modules, как нетрудно догадаться, компилирует модули ядра. По команде make **modules_install** файлы, содержащие эти модули, копируются в соответствующие позиции в каталоге /lib/modules. В частности, в каталоге /lib/modules создается каталог, имя которого отражает версию ядра, а в нем, в свою очередь, — подкаталоги для конкретных классов драйверов.

Команды **make** dep, make bzImage (или эквивалентную ей команду) **и** make modules может выполнить любой пользователь, при условии, что он обладает правами чтения и записи данных в каталогах, содержащих исходные коды ядра. Выполнить команду make **modules_install** может только пользователь root.

В зависимости от установленных опций и от быстродействия компьютера, процесс компиляции ядра может занять от нескольких минут до нескольких часов. Как правило, основной файл ядра создается дольше, чем модули, но если число модулей велико, для их создания может потребоваться больше времени, чем для создания ядра. При компиляции на экран монитора выводится большое число сообщений, описывающих ход обработки исходных файлов. Иногда отображаются предупреждающие сообщения, на которые можно не обращать внимание. При появлении сообщения об ошибке компиляция прерывается.

Проблемы, возникающие при компиляции ядра

Если вы корректно установили опции, компиляция ядра, как правило, проходит без проблем, но в некоторых случаях возникают ошибки. Проблемы, встречающиеся при компиляции ядра, описаны ниже.

- **Ошибки в исходном коде или несовместимость кода.** Иногда встречаются драйверы, которые содержат ошибки в исходном коде либо несовместимы с остальными компонентами ядра. Такая ситуация может возникнуть при работе с ядром, находящимся в процессе разработки, либо при попытке включить в состав ядра нестандартный драйвер. В этом случае при компиляции отображается одно или несколько сообщений об ошибке. Чтобы избавиться от ошибок, надо обновить версию ядра или, по крайней мере, заменить драйвер, который стал источником проблем. Если без этого драйвера можно обойтись, лучше вовсе отказаться от его использования.

- **Отсутствие информации о зависимости файлов.** Если работа драйвера зависит от другого драйвера, то первый драйвер должен выбираться лишь после того, как будет выбран второй. В некоторых случаях сценарий, посредством которого выполняется конфигурирование системы, работает некорректно. При компиляции ядра это проявляется следующим образом: каждый файл по отдельности компилируется нормально, но собрать файл ядра не удается. Если драйвер компилируется как отдельный модуль, то при попытке загрузить его отображается сообщение об ошибке. Иногда в сообщении об ошибке содержится информация о том, какие действия надо предпринять, чтобы избавиться от нее. В некоторых случаях решить проблему удается, вызвав команду make dep, а затем повторно скомпилировав ядро. Иногда работоспособное ядро можно получить, отказавшись от включения драйвера непосредственно в основной файл и скомпилировав его в виде отдельного модуля (в некоторых случаях приходится принимать обратное решение, т. е. включать в ядро драйвер, который, будучи подготовленным в виде отдельного модуля, стал источником проблем).

- **Устаревшие объектные файлы.** Если вы компилируете ядро, а затем изменяете конфигурацию и компилируете его повторно, утилита `make` автоматически определяет, какие файлы затрагивают внесенные изменения, и повторно компилирует их. Если при работе данной утилиты возникает сбой, это может привести к ошибке при создании ядра. В одних случаях не удается собрать ядро из готовых компонентов, в других случаях ошибки возникают при компиляции отдельных файлов. Для того чтобы решить эту проблему, надо выполнить команду `make clean`, которая удалит существующие объектные файлы, а затем повторить компиляцию системы.

- **Ошибка компилятора.** GNU C Compiler (GCC) считается надежным компилятором, но бывают случаи, когда он становится источником проблем. Версия GCC, которая входит в состав Red Hat 7.0, не может скомпилировать ядро 2.2.x, но эта проблема устранена в версии ядра 2.4.x. (С Red Hat 7.0 поставляются две версии GCC; для того чтобы компиляция ядра была выполнена успешно, надо вместо дсс использоватькgcc.)

- **Проблемы с использованием аппаратных средств.** GCC более интенсивно использует ресурсы компьютера по сравнению с другими программами, поэтому сбои аппаратных средств чаще проявляются в процессе компиляции ядра системы. Такие ошибки называются *ошибками signal 11*, так как GCC возвращает именно такое сообщение. Причиной подобных ошибок, как правило, являются неисправности в процессоре и оперативной памяти. Дополнительную информацию об этих проблемах и способах их устранения можно найти по адресу `http://www.bitwizard.nl/sig11`.

Если вы не можете самостоятельно разрешить проблемы, возникающие при компиляции ядра, отправьте сообщение в одну из групп новостей, посвященных системе Linux, например в comp.os.linux.misc. Подробно опишите ваш дистрибутивный пакет, укажите версию ядра, которую вы пытаетесь скомпилировать, и сообщения об ошибках. (Сообщения компилятора, не связанные с возникновением ошибок, лучше не указывать.)

Инсталляция нового ядра и его использование

Чтобы готовое ядро можно было использовать, его необходимо инсталлировать. Как было сказано ранее, скомпилированное ядро помещается в каталог /usr/src/linux/arch/i386/boot (вместо i386 может присутствовать другой каталог, имя которого отражает название процессора). Файл ядра надо скопировать или переместить в каталог /boot. Желательно переименовать файл так, чтобы его имя отражало версию ядра и изменения, которые вы внесли в него. Например, вы можете назвать файл ядра bzImage-2.4.17 или bzImage-2.4.17-xfs. Если команда make modules_install до сих пор не выполнялась, надо вызвать ее, инсталлировав тем самым модули ядра в каталог /lib/modules/x.y.z, где x.y.z — это номер версии ядра.

Копирования файла ядра в каталог /boot недостаточно. Чтобы ядро можно было использовать, необходимо также модифицировать загрузчик. Большинство дистрибутивных пакетов содержит Linux Loader (LILO); настройка этого загрузчика на новое ядро осуществляется путем редактирования файла конфигурации /etc/lilo.conf. В листинге 1.1 показано содержимое файла lilo.conf, настроенного на загрузку одного ядра.

Листинг 1.1. Простой файл lilo.conf

```
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
prompt
default=linux
timeout=50
image=/boot/vmlinuz
    label=linux
    root=/dev/sda6
    read-only
```

LILO используется на компьютерах x86. Работая на других компьютерах, вам придется ориентироваться на другие типы загрузчиков. Эти загрузчики во многом напоминают LILO и отличаются от него лишь некоторыми деталями. Описания этих загрузчиков можно найти в документации на конкретные дистрибутивные пакеты.

Для того чтобы при загрузке системы вы могли выбирать между старой версией ядра и вновь созданным ядром, выполните следующие действия.

1. Откройте файл /etc/lilo.conf в текстовом редакторе.

2. Продублируйте строки файла, описывающие текущее ядро, используемое по умолчанию. Первая из этих строк начинается символами `image=`. Описание ядра продолжается до конца файла либо до тех пор, пока не встретится выражение image= или `other=`. В листинге 1.1 ядро описывается в последних четырех строках.

3. Модифицируйте строку `image=` так, чтобы она указывала на новый файл ядра. Например, вместо выражения `image=/boot/vmlinuz` включите выражение `image=/boot/bzImage-2.4.17`. (Во многих дистрибутивных пакетах Linux по умолчанию загружается файл ядра `vmlinuz`.)

4. Измените строку label= для нового ядра, указав после символа = новое значение, например mykernel или 2417. Это значение позволит отличать новое ядро от прежнего. В процессе загрузки вам будет предложено выбрать имя требуемого ядра в меню или ввести его в командной строке.

5. Сохраните внесенные изменения.

6. Введите команду `lilo` для того, чтобы установить модифицированный загрузчик на жесткий диск.

ВНИМАНИЕ Приведенные выше инструкции предполагают, что файл `/etc/lilo.conf` не содержит ошибок. При наличии ошибок выполнение п. 6 может привести к повреждению данных на жестком диске. Включая информацию о новом ядре, не изменяйте и не удаляйте другие данные, содержащиеся в файле.

При следующей загрузке компьютера LILO предложит вам указать, какое ядро должно использоваться при работе системы. В зависимости от настройки, вы сможете либо выбрать ядро в меню, либо указать имя ядра в строке после символов `lilo:`.

Если качество нового ядра устраивает вас, модифицируйте файл `/etc/lilo.conf` так, что это ядро будет загружаться по умолчанию. Для этого надо изменить строку `default=`. Измените текст после знака =, указав имя нового ядра, которое вы присвоили ему в п. 4, а затем в командной строке снова вызовите `lilo`.

Ядро Linux можно загрузить без использования LILO. В некоторых дистрибутивных пакетах вместо LILO содержится загрузчик Grand Unified Boot Loader (GRUB). Особенности конфигурирования GRUB описаны в документации на этот загрузчик. Кроме того, для загрузки Linux также можно использовать DOS-программу LOADIN. Чтобы такая загрузка стала возможной, надо скопировать ядро на диск, доступный из DOS, например, в раздел DOS или на гибкий диск. Для загрузки Linux надо ввести следующую команду:

```
C:> LOADIN BZIMAGE root=/dev/sda6 ro
```

В данном примере BZIMAGE — это имя файла ядра, который доступен из DOS, а `/dev/sda6` — идентификатор корневого раздела, используемый в Linux. Опция `ro` указывает на то, что данный раздел должен монтироваться в режиме, допускающем только чтение (впоследствии Linux повторно смонтирует этот раздел в режиме чтения и записи). LOADIN — удобный инструмент, позволяющий тестировать новые версии ядра, не изменяя настройку LILO. Кроме того, LOADIN дает возможность загрузить Linux, если LILO поврежден. Если на вашем компьютере отсутствует система DOS, вы можете воспользоваться системой FreeDOS (`http://www.freedos.org`), которая также позволяет выполнить данную задачу. Утилита LOADIN входит в состав большинства дистрибутивных версий Linux. На компакт-диске она обычно располагается в каталоге `dosutils`.

Резюме

Ядро Linux непосредственно участвует в выполнении всех операций ввода-вывода, в частности в передаче данных по сети. Если компьютер под управлением Linux планируется подключать к сети, необходимо установить соответствующие опции ядра. Вы можете оптимизировать ядро для выполнения конкретной задачи, включив необходимые опции и отключив средства, которые не используются, а лишь напрасно занимают оперативную память компьютера. Большинство опций, имеющих отношение к взаимодействию по сети, располагаются в двух меню: Networking Options и Network Device Support. Эти меню включают ряд подменю. Установив требуемые опции, надо скомпилировать ядро системы, для чего необходимо выполнить несколько команд. Для того чтобы обеспечить загрузку ядра, следует изменить конфигурацию LILO.

Глава 2

Настройка сетевых средств TCP/IP

Несмотря на то что ядро является главным компонентом системы Linux и помимо выполнения прочих задач контролирует процесс обмена данными по сети, настройка системы для работы в сети не исчерпывается конфигурированием ядра. В данной главе рассматриваются вопросы, имеющие непосредственное отношение к организации сетевого взаимодействия: использование статических IP-адресов, а также применение протоколов DHCP (Dynamic Host Configuration Protocol — протокол динамической настройки узла) и PPP (Point-to-Point Protocol). Протокол DHCP позволяет организовать автоматическое выделение IP-адресов, а протокол PPP обеспечивает соединение по коммутируемой линии. При использовании статических IP-адресов приходится устанавливать конфигурацию соответствующих компонентов системы вручную. Существует большое количество инструментальных средств, упрощающих как автоматическую, так и ручную установку конфигурации системы. Если вы собираетесь выполнять работы по администрированию системы, вам следует ознакомиться с этими инструментами. Однако, перед тем как приступать к обсуждению вопросов, связанных с настройкой системы, необходимо рассмотреть процесс загрузки сетевых драйверов.

Загрузка сетевых драйверов

Первым шагом в настройке сетевых устройств является загрузка соответствующих драйверов. Как было сказано в главе 1, драйверы подготавливаются к работе одним из двух способов: драйвер может быть непосредственно включен в состав ядра Linux либо скомпилирован в виде отдельного модуля. В первом случае загрузка сетевого драйвера не вызывает затруднений. Драйверам некоторых сетевых карт приходится передавать параметры, используя для этого опции загрузки. Если вы применяете LILO, параметры передаются посредством опции append, содержащейся в файле `/etc/lilo.conf`. Например, приведенная ниже строка сообщает ядру о том, что устройство ethO (первая сетевая карта) подключено через порт с номером 0x240.

```
append="ether=0,0,0x240,eth0"
```

После ключевого слова append можно указать несколько значений, поместив их в кавычки и разделив пробелами. Указание порта для конкретного устройства чаще всего используется в системах, содержащих несколько сетевых интерфейсов; в данном примере логическое устройство явным образом связывается с конкретным физическим устройством. В большинстве случаев передавать параметры драйверам, встроенным в ядро, нет необходимости. Драйвер выявляет сетевую карту и обеспечивает доступ к ней без вмешательства администратора.

Если драйвер скомпилирован как отдельный модуль, параметры передаются ему посредством файла /etc/modules .conf (в некоторых системах этот файл имеет имя /etc/conf.modules). Например, данный файл может содержать следующие строки:

```
alias eth0 ne
options ne io=0x240
```

Приведенные выше две строки сообщают системе о том, что для устройства ethO, подключенного через порт ввода-вывода 0x240, должен использоваться драйвер, содержащийся в модуле пе. В большинстве случаев в подобном указании нет необходимости. Оно нужно в основном тогда, когда в системе присутствует несколько сетевых интерфейсов. Инструментальные средства настройки, содержащиеся в составе многих дистрибутивных пакетов, позволяют автоматизировать этот процесс. Вам достаточно выбрать из списка модель сетевой карты и драйвер, после чего требуемые записи будут автоматически включены в файл /etc/modules.conf.

Если вы включили требуемую запись в файл /etc/modules . conf, то при попытке активизировать сетевой интерфейс система Linux автоматически загрузит сетевой драйвер. Если по каким-либо причинам вы хотите сделать это вручную, воспользуйтесь командой insmod.

```
# insmod ne
```

В результате выполнения этой команды модуль пе будет загружен и готов к использованию. Если средства автозагрузки модулей работают ненадежно, вам, возможно, придется включить указанную выше команду в файл /etc/rc.d/rc.local или /etc/rc.d/boot.local.

В некоторых случаях передача данных происходит с помощью протоколов PPP, SLIP или PLIP, а компьютеры соединяются через последовательные или параллельные порты. При этом приходится отдельно загружать драйвер, предназначенный для управления устройством, и драйвер, поддерживающий протокол обмена данными. Такие драйверы подготавливаются так же, как и драйверы сетевых карт: они либо встраиваются непосредственно в ядро, либо компилируются в виде отдельных модулей. В некоторых случаях требуются дополнительные драйверы. Например, для использования модема, подключенного через интерфейс USB, требуются два или три драйвера.

Использование клиента DHCP

Если в вашей локальной сети присутствует сервер DHCP, вы можете сконфигурировать систему Linux так, что компьютер будет автоматически получать у сервера IP-адрес, используя для этого клиентскую программу DHCP. Клиент DHCP ищет сервер DHCP, посылая в широковещательном режиме запрос, который принимают все компьютеры локальной сети. Если сервер отвечает на запрос и последующие переговоры заканчиваются

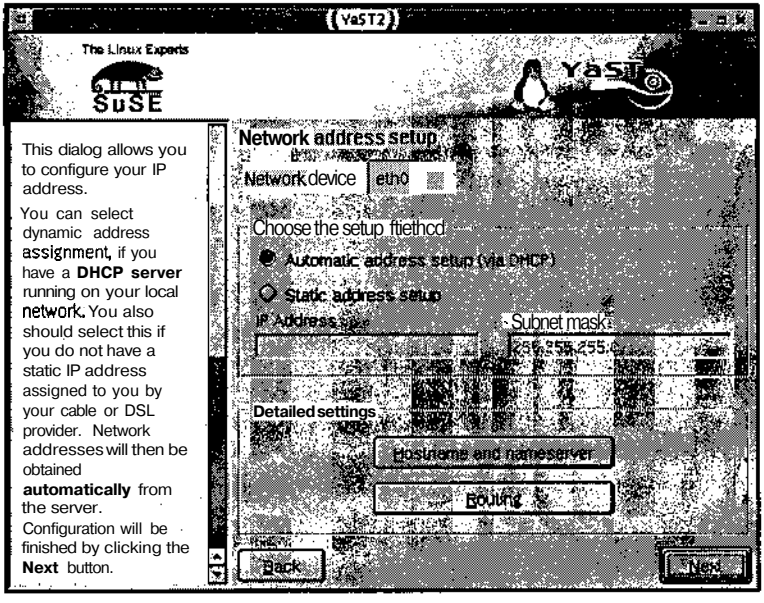


Рис. 2.1. Специальные инструменты с графическим пользовательским интерфейсом упрощают использование клиента DHCP

успешно, то система получает IP-адрес, кроме того, выполняются настройки, необходимые для осуществления сетевого обмена.

Если вы хотите, чтобы ваш компьютер действовал как сервер DHCP, т. е. предоставлял IP-адреса другим системам, вам надо внимательно прочитать главу 5. Серверу DHCP должен быть присвоен статический IP-адрес.

Большинство дистрибутивных пакетов Linux позволяет включать поддержку DHCP в процессе инсталляции системы, в частности, при настройке сетевых средств. Если соответствующая опция отсутствует или если вы хотите изменить конфигурацию системы после ее инсталляции, проще всего сделать это, используя специальный инструмент с графическим пользовательским интерфейсом. К таким инструментальным средствам относятся Linuxconf (Red Hat или Mandrake), COAS (Caldera), YaST и YaST2 (SuSE). На рис. 2.1 показано окно YaST2 с установленной опцией Automatic address setup (via DHCP). В результате установки данной опции система настраивается для получения IP-адресов посредством DHCP.

К сожалению, при настройке средств DHCP иногда возникают проблемы; некоторые из них описаны ниже.

- **Несовместимый клиент DHCP.** В системе Linux применяются четыре клиента DHCP: pump, dhclient, dhcpxd и dhcpcd (обратите внимание на различия между именами последних двух клиентов и именем сервера DHCP dhcpd). В большинстве случаев все четыре клиентские программы работают корректно, но в некоторых сетях могут использоваться серверы DHCP, несовместимые с некоторыми клиентами DHCP, применяемыми в системе Linux. Если возникнет подобная ситуация, вам придется заменить клиент-программу DHCP на другую.

- **Несовместимые опции DHCP.** В некоторых случаях причиной возникновения проблем могут быть опции, передаваемые клиенту DHCP. Несовместимые опции проявляют себя так же, как и несовместимые клиент-программы, но для устранения неисправности в этом случае требуются менее радикальные меры. Вам достаточно отредактировать сценарий, используемый для запуска DHCP, и изменить опции. Чтобы понять, какие опции следует изменять, необходимо тщательно изучить документацию на конкретную клиент-программу DHCP. В этом вам поможет справочная информация о системе.

- **Использование нескольких сетевых карт.** Если в вашем компьютере установлены две или больше сетевых карт (NIC — network interface card), может возникнуть необходимость использовать клиент DHCP для получения IP-адресов лишь для некоторых из этих карт. Возможно, вы захотите, чтобы для каких-либо карт часть информации (например, адрес шлюза) не принималась во внимание. В этом случае вам также придется отредактировать сценарий запуска DHCP, либо написать собственный сценарий, который изменял бы автоматически выбранную конфигурацию.

В табл. 2.1 для наиболее популярных дистрибутивных пакетов Linux представлены клиент DHCP, используемый по умолчанию, альтернативный клиент DHCP, расположение сценария запуска, а также расположение основных конфигурационных файлов DHCP. (Инструмент `ifup` для Debian, в отличие от одноименных файлов, используемых другими системами, представляет собой программу, в которой реализованы средства настройки клиента DHCP. Управлять работой программы `ifup` можно, изменяя содержимое конфигурационного файла `/etc/network/interfaces`.) Если клиент DHCP, с которым вы предпочитаете работать, отсутствует в дистрибутивном пакете, вы все равно можете установить и использовать его. Возможно, вам придется внести некоторые изменения в сценарий запуска, расположение которого приведено в табл. 2.1, или самостоятельно реализовать процедуру запуска клиента DHCP.

Если вы считаете, что источником проблем являются опции клиента DHCP, несовместимые с присутствующим в сети сервером DHCP, то для решения этих проблем вам надо отредактировать сценарий запуска. Найдите строку, отвечающую за запуск клиент-программы, и проанализируйте передаваемые ей опции. В этом вам помогут страницы справочной информации, посвященные клиенту DHCP. Удаляя или добавляя опции, постарайтесь добиться желаемого поведения программы. Например, некоторые серверы DHCP требуют, чтобы клиент передавал имя узла; если вы используете программу dhcpcd, вам придется добавить опцию -h *имя_узла*. Часто в сценариях используются данные из конфигурационного файла (их расположение также приведено в табл. 2.1), однако чаще всего эти файлы сообщают системе, следует ли использовать статические IP-адреса или надо воспользоваться DHCP.

Использование статических IP-адресов

Несмотря на то что система DHCP используется во многих сетях, в ряде случаев приходится выделять IP-адреса другими способами. Некоторым компьютерам (например, на которых выполняются серверы DHCP) чрезвычайно трудно присваивать адреса с помощью DHCP. Кроме того, сервер DHCP попросту может отсутствовать в сети. В подобных случаях приходится распределять IP-адреса вручную. Средства для решения данной зада-

Таблица 2.1. Информация о клиентах DHCP для наиболее популярных дистрибутивных пакетов Linux

Версия Linux	Клиент DHCP по умолчанию	Альтернативный клиент DHCP	Сценарий запуска клиента DHCP	Дополнительные конфигурационные файлы
Caldera OpenLinux Server 3.1	dhclient	Отсутствует	/etc/sysconfig/network-scripts/ifup-dhcp	/etc/sysconfig/network, /etc/sysconfig/network-scripts/ifcfg-eth0, /etc/dhcp/dhclient.conf
Debian GNU/Linux 2.2	pump	dhcpcd	/sbin/ifup (двоичный файл)	/etc/network/interfaces
Linux Mandrake 8.1	dhcpcd	dhclient, dhcpxd	/sbin/ifup	/etc/sysconfig/network, /etc/sysconfig/network-scripts/ifcfg-eth0
Red Hat Linux 7.2	pump	dhcpcd	/sbin/ifup	/etc/sysconfig/network, /etc/sysconfig/network-scripts/ifcfg-eth0
Slackware Linux 8.0	dhcpcd	Отсутствует	/etc/rc.d/rc.inet1	Отсутствуют
SuSE Linux 7.3	dhcpcd	dhclient	/etc/init.d/dhclient	/etc/rc.config
TurboLinux 7	dhclient	Отсутствует	/sbin/ifup	/etc/sysconfig/network, /etc/sysconfig/network-scripts/ifcfg-eth0

чи рассматриваются в данном разделе. Кроме того, далее в этой главе рассказывается, как настроить систему, чтобы ее конфигурация автоматически устанавливалась при загрузке.

Как правило, компьютерам, на которых выполняются программы-серверы, присваивают статические IP-адреса; при этом адрес не изменяется с течением времени. Кроме того, связывание статических IP-адресов с доменными именами не вызывает трудностей. (Вопросы функционирования серверов DNS и установления соответствия между IP-адресами и доменными именами рассматриваются в главе 18.) Чтобы связать доменное имя с динамическим IP-адресом, вам надо обеспечить, чтобы сервер DHCP выделял компьютеру один и тот же адрес (как это сделать, вы узнаете в главе 5), либо использовать динамические средства DNS.

Настройка сетевых интерфейсов

Загрузка драйвера — это лишь первое действие, которое надо выполнить, чтобы обеспечить доступ к сетевому интерфейсу. Для того чтобы интерфейс можно было использовать, ему необходимо присвоить IP-адрес и выполнить дополнительные настройки, например задать маску подсети. Для решения этой задачи используется утилита ifconfig, которая, в зависимости от способа ее вызова, либо отображает информацию об интерфейсе, либо изменяет его конфигурацию.

Использование ifconfig

Синтаксис `ifconfig` достаточно прост. Для вызова данной утилиты надо задать в командной строке следующее выражение:

`ifconfig [`*`интерфейс`*`] `*`[опции]`*

Набор передаваемых параметров определяет поведение `ifconfig`. Данная утилита может выполнять следующие действия.

- Если `ifconfig` вызывается без параметров, она возвращает информацию о состоянии всех активных сетевых интерфейсов, т. е. действует как инструмент диагностики.
- Если данной утилите передано только имя интерфейса (например, `eth0` или `tr1`), то она возвращает информацию лишь о состоянии этого интерфейса.
- Если помимо имени интерфейса заданы некоторые опции, `ifconfig` модифицирует данный интерфейс в соответствии со значениями переданных опций. Чаще всего с помощью данной утилиты интерфейс активизируется либо переводится из активного в неактивное состояние.

Если вы собираетесь использовать `ifconfig` для настройки интерфейса, вам необходимо изучить назначение опций, которые передаются данной утилите. Список опций, которые приведены на страницах справочной системы, посвященных `ifconfig`, очень велик. Наиболее важные из них описаны ниже.

- up *адрес*. Данная опция активизирует интерфейс и связывает с новым интерфейсом указанный IP-адрес. Если в составе команды не указана маска подсети, используется маска, определяемая исходя из класса адреса (классы IP-адресов описаны в табл. 2.2). В большинстве случаев ключевое слово up можно не указывать;

Таблица 2.2. Классы IP-адресов и соответствующие им маски подсети

Класс	Диапазон адресов	Адреса, предназначенные для внутреннего использования	Маска подсети
Class A	1.0.0.0-127.255.255.255	10.0.0.0-10.255.255.255	255.0.0.0
Class B	128.0.0.0-191.255.255.255	172.16.0.0-172.31.255.255	255.255.0.0
Class C	192.0.0.0-223.255.255.255	192.168.0.0-192.168.255.255	255.255.255.0

если при вызове `ifconfig` заданы имя интерфейса и IP-адрес, оно предполагается по умолчанию.

- down. Эта опция противоположна опции up, т. е. она делает интерфейс неактивным ("закрывает" его).
- `netmask` *nm*. Данная опция устанавливает маску подсети для интерфейса. Маска подсети определяет, какое число битов в составе IP-адреса выделяется для представления адреса сети; остальные биты адреса идентифицируют компьютер в составе сети. Если данная опция не указана, по умолчанию принимается маска подсети, определяемая на основании адреса (табл. 2.2). Маску подсети можно также задать с помощью опции up *адрес* как число бит, соответствующих адресу сети.
- `[-]promisc`. По умолчанию сетевая карта принимает только те пакеты, которые непосредственно адресованы ей. Данная опция включает (promise) или отключает (`-promisc`) так называемый *режим сбора пакетов,* или *резким прослушивания* (promiscuous mode), в котором карта принимает все пакеты, передаваемые по сети. Режим сбора пакетов применяется для диагностики сети. (Этот режим часто используют хакеры для перехвата паролей, передаваемых в незакодированном виде.) Некоторые программы также могут включать режим сбора данных.
- mtu *n*. Данная опция устанавливает значение MTU (Maximim Transfer Unit — максимальный размер передаваемого блока), т. е. максимальный размер пакета нижнего уровня. Для сетей Ethernet значение MTU обычно принимается равным 1500, но при необходимости вы можете изменить его. (Ряд маршрутизаторов использует меньшее значение MTU, кроме того, некоторые протоколы накладывают ограничения на величину MTU. Если установленный в системе максимальный размер пакета превышает предельно допустимое значение для сети, это приводит к снижению производительности, так как перед передачей пакет разбивается на кадры меньшего размера.)
- add *адрес/длина_префикса*. Данная опция выполняет те же действия, что и опции up и netmask, но она ориентирована на протокол IPv6. (Протокол IPv6 представляет собой новый стандарт обмена данными в Internet.) Как было сказано в главе 1, IPv6 поддерживает значительно больше адресов, чем IPv4. На момент написания данной книги, т. е. в 2002 г., этот протокол еще использовался очень редко.
- del *адрес/длина_префикса*. Эта опция противоположна опции add, т. е. она отменяет IPv6-адрес, присвоенный ранее интерфейсу.

- media *тип*. Некоторые сетевые карты допускают подключение нескольких разъемов (например, 10Base-2 и 10Base-T). Данная опция позволяет определить, какой разъем должен использоваться (например, media 10Base-T). Подробную информацию о поддерживаемых типах разъемов можно найти в описании конкретного драйвера.

- hw *класс адрес*. Данная опция позволяет задавать аппаратный адрес сетевой карты. Если вам потребовалось заменить сетевую карту, но вы хотите, чтобы сервер DHCP продолжал выделять тот же IP-адрес, вам надо воспользоваться данной опцией и задать для новой карты аппаратный адрес, использовавшийся ранее. Бывают также случаи, когда разные производители выпускают карты с одинаковыми адресами. Такие устройства нельзя использовать в рамках одной локальной сети; в этом случае опция hw также может оказаться полезной. Данная опция предполагает два значения: класс сетевого устройства (например, ether для Ethernet или ARCnet для ARCnet) и аппаратный адрес. Заметьте, что на некоторые сетевые карты данная опция не оказывает влияния.

- txqueulen *длина*. Эта опция задает длину очереди, т. е. число пакетов, ожидающих передачи через определенный интерфейс. По умолчанию принимается значение 100, что в большинстве случаев обеспечивает нормальную работу сети. Уменьшая длину очереди, можно несколько увеличить скорость обмена посредством таких протоколов, как Telnet и SSH.

В большинстве случаев выполнение команды ifconfig обеспечивает активизацию интерфейса. Ниже приведен пример команды, которая активизирует Ethernet-карту и присваивает ей адрес 172.23.45.67.

```
# ifconfig ethO 172.23.45.67
```

Добавляя дополнительные параметры, можно уточнить конфигурацию интерфейса.

```
# ifconfig ethO 172.23.45.67 netmask 255.255.255.0 mtu 1420
```

Как было сказано выше, маска подсети определяет, какая часть IP-адреса должна представлять адрес сети, а какая — адрес компьютера в этой сети. Компьютер использует эту информацию для определения адресов назначения исходящих пакетов; если установить маску подсети неправильно, некоторые компьютеры будут не доступны. Если представить маску подсети в двоичном виде, нетрудно заметить, что она начинается последовательностью единиц, за которой следует последовательность нулей. Например, маска 255.255.255.0 состоит из 24 единиц и восьми нулей. Вместо указания маски можно задать в составе адреса число битов, используемых как адрес сети. Информация о числе битов, представляющих адрес подсети, отделяется от основной части IP-адреса косой чертой. Например, 172.23.45.67/24 соответствует адресу 172.23.45.67 и маске подсети 255.255.255.0. Такое выражение можно использовать при вызове утилиты ifconfig в составе опции up *адрес; в* этом случае опцию netmask *nm* можно не указывать.

Классы IP-адресов

В качестве примеров IP-адресов в данной книге используются зарезервированные адреса, предназначенные для организации работы внутренних сетей. Сделано это для того, чтобы неопытные читатели случайно не использовали адреса существующих узлов глобальной сети. Для внутренних сетей зарезервированы адреса 192.168.x.x (класс C), 172.16.0.0-172.31.255.255 (класс B) и 10.x.x.x (класс A). Узлы с такими адресами гарантированно отсутствуют в Internet.

В дополнение к классам A, B и C, описанным в табл. 2.2, существуют также классы адресов D и E. Адреса класса D применяются для группового вещания (передаваемый пакет адресуется сразу нескольким узлам), а адреса класса E зарезервированы для дальнейшего использования.

В табл. 2.2 приведены маски подсетей для различных классов адресов. G начала 1990-х этот стандарт стал претерпевать некоторые изменения. Дело в том, что, согласно традиционной схеме распределения IP-адресов, предусмотрено слишком много сетей класса A, каждая из которых может насчитывать больше десяти миллионов компьютеров, в то время как число сетей класса C оказывается недостаточным. Спецификация CIDR (Classless Inter-Domain Routing — бесклассовая междоменная маршрутизация) позволяет задавать произвольные диапазоны IP-адресов, используя для этого маски подсетей. Так, например, организации, которой требуются две сети класса C, могут быть предоставлены адреса 10.34.56.0/24 и 10.34.57.0/24. Благодаря такому принципу распределения адреса используются гораздо эффективнее, чем это позволяет сделать традиционная схема, предусматривающая классы A, B и C. Однако при этом администраторы сетей и пользователи должны внимательно следить за назначением масок подсетей. Если, например, вы предоставите утилите `ifconfig` возможность самостоятельно назначить маску подсети для компьютера 10.34,56.78, то по умолчанию будет использована маска 255.0.0.0 и маршрутизация будет выполняться некорректно. Очевидно, что для сети 10.34.56.0/24 маска подсети должна иметь значение 255.255.255.0.

Настройка нескольких сетевых интерфейсов

Если компьютер содержит несколько сетевых интерфейсов, утилиту `ifconfig` надо вызвать для каждого из интерфейсов. Рассмотрим следующие команды:

```
# ifconfig eth0 up 192.168.1.1
# ifconfig eth1 up 172.23.45.67/24
```

В результате их выполнения с интерфейсом eth0 связывается адрес 192.168.1.1, а с интерфейсом `eth1` — адрес 172.23.45.67; для `eth1` будет использоваться маска подсети 255.255.255.0. Оба интерфейса работоспособны. Но как определить, через какой интерфейс следует передавать тот или иной пакет? Предположим, что прикладная программа, выполняющаяся на этом компьютере, должна установить соединение с узлом, имеющим адрес 10.9.8.7. Как узнать, на какой интерфейс надо передать пакет, предназначенный этому узлу? Для решения этой задачи (задачи маршрутизации) используются таблицы маршрутизации. Как вы вскоре увидите, задачу маршрутизации приходится решать, даже если на компьютере присутствует лишь один сетевой интерфейс.

Заполнение таблицы маршрутизации

Таблица маршрутизации выполняет две задачи. Во-первых, она сообщает системе, на какой из интерфейсов следует передавать информационные пакеты. На первый взгляд может показаться, что если на компьютере установлен лишь один сетевой интерфейс, то ответ на этот вопрос очевиден. На самом деле это не так. Дело в том, что на каждом из компьютеров, работающих под управлением системы Linux, поддерживается интерфейс обратной петли. Этот интерфейс соответствует сети 127.0.0.0/8, но реально при работе с ним используется лишь один IP-адрес 127.0.0.1. Поскольку этот интерфейс присутствует на всех компьютерах, многие программы используют его для взаимодействия с другими локальными программами. При этом обеспечивается более высокая скорость обмена, чем при использовании традиционных сетевых интерфейсов. Для того чтобы распределять трафик между интерфейсом локальной петли и обычными сетевыми интерфейсами, существуют специальные правила. Вторая задача, которую выполняет таблица маршрутизации, состоит в управлении трафиком, предназначенным для компьютеров в локальной сети. Для маршрутизации в локальной сети используется протокол ARP (Address Resolution Protocol — протокол преобразования адресов). Пакеты, предназначенные узлам локальной сети, непосредственно передаются соответствующим компьютерам, а пакеты, адресованные удаленным узлам, передаются посредством маршрутизатора, или шлюза. В большинстве случаев в таблице маршрутизации Linux указывается лишь один шлюз, но встречаются также более сложные конфигурации с несколькими шлюзами. Для заполнения таблицы маршрутизации используется команда route.

В Internet на пути от одного компьютера к другому может находиться большое число маршрутизаторов, но каждый компьютер должен знать адрес лишь одного маршрутизатора. Получив пакет, который должен быть передан по определенному адресу, маршрутизатор определяет адрес следующего маршрутизатора; этот процесс повторяется до тех пор, пока пакет не прибудет по назначению.

Структура таблицы маршрутизации

Таблица маршрутизации содержит набор записей, которые определяют, как должны обрабатываться пакеты, в зависимости от адреса их назначения. Когда программа передает пакет, предназначенный для передачи ядру, последнее сравнивает адрес назначения с адресами или диапазонами адресов, указанными в записях таблицы, начиная с наиболее конкретных адресов, т. е. с диапазона, определяющего сеть наименьшего размера. Если адрес назначения пакета соответствует очередному адресу или диапазону, для передачи пакета используется правило, указанное в таблице маршрутизации, в противном случае сравнение продолжается. Самое универсальное из правил носит название маршрута по умолчанию, оно определяет любой адрес Internet. Маршрут по умолчанию обычно направляет пакет через шлюз локальной сети.

Для того чтобы лучше понять, как используется таблица маршрутизации, рассмотрим пример такой таблицы. На рис. 2.2 показана таблица маршрутизации, которая отображается в результате выполнения команды route -n (более подробно команда route будет рассмотрена в следующем разделе). Записи таблицы, изображенной на рисунке, упорядочены так, что в начале расположены записи, определяющие наиболее конкретные правила обработки, а в конце таблицы находятся наиболее универсальные правила. В первой записи указан адрес назначения 255.255.255.255, т. е. широковещательный адрес. Широко-

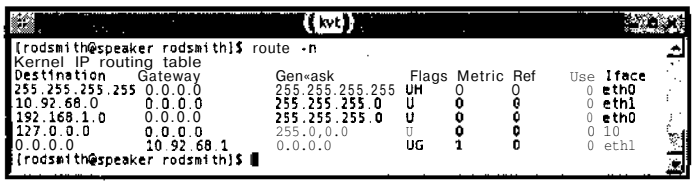


Рис. 2.2. Для того чтобы определить маршрут пакета, надо сравнить его адрес назначения с адресом, указанным в столбце Destination, и учесть при этом маску подсети, значение которой отображается в столбце Genmask

вещательные пакеты передаются через интерфейс ethO, при этом шлюз не используется. В последующих двух записях содержатся адреса назначения 10.92.68.0 и 192.168.1.0, которые представляют собой адреса локальных сетей; им соответствует маска подсети 255.255.255.0, которая указана в столбце Genmask. Эти две записи направляют трафик соответственно через интерфейсы eth1 и ethO. Если компьютер содержит только один сетевой интерфейс, в таблице маршрутизации будет указана лишь одна подобная запись. Четвертая запись соответствует интерфейсу обратной петли (в некоторых разновидностях Linux, например в системе Debian, при выводе таблицы маршрутизации этот маршрут не отображается, но он учитывается при обработке пакетов). Обратите внимание, что этот интерфейс имеет имя 1o (оно содержится в столбце Iface таблицы). Последняя запись, в которой указан адрес назначения 0.0.0.0, определяет маршрут по умолчанию. Этот адрес вместе с маской подсети 0.0.0.0 соответствует любому адресу, при сравнении которого с адресами, указанными в предыдущих правилах, был получен отрицательный результат. В этом случае трафик направляется через интерфейс eth1. Маршрут по умолчанию — единственный маршрут в таблице, для которого был указан шлюз (в данном случае 10.92.68.1).

При активизации интерфейса с помощью ifconfig эта утилита автоматически включает в таблицу маршрутизации запись, соответствующую активизированному интерфейсу. Эта запись определяет маршрут к сети, которая подключена через данный интерфейс. Сценарий, выполняющихся при загрузке Linux, добавляет в таблицу запись для интерфейса обратной петли. Запись, соответствующая широковещательному маршруту, не обязательна, но используется некоторыми утилитами. Во многих случаях единственной записью, которую приходится создавать вручную, остается маршрут по умолчанию.

Использование route

Если утилита route вызывается без параметров, она отображает текущее содержимое таблицы маршрутизации. Такой же результат будет получен при указании некоторых опций (например, опции -n, которая указывает на то, что при выводе содержимого таблицы вместо доменных имен должны отображаться числовые IP-адреса). Однако в основном route предназначена для добавления, удаления и изменения записей о маршрутах. Синтаксис route имеет следующий вид:

```
route add | del [-net | -host] target [netmask nm] [gateway gw]
[metric m] [mss m] [window W] [[dev] interface]
```

Ниже перечислены опции данной утилиты и описано их назначение.

- add I del. Опция add задается тогда, когда необходимо добавить в таблицу запись о новом маршруте, а опция del позволяет удалить существующую запись. При добавлении нового маршрута необходимо задать дополнительную информацию. При удалении можно ограничиться указанием адреса назначения.

- [-net I -host]. В качестве адреса назначения вы можете задать либо адрес сети (-net), либо адрес конкретного компьютера (-host). В большинстве случаев route способна самостоятельно отличить адрес сети от адреса узла, но иногда необходимо явно указать тип адреса. Чаще всего данную опцию приходится задавать, определяя маршрут к небольшой сети, подключенной с помощью отдельного шлюза.

- *адрес_назначения*. Адрес назначения принадлежит сети или отдельному компьютеру, которому маршрутизатор должен передать пакет. Для маршрута по умолчанию используется адрес 0.0.0.0 либо эквивалентное ему ключевое слово default. Этот параметр необходимо указывать при добавлении или удалении маршрута.

- [netmask *nm*]. Если адреса сети, которой должны быть переданы пакеты, соответствуют традиционной схеме распределения адресов, утилита route, пользуясь сетевыми средствами Linux, сама определит значение маски подсети. В противном случае вам необходимо явно задать маску подсети, указав при вызове route параметр netmask *nm*. (Вместо использования данного параметра вы можете указать число бит, выделяемых для представления адреса сети, в составе адреса назначения.)

- [gateway *gw*]. Если вы определяете маршрут, который не проходит через шлюз, можете не указывать этот параметр. Если же целевой узел подключен через шлюз, необходимо задать адрес этого шлюза, указав при вызове route gateway *gw*. В частности, данный параметр используется при определении маршрута по умолчанию.

- [metric *m*]. На рис. 2.2 среди прочих изображен столбец Metric. В нем отображается *метрика* маршрута, т. е. "стоимость" передачи пакета. Чаще всего за "стоимость" принимается время передачи пакета. Таким образом, маршрутам, на которых встречаются линии с низким быстродействием, соответствуют высокие значения метрики, а "быстрым" маршрутам — низкие значения метрики. Параметр metric *m* используется только в том случае, если компьютер выполняет роль маршрутизатора. Подробно вопросы настройки маршрутизаторов будут рассмотрены в главе 24.

- [mss *m*]. Параметр mss *m* задает максимальный размер сегмента (MSS — Maximum Segment Size). Подобно metric *m,* данный параметр используется в основном в маршрутизаторах.

- [window *W*]. Размер окна (TCP Window Size) — это объем данных, которые могут быть переданы передающим узлом, не дожидаясь получения подтверждения с принимающего узла. Если задано небольшое значение данного параметра, скорость обмена данными уменьшится, так как передающий компьютер будет простаивать,

ожидая подтверждения приема пакета. Если указать слишком большой размер окна, повышается вероятность того, что вследствие возникновения ошибки передающему узлу придется повторять передачу большого объема информации. Поэтому наилучшее решение — использовать размер окна по умолчанию (в системе Linux он составляет 64 Кбайт). Если данные по линии передаются быстро, но с большой задержкой (например, если используется спутниковая связь), то целесообразно увеличить размер окна до 128 Кбайт.

- [[dev] *имя_интерфейса*]. Как правило, система Linux по IP-адресу самостоятельно определяет используемый интерфейс. Однако в некоторых случаях необходимо указать интерфейс явно, задавая при вызове route параметр [dev] *имя_интерфейса*. (Ключевое слово dev указывать не обязательно, достаточно задать имя интерфейса, например eth0 или tr1.)

Наиболее часто с помощью утилиты route задается маршрут по умолчанию. Делается это после того, как посредством утилиты ifconfig был активизирован сетевой интерфейс. Пример определения маршрута по умолчанию с помощью route приведен ниже.

```
# route add 0.0.0.0 gw 10.92.68.1
```

Адрес 0.0.0.0 можно заменить ключевым словом default; результат выполнения команды от этого не изменится. Несколько реже при вызове route приходится указывать имя устройства, опцию -net и некоторые другие опции.

Использование нескольких интерфейсов и одного шлюза

Как было сказано ранее, при каждой активизации интерфейса посредством ifconfig данная утилита автоматически включает в таблицу маршрутизации запись для нового интерфейса. Однако при этом не добавляется информация о шлюзах. Поэтому настройка большинства компьютеров, содержащих несколько интерфейсов, включает следующие действия.

- Вызов ifconfig для каждого из интерфейсов компьютера.
- Одиночный вызов route для добавления в таблицу маршрутизации маршрута по умолчанию.

Эти действия типичны для компьютеров под управлением Linux, которые выполняют функции маршрутизаторов для сетей небольших отделов. Для того чтобы компьютер действовал как маршрутизатор, необходимо разрешить перенаправление IP-пакетов. Сделать это можно, выполнив Следующую команду:

```
# echo "1" > /proc/sys/net/ipv4/ip_forward
```

Если компьютер содержит два сетевых интерфейса (т. е. одновременно принадлежит двум сетям), но не должен выполнять функции маршрутизатора, не следует разрешать перенаправление IP-пакетов.

Маршрутизатор не обязательно должен быть выделенным. Компьютер, выполняющий функции маршрутизатора, может одновременно решать другие задачи. Однако при этом необходимо учитывать, что действия, не связанные с маршрутизацией пакетов, занимают время процессора и создают дополнительную нагрузку на сетевые интерфейсы, в результате производительность маршрутизатора снижается, что может привести к уменьшению пропускной способности всей сети. Кроме того, подобное совмещение функций может создавать угрозу безопасности сети. В настоящее время маршрутизаторы выполняют также функции брандмауэров, и работа дополнительных программных продуктов на таком компьютере может открывать дополнительные возможности для атак, предпринимаемых злоумышленниками.

Если провайдер выделил для вашего компьютера лишь один IP-адрес, но вы хотите организовать доступ к Internet с нескольких компьютеров, подключенных к локальной сети, вам необходимо использовать специальный тип маршрутизатора, в котором используется технология NAT (Network Address Translation — преобразование сетевых адресов). Эта технология подробно описана в главе 25. Настройка системы NAT выполняется подобно настройке обычного маршрутизатора, кроме того, в этом случае приходится выполнять дополнительные команды, разрешающие преобразование адресов. В результате такого преобразования вся локальная сеть выглядит извне как один компьютер.

Использование нескольких интерфейсов и шлюзов

Если компьютер с несколькими интерфейсами должен передавать пакеты на различные шлюзы, его настройка несколько усложняется. Большинство систем работает с одним шлюзом, через который проходит маршрут по умолчанию. Такой шлюз соединяет локальную сеть с другой сетью, и в большинстве случаев посредством этого же шлюза осуществляется взаимодействие с Internet. Однако возможны и другие варианты конфигурации сети. Рассмотрим локальные сети, представленные на рис. 2.3. Как видно на рисунке, две локальные сети, принадлежащие различным подразделениям одной организации, соединены с помощью маршрутизаторов. Конфигурация обычных компьютеров, принадлежащих этим сетям, очень проста; в маршруте по умолчанию в качестве адреса шлюза указан адрес маршрутизатора, через который локальная сеть подключена к другой сети. Несмотря на то что маршрутизатор сети Office 2 имеет два интерфейса, в маршруте по умолчанию, заданном в его таблице маршрутизации, роль шлюза играет маршрутизатор сети Office 1. Маршрутизатор сети Office 1 имеет более сложную конфигурацию. Его маршрут по умолчанию обеспечивает обмен пакетами с Internet, кроме того, трафик, предназначенный для сети 172.20.0.0/16, должен передаваться на маршрутизатор Office 2. Чтобы такая передача пакетов могла выполняться, необходимо вызвать следующую команду:

```
# route add -net 172.20.0.0 netmask 255.255.0.0 gw 172.21.1.1
```

Структура, показанная на рис. 2.3, имеет смысл только в том случае, если сети Office 1 и Office 2 расположены далеко друг от друга и для их взаимодействия используется один из протоколов поддержки удаленного соединения. Если же подразделения находятся рядом, например в одном здании, целесообразно подключить обе сети к одному концентратору или коммутатору. При этом обе сети могут обслуживаться одним маршрутизатором.

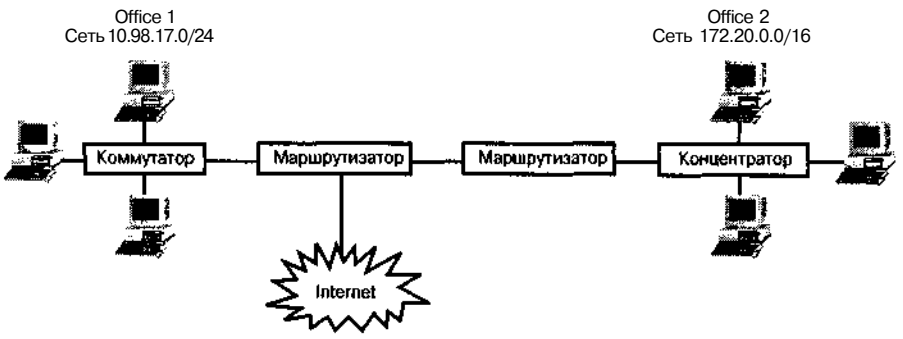


Рис. 2.3. Чтобы маршрутизатор, содержащий больше двух интерфейсов, работал корректно, для него должны быть определены как минимум два шлюза

В данном случае предполагается, что маршрутизатор Office 2 использует для соединения с маршрутизатором Office 1 сетевой интерфейс с адресом 172.21.1.1. Заметьте, что этот адрес не принадлежит сети Office 2 (все компьютеры сети Office 2 соединены с маршрутизатором Office 2 через один интерфейс, а маршрутизатор Office 1 подключен к нему через другой интерфейс). Если кроме приведенной выше команды для маршрутизатора Office 1 также задать с помощью утилиты `route` маршрут по умолчанию, то в результате в таблице маршрутизации будут определены два шлюза: один в качестве маршрута по умолчанию, а другой — для управления трафиком, предназначенным для сети Office 2. Заметьте, что остальные компьютеры в сети Office 1 не обязаны знать об особенностях настройки маршрутизатора, в них должна содержаться лишь информация о маршруте по умолчанию, в котором роль шлюза выполняет маршрутизатор этой сети.

Подобная конфигурация маршрутизатора может потребоваться и в других случаях. Предположим, что в сети Office 1 присутствует второй маршрутизатор, посредством которого локальная сеть подключается к Internet. При этом для каждого компьютера сети Office 1 должны быть определены два шлюза: шлюз по умолчанию, т. е. компьютер, посредством которого осуществляется соединение сети с Internet, и второй шлюз, через который походит маршрут к компьютерам сети Office 2. (Компьютеры сети Office 1 могут быть сконфигурированы и по-другому, для них может быть определен только шлюз по умолчанию, который, в свою очередь, будет передавать пакеты на второй шлюз. Как нетрудно заметить, использование такой конфигурации увеличивает трафик локальной сети.) Поскольку использование двух маршрутизаторов затрудняет настройку компьютеров, желательно использовать в сети один маршрутизатор.

Настройка DNS

После активизации интерфейсов и установки маршрутов компьютер может обмениваться пакетами как с компьютерами локальной сети, так и с любыми другими компьютерами, с которыми он соединен системой шлюзов. Для указания адреса назначения пакета используются IP-адреса. Такая адресация естественна для маршрутизаторов, но чрезвычайно неудобна для пользователей. Преобразование символьных имен (например, `www.awl.com`) в IP-адреса, используемые при маршрутизации пакетов, осуществляет

система доменных имен (DNS — Domain Name System). Кроме того, DNS может также осуществлять обратное преобразование.

DNS поддерживает глобальную распределенную базу данных, для работы с которой используется большое количество серверов. Для того чтобы пользоваться этой базой, компьютер должен знать адрес лишь одного сервера DNS. Большинство организаций и провайдеров Internet устанавливают у себя один или несколько серверов. Чтобы узнать адрес такого сервера, надо обратиться к администратору сети. Получив эти сведения, надо включить их в файл `/etc/resolv.conf`. В данном файле может содержаться до трех строк, начинающихся с ключевого слова `nameserver`, за которым следует IP-адрес сервера DNS. В этом файле также указывается домен по умолчанию (для этого используется ключевое слово `domain`) и произвольное число доменов, в которых выполняется поиск имени. Поиск проводится в том случае, если указано лишь имя компьютера, а имя домена пропущено (например, если вместо `mail.threeroomco.com` пользователь задал имя `mail`). Пример файла `/etc/resolv.conf`, содержащего все три типа записей, приведен в листинге 2.1.

Листинг 2.1. Пример файла `/etc/resolv.conf`

```
domain threeroomco.com
search tworoomco.com fourroomco.com
nameserver 10.98.17.34
nameserver 172.20.13.109
```

ВНИМАНИЕ

Несмотря на то что запись `search` позволяет сэкономить время при вводе доменного имени, желательно воздержаться от ее использования. Предположим, что в обоих доменах, указанных в листинге 2.1 (`tworoomco.com` и `fourroomco.com`), содержится компьютер с именем `www`. Если, работая на компьютере, на котором находится приведенный выше файл `/etc/resolv.conf`, пользователь введет имя www, он может получить документ, содержащийся на сервере одного домена, и считать при этом, что он работает с другим доменом. Кроме того, при поиске затрачивается время, в течение которого обработка других запросов на преобразование адресов замедляется. Более того, даже если вы зададите полное имя, система сначала попытается найти его в доменах, определенных посредством записей `domain` и `search`. Например, если на компьютере, на котором находится рассматриваемый файл `/etc/resolv.conf`, вы зададите имя `www.awl.com`, то сначала будет предпринята попытка найти имена `www.awl.com.threeroomco.com`, `www.awl.com.tworoomco.com` и `www.awl.com.fourroomco.com` и лишь затем начнется обработка имени `www.awl.com`. Успехом увенчается лишь попытка преобразования имени, в которое после домена com будет стоять точка.

После того как вы отредактируете файл `/etc/resolv.conf` в соответствии со своими потребностями, можете начинать работу в сети. Для активизации внесенных изменений не требуются никакие дополнительные команды. Linux автоматически начнет работать с указанным сервером имен и выполнять поиск в указанных доменах.

Если вы хотите, чтобы ваш компьютер под управлением Linux выполнял функции сервера DNS, вам надо выполнить специальные настройки. Сделать это поможет мате-

риал, изложенный в главе 18. В главе 18 приводятся сведения об особенностях работы сервера имен. В зависимости от конфигурации, к этому серверу могут обращаться как компьютеры, находящиеся в той же локальной сети, так и другие узлы Internet.

Определение имени узла

При использовании многих протоколов семейства TCP/IP необходимо, чтобы к компьютеру можно было обращаться по имени. Для того чтобы упростить настройку отдельных программ, в Linux содержится специальная утилита hostnane, позволяющая определить имя узла. Если вызвать эту утилиту без параметров, она выведет текущее имя узла. Если за именем утилиты следует имя узла (например, `hostnane larch.threeroomco.com`), это имя присваивается узлу. Имя узла можно хранить в файле и с помощью опции -f или `-file` передавать `hostnane` имя того файла, например `hostnane -f /etc/HOSTNAME`. В большинстве дистрибутивных пакетов предусмотрена автоматическая установка имени узла при загрузке системы, но имя узла в различных системах хранится в разных файлах. Это может быть файл `/etc/hostnane`, /etc/HOSTNAME или файл, указанный в составе дополнительного конфигурационного файла (см. табл. 2.1).

Имя узла должно устанавливаться единожды, но это не всегда возможно. Некоторые прикладные программы, в частности почтовые клиенты и программы просмотра сообщений Usenet, позволяют пользователям переопределять имена, используемые по умолчанию. Задать имя узла можно также в файле `/etc/hosts`. Этот файл используется при работе системы преобразования имен, альтернативной DNS. В файле `/etc/hosts` содержатся строки, начинающиеся с IP-адреса, за которым следует набор имен узла. Чаще всего первым после IP-адреса указывается *полностью определенное доменное имя*, в его состав входит имя компьютера и домен, которому он принадлежит, например `larch.threeroomco.com`. За полностью определенным доменным именем следуют так называемые *псевдонимы*. Обычно они представляют собой сокращенную форму имени, например `larch`. Если ваш компьютер корректно настроен для работы с сервером DNS и если на этом сервере содержатся записи для вашего компьютера, нет необходимости определять имя узла в файле `/etc/hosts`. Если сервер DNS работает ненадежно или если в результате некорректной работы маршрутизаторов сервер DNS периодически становится недоступным, записи в `/etc/hosts` повысят надежность работы вашего компьютера в сети. Кроме того, вы, возможно, захотите поставить в соответствие адресу 127.0.0.1 имена `localhost.localdovaib` и `localhost`. Примеры записей в файле `/etc/hosts` приведены ниже.

```
10.92.68.1 larch.threeroomco.com larch
127.0.0.1 localhost.localdomain localhost
```

СОВЕТ ✔ Если в процессе загрузки системы возникает пауза в несколько секунд и даже несколько минут (в особенности такая пауза бывает заметной при запуске программы `sendmail`), это может означать, что при соединении с сервером DNS возникают проблемы и вам желательно определить имя узла в файле `/etc/hosts`.

Если компьютер содержит несколько сетевых интерфейсов, вы можете задать одно имя узла посредством команды `hostnane` или определить в файле `/etc/hosts` отдельное

имя для каждого интерфейса. (Сервер DNS также позволяет задать для одного компьютера несколько имен.)

СОВЕТ Настраивая небольшую сеть, вы можете указать имена всех компьютеров в файлах `/etc/hosts`; при этом необходимость в использовании сервера DNS отпадает. Однако при увеличении размеров сети редактировать файлы `/etc/hosts` становится все труднее. В этом случае целесообразно перейти к использованию централизованного сервера DNS.

Сохранение внесенных изменений

Некоторые из описанных выше процедур настройки системы предполагают редактирование конфигурационных файлов. К таким процедурам относятся установка имени узла в файле `/etc/hosts` и указание адресов серверов DNS в файле `/etc/resolv.conf`. Установки, выполненные таким способом, продолжают действовать до тех пор, пока соответствующий файл не будет поврежден, либо до переинсталляции системы. Другие изменения конфигурации носят временный характер. Характеристики системы, установленные с помощью утилит `ifconfig`, `route` или `hostnane`, действуют лишь до перезагрузки компьютера либо до тех пор, пока установки не будут изменены теми же средствами. Чтобы сохранить произведенные установки, надо внести соответствующие изменения в сценарий запуска системы либо отредактировать конфигурационный файл. Для этого используются текстовый редактор либо специальные инструментальные средства.

Использование инструментов с графическим интерфейсом

Один из самых простых способов сохранения внесенных изменений — использование инструментов с графическим пользовательским интерфейсом (если такие средства входят в состав дистрибутивного пакета; в Debian и Slackware, например, подобные инструменты отсутствуют).

- Red Hat и Mandrake. Эти системы содержат программу Linuxconf с графическим интерфейсом, предназначенную для настройки системы. Данная программа может также использоваться в других системах, например в LinuxPPC. Версии данной программы, поставляемые в составе разных дистрибутивных пакетов, предоставляют варианты интерфейса, несколько отличающиеся друг от друга. Чтобы запустить программу linuxconf, достаточно ввести в командной строке ее имя. Эта программа может работать в текстовом режиме (в этом случае меню создается алфавитно-цифровыми средствами), в графическом режиме, а также в режиме Web-сервера, позволяющем выполнять удаленное администрирование.

- SuSE. В системе SuSE содержатся инструменты YaST (Yet Another Setup Tool) и YaST2. YaST работает в текстовом режиме и формирует систему меню. YaST2 выполняет те же функции, что и YaST, но предоставляет графический пользовательский интерфейс. Окно YaST2 показано на рис. 2.1. Для запуска этих программ используются соответственно команды `yast` и `yast2`.

- Caldera. В системе Caldera используется инструмент COAS (Caldera Open Administration System) с графическим интерфейсом. Для его запуска надо вызвать в окне `xterm` команду `coastool`.

- **TurboLinux**. В TurboLinux для установки конфигурации применяется графическая программа TurboLinux Configuration Crnter. Для ее запуска используется команда `turbocfgcenter`.

- Все дистрибутивные пакеты. В рамках проекта Webmib (`http://www.webmib.com/webmib/`) создан инструмент администрирования, работающий на базе Web. Его можно использовать в различных разновидностях Linux, а также в системах, отличных от Linux, например в разных версиях UNIX. При установке Linux данный инструмент не инсталлируется, но если Webmib поддерживает конкретную систему, его установка не составляет труда.

Различные инструменты конфигурации реализованы по-разному, но все они позволяют устанавливать параметры системы посредством меню. Выбрав соответствующие пункты меню, можно выполнять установки, которые будут сохраняться постоянно. Например, в окне, показанном на рис. 2.1, можно щелчком мышью установить опцию Static Address Setup, ввести IP-адрес, а также активизировать кнопку Hostname and Nameserver или Routing и выполнить дополнительные установки.

Инструменты с графическим интерфейсом имеют один недостаток. Часто они не позволяют выполнить подробную настройку системы. Так, например, с помощью некоторых инструментов невозможно сконфигурировать систему для работы с несколькими интерфейсами и шлюзами. Как правило, подобные инструменты ориентированы на простые конфигурации. В таких случаях приходится вручную корректировать содержимое конфигурационных файлов.

Редактирование конфигурационных файлов

В табл. 2.1 показано расположение некоторых конфигурационных файлов, содержащих команды клиента DHCP и некоторую дополнительную информацию о настройках. В этих файлах также находится информация о статических IP-адресах. Перед тем как изменять настройки компьютера, следует внимательно просмотреть эти файлы и найти в них вызовы утилит `ifconfig`, `route`, `hostnane` и другие подобные команды. Некоторые файлы не содержат команд, вместо этого в них присутствуют переменные окружения, которые представляют информацию о том, пользуется ли система услугами сервера DHCP или она настроена на работу со статическими IP-адресами. Как правило, проанализировав сценарии и конфигурационные файлы, можно получить информацию, достаточную для того, чтобы изменить настройку системы в соответствии со своими потребностями.

Если посредством редактирования конфигурационных сценариев и конфигурационных файлов не удается сделать необходимые установки, вам следует включить в сценарии запуска команды конфигурирования системы. В большинстве версий системы сценарий запуска содержится в файле `/etc/rc.d/rc.local`, а в системе SuSE используется сценарий, находящийся в файле `/etc/rc.d/boot.local`. В Debian локальный сценарий отсутствует, но вы можете создать собственный сценарий и поместить его в каталог `/etc/rc.boot`. В состав такого сценария можно включить любые команды, в том числе вызовы `ifconfig` и `route`. Этот сценарий получает управление после выполнения основных сценариев, используемых при запуске системы, таким образом, подобное расположение команд конфигурации сети нельзя признать идеальным. Тем не менее в таком сценарии удобно выполнять некоторые настройки, например добавлять новые маршруты.

Использование PPP-соединений

При рассмотрении вопросов сетевого взаимодействия предполагается, что компьютер под управлением Linux подключен к обычной локальной сети, узлы которой соединены посредством сетевых кабелей (например, к сети Ethernet). В такой среде можно инсталлировать серверы (работа серверов рассматривается в части II и III). При организации работы серверов необходимо учитывать вопросы безопасности, которые обсуждаются в части IV. В реальных сетях не все узлы подключены посредством постоянных соединений. Некоторые компьютеры подключаются через коммутируемые линии с помощью модемов. В большинстве случаев устанавливать сервер на компьютере, подключаемом через телефонную линию, не имеет смысла, но иногда модемы используются для подключения небольших сетей к Internet; в такой сети могут присутствовать различные серверы. Возможно, что при подключении по коммутируемой линии возникнет необходимость обеспечить взаимодействия с Internet всех компьютеров локальной сети, имея в наличии лишь один IP-адрес. В этом случае придется воспользоваться средствами NAT, которые будут подробно обсуждаться в главе 25. Однако, для того чтобы средства NAT могли работать, необходимо сначала установить PPP-соединение. Вопросы PPP-взаимодействия обсуждаются в последующих разделах.

PPP через Ethernet

В некоторых низкоуровневых соединениях DSL используется разновидность протокола PPP, известная под названием PPPoE. Поддержка PPPoE реализована в ядре Linux, но соответствующие средства считаются экспериментальными. На момент написания данной книги для поддержки PPPoE чаще всего использовалась клиент-программа Roaring Penguin PPPoE (`http://www.roaringpenguin.com/pppoe/`). Эта программа реализована для различных платформ и распространяется в исходных кодах либо как пакет RPM.

После инсталляции Roaring Penguin необходимо сконфигурировать эту программу, для чего надо выполнить команду `asdl-setup` либо `tkpppoe` (`asdl-setup` соответствует алфавитно-цифровому варианту данной программы, а посредством команды `tkpppoe` выполняется настройка разновидности Roaring Penguin, предоставляющей графический пользовательский интерфейс). После запуска сценарий настройки запрашивает пользовательское имя и пароль и создает сценарий запуска с именем `asdl-start`. Сценарий `asdl-start` впоследствии используется для инициализации PPPoE-соединения.

Заметьте, что Roaring Penguin предполагает, что средства поддержки сетевых устройств сконфигурированы правильно. Linux обеспечивает обмен посредством модемов DSL, действующих на базе Ethernet; при этом очевидно, что в системе должна быть включена поддержка Ethernet-карты. Если же для взаимодействия с модемом DSL используется интерфейс USB либо если к компьютеру подключен внутренний модем, для таких устройств надо установить специальные драйверы. На момент написания данной книги были доступны лишь экспериментальные варианты этих драйверов.

Использование программы с графическим интерфейсом для обмена по коммутируемой линии

PPP — достаточно сложный протокол; при его настройке используется большое число различных опций. Если значения этих опций выбраны неправильно, взаимодействие посредством PPP может не состояться. По этой причине многие пользователи избегают работать с конфигурационными сценариями и предпочитают устанавливать PPP-соединение с помощью специальной программы с графическим интерфейсом. Программы, поддерживающие PPP-соединение в системе Linux, очень похожи на программы аналогичного назначения, используемые в других операционных системах, например в Windows. Поэтому для тех, кто имеет опыт в использовании PPP-соединений в других системах, не составит труда решить ту же задачу в Linux.

Разные программы установления соединений по коммутируемым линиям различаются между собой лишь в деталях. В данном разделе рассматривается популярная программа KPPP, которая входит в состав K Desktop Environment (KDE). С KPPP можно работать, даже не используя KDE. Кроме KDE для установления соединений по телефонным линиям можно применять программу GNOME PPP (она входит в состав GNU Network Object Model Environment, или GNOME), а также инструменты, не являющиеся частями интегрированных сред, например X-ISP (`http://xisp.hellug.gr`).

Перед использованием KPPP необходимо получить учетную запись и протестировать модем. Чтобы убедиться в работоспособности модема, надо прежде всего подключить его к компьютеру и включить питание. Далее следует попытаться передать данные на `/dev/ttyS0`, `/dev/ttyS1` или какой-либо другой порт. Если файлы устройств автоматически создавались с помощью `devfs` (`http://www.atnf.csiro.au/~rgooch/linux/docs/devfs.html`), то вы можете использовать `/dev/tts/0`, `/dev/tts/1` и т. д. Для тестирования модема можно воспользоваться коммуникационными программами, такими как `minicom` или Seyon (обе они аоставляются в составе большинства дистрибутивных пакетов Linux). Если вы получите от модема сообщение AT, это значит, что модем можно использовать в системе Linux.

Для того чтобы запустить KPPP, надо выбрать соответствующий пункт меню на рабочем столе либо ввести в окне `xterm` команду `kppp`. В результате на экране отобразится окно, показанное на рис. 2.4. Если вы запускаете KPPP впервые, в списке Connect to будет отсутствовать имя провайдера, а поле Ligin ID будет пустым. Для того чтобы настроить программу так, чтобы с ее помощью можно было пользоваться учетной записью, выполните следующие действия.

1. Щелкните на кнопке Setup. В появившемся диалоговом окне KPPP Configuration (рис. 2.5) вам надо ввести основные параметры, определяющие особенности установления соединения.

2. Для регистрации новой учетной записи щелкните на кнопке New. Программа спросит, хотите ли вы воспользоваться мастером или предпочитаете выполнять настройку с помощью диалоговых окон. Несмотря на то что использование мастера упрощает задачу, он начинает диалог с того, что предлагает выбрать страну, причем в перечне стран отсутствуют США. Поэтому я выбираю вариант настройки про-

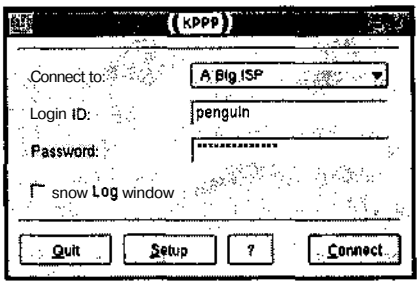


Рис. 2.4. Программы поддержки соединений по телефонным линиям с графическим интерфейсом предоставляют возможность выбрать учетную запись, задать пользовательское имя и пароль, а также инициировать соединение

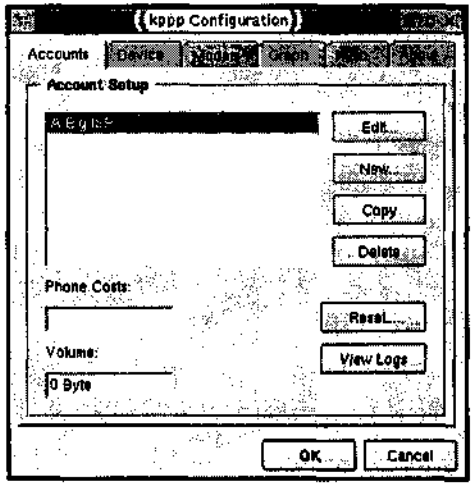


Рис. 2.5. Диалоговое окно KPPP Configuration позволяет управлять аппаратными средствами, посредством которых устанавливается PPP-соединение, в частности, выбрать устройство, к которому подключен модем, и модифицировать данные об учетных записях

граммы с помощью диалоговых окон, в результате чего отображается окно New Account, показанное на рис. 2.6.

3. Введите в поле Connection Name имя провайдера.

4. Щелкните на кнопке Add. В результате отобразится небольшое окно, в котором вы можете задать номер телефона вашего провайдера. После того, как вы щелкнете на кнопке OK, введенный вами номер отобразится в текстовой области Phone Number. Повторяя эту операцию, вы можете ввести несколько номеров. При попытке установления соединения программа будет набирать эти номера последовательно один за другим до тех пор, пока не обнаружит свободный номер.

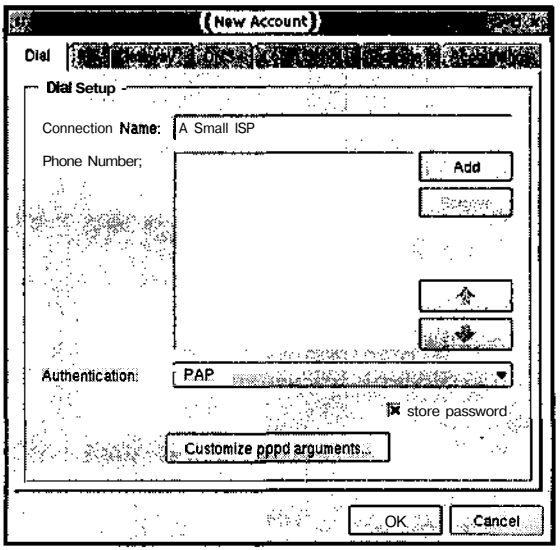


Рис. 2.6. Диалоговое окно New Account позволяет ввести информацию об учетной записи

5. В настоящее время большинство провайдеров используют протокол PAP (Password Authentication Protocol — протокол аутентификации с помощью пароля), поэтому целесообразно принять протокол PAP, выбранный в окне New Account по умолчанию. Вы можете изменить протокол, в частности, некоторые провайдеры используют вместо PAP протокол CHAP (Challenge Handshake Authentication Protocol — протокол аутентификации путем опроса при начальном обмене параметрами).

6. Если провайдер предоставит вам список адресов серверов DNS, активизируйте вкладку DNS в окне New Account и введите адрес каждого сервера в поле DNS IP Address, завершая ввод каждого адреса щелчком на кнопке Add.

7. Щелкните на кнопке OK в окне New Account. После этого вы увидите, что в списке учетных записей в окне KPPP Configuration появился новый пункт (см. рис. 2.5).

8. Выберите вкладку Device в диалоговом окне KPPP Configuration. В качестве значения Modem Device установите имя устройства, используемого в вашей системе для подключения модема. Чаще всего это устройство имеет имя `/dev/modem`, но могут также использоваться `/dev/ttySO`, `/dev/ttyS1` и некоторые другие устройства. На этой же вкладке вы можете задать значение опции Connection Speed. По умолчанию используется значение 57600, но скорость 115200 часто позволяет добиться лучших результатов. (Большинство аппаратных средств не поддерживает более высокие скорости.) Данная опция задает скорость обмена между вашим компьютером и модемом. Скорость обмена между вашим модемом и модемом провайдера скорее всего будет гораздо ниже. Если модемы используют сжатие данных, желательно установить скорость обмена между компьютером и модемом как минимум вдвое выше, чем скорость обмена между модемами.

9. Щелкните на кнопке OK в окне KPPP Configuration. Теперь новая учетная запись станет доступной в главном окне KPPP (см. рис. 2.4).

Окно KPPP Configuration, как и New Account, содержит несколько вкладок. Опции, доступные посредством некоторых из вкладок, здесь не рассматривались. В большинстве случаев вы можете использовать значения по умолчанию, но иногда приходится явно задавать некоторые параметры. Если при попытке соединения с компьютером провайдера у вас возникли проблемы, просмотрите значения опций, расположенных на различных вкладках, и при необходимости измените их. В документе PPP HOWTO (`http://www.linuxdoc.org/HOWTO/PPP-HOWTO/`) можно найти дополнительную информацию о PPP-соединении, в том числе об установлении PPP-соединения в режиме отладки.

С помощью программы с графическим интерфейсом PPP-соединение устанавливается чрезвычайно просто. После загрузки программы достаточно щелкнуть на кнопке Connect (в некоторых программах эта кнопка может иметь другое имя). Некоторые программы предоставляют пользователю сведения о ходе установления соединения, кроме того, модем может воспроизводить звуковой сигнал во время ведения "переговоров" о параметрах, которые будут использоваться при обмене данными. Работая с KPPP, вы можете щелкнуть на кнопке Show Log Window и получить дополнительную информацию о соединении. Некоторые программы, в том числе KPPP, требуют, чтобы перед активизацией кнопки Connect было введено пользовательское имя (поле Login ID) и пароль. Другие программы запрашивают эти сведения после щелчка на кнопке Connect. Кроме того, пользователь может сохранять пароль на диске (в KPPP для этого надо установить флажок опции Store Password в диалоговом окне New Account).

После установления соединения на кнопке Connect отобразится новая запись. С этого момента щелчок на данной кнопке приведет к разрыву соединения. (В некоторых программах для разрыва соединения используется другая кнопка, либо после установления соединения отображается новое диалоговое окно.) Не забывайте о том, что после окончания работы соединение необходимо разорвать, в особенности, если телефонная компания или провайдер предоставляет вам услуги на условиях поминутной оплаты. В противном случае вы рискуете получить счет, в котором будет указана крупная сумма.

Сохраняя пароль на диске, вы создаете угрозу безопасности системы. Если к вашему компьютеру никто другой не имеет доступа, риск невелик. Если же на этом компьютере работают и другие пользователи, при этом некоторые из них не имеют учетных записей и не могут устанавливать PPP-соединения, опасность существенно увеличится. В любом случае желательно, чтобы пароль, применяемый для установления PPP-соединения, не использовался в других целях (например, для входа в систему или для доступа к защищенным данным). И даже если этот пароль будет похищен, вам достаточно будет задать для PPP-взаимодействия новый пароль.

Редактирование конфигурационных сценариев

Программы с графическим интерфейсом удобны для тех пользователей, которые не имеют достаточного опыта работы с PPP-соединениями, но в некоторых случаях эти инструменты оказываются непригодными. Например, если вы хотите, чтобы PPP-соедине-

ние устанавливалось автоматически, программа с графическим интерфейсом не позволит вам сделать это. В этом случае для инициализации соединения приходится использовать сценарии. Такие сценарии можно запускать либо вручную, либо как часть системы автоматического соединения. При этом необходимо установить опции аутентификации и задать конфигурацию самих сценариев.

Использование опций аутентификации

Как было сказано ранее, для аутентификации пользователей, обращающихся по коммутируемым линиям, большинство провайдеров применяет протокол PAR Для того чтобы сценарий, предназначенный для установления соединения, мог использовать этот протокол, необходимо отредактировать файл /etc/ppp/pap-secrets. (При работе с протоколом CHAP используется файл /etc/ppp/chap-secrets. Содержимое файла chap-secrets имеет тот же формат, что и данные в файле pap-secrets.) В файле /etc/ppp/pap-secrets содержится последовательность строк; каждая строка соответствует отдельной учетной записи PPP и имеет следующий формат:

имя_пользователя сервер пароль IP-адрес

Компоненты строки отделяются друг от друга одним или несколькими пробелами или символами табуляции. Назначение этих компонентов описано ниже.

- *имя_пользователя*. Имя, используемое для идентификации пользователя на компьютере провайдера. Это имя никак не связано с именем пользователя, которое указывается при регистрации в системе Linux; оно проверяется только при попытке зарегистрироваться на сервере провайдера.
- *сервер*. Имя компьютера, к которому обращается клиент при попытке установить PPP-соединение. Как правило, пользователь не обязан знать имя сервера, поэтому чаще всего в данном поле содержится символ *, который указывает на то, что PPP-соединение может быть установлено с любым узлом.
- *пароль*. Как нетрудно догадаться, в этом поле указывается пароль, используемый для регистрации на удаленном компьютере.
- *IP-адрес*. IP-адрес, который система предполагает получить. Большинство серверов не гарантирует, что при установлении PPP-соединения будет выделен конкретный IP-адрес, поэтому данное поле обычно оставляют пустым (т. е. строка содержит только три поля).

ВНИМАНИЕ В файле pap-secrets пароль хранится в незашифрованном виде. Это значит, что каждый, кто получит в свое распоряжение данный файл, сможет установить PPP-соединение под вашим именем. Поэтому данный пароль следует использовать только по его прямому назначению. Для получения почты, регистрации в сети и прочих подобных случаев надо применять другие пароли. Для повышения безопасности во многих дистрибутивных пакетах в качестве владельца файла pap-secrets указан пользователь root, и доступ к файлу имеет только он. Без крайней необходимости не следует изменять права доступа к файлу pap-secrets.

В большинстве случаев каждый конкретный компьютер настраивается для взаимодействия с единственным провайдером, поэтому в файле pap-secrets такого компьютера

содержится только одна строка. Пример содержимого файла pap-secrets приведен ниже.

```
penguin * w8terfow1
```

Настройка сценариев

Отредактировав файл аутентификации, ориентированный на использование протокола PAP или CHAP, можно приступать к настройке сценария, предназначенного для установления соединения. Поскольку для взаимодействия посредством протокола PPP все чаще используются программы с графическим интерфейсом, сценарии размещаются в каталоге с документацией, например /usr/share/doc/ppp-*версия*/scripts; где *версия* означает версию PPP, используемую в конкретном дистрибутивном пакете, например 2.4.0. Наибольший интерес представляют три сценария.

- ppp-on. Этот сценарий устанавливает основные переменные, в частности, задает номер телефона провайдера и вызывает Linux-программу поддержки протокола PPP (pppd).
- ppp-on-dialer. Сценарий ppp-on передает ppp-on-dialer программе pppd, которая использует ppp-on-dialer для управления начальными этапами взаимодействия с компьютером провайдера.
- ppp-off. Этот сценарий завершает сеанс PPP-взаимодействия.

Для того чтобы обеспечить взаимодействие с провайдером, необходимо отредактировать сценарий ppp-on, а в некоторых случаях и сценарий ppp-on-dialer. Кроме того, вы, вероятно, захотите переместить все три сценария в каталог, из которого их было бы удобно взывать, например в /usr/local/bin. В сценарий ppp-on надо внести следующие изменения.

- Найдите переменную TELEPHONE и задайте в качестве ее значения номер телефона провайдера. В результате соответствующая запись должна иметь вид TELEPHONE=123-4567.
- Задайте значения переменных ACCOUNT и PASSWORD. Если провайдер использует протокол PAP, данные переменные не будут использоваться; в этом случае вы можете оставить их значения без изменения.
- Если провайдер предоставляет вам фиксированный IP-адрес и если вы знаете IP-адрес сервера провайдера, можете указать эти адреса в переменных LOCAL_IP и REMOTE_IP. Аналогично, если вам известна маска подсети, вы можете задать ее в качестве значения переменной NETMASK. В противном случае все три переменные можно оставить без изменения.
- Найдите переменную DIALER_SCRIPT и задайте ее значение так, чтобы она ссылалась на сценарий ppp-on-dialer. (Понятно, что DIALER_SCRIPT должна указывать не на исходный вариант файла, содержащийся в каталоге с документами, а на файл, содержимое которого вы изменили в соответствии с вашими требованиями.) По умолчанию для этой переменной задано значение /etc/ppp/ppp-on-dialer, но, как было сказано выше, вы можете выбрать расположение файла ppp-on-dialer по своему усмотрению.

- В конце сценария содержится вызов pppd. Эта программа поддерживает большое количество опций. Опции, указанные в сценарии, за исключением некоторых, изменять не следует. Возможно, вам придется задать имя файла устройства, используемого для подключения модема (по умолчанию указано устройство /dev/ttySO), а также скорость взаимодействия компьютера с модемом (по умолчанию используется значение 38400, но скорость 115200, как правило, дает лучшие результаты).

Скорректировав содержимое ppp-on, можно приступать к редактированию сценария ppp-on-dialer. Этот сценарий управляет взаимодействием программы pppd с модемом, в частности, использованием команд, предназначенных для установления взаимодействия, а также процессом аутентификации (в случае, если провайдер не использует средства PAP или CHAP). Сценарий вызывает утилиту chap, предназначенную для обмена текстовыми данными. Основную часть сценария составляют пары строк, представляющие собой ожидаемые сообщения и ответы на них, расположенные в два столбца. В первом столбце указаны сообщения, которые ожидает получить сценарий, а во втором столбце — последовательности символов, которые программа chat посылает в ответ. Некоторые из сообщений имеют специальное назначение. Например, ABORT сообщает chat о необходимости прекращения работы в случае ошибки. Большинство строк оканчивается обратной косой чертой (\), а это означает, что следующая строка является продолжением предыдущей. (На самом деле программе chat передается одна строка параметров; пары "сообщение-ответ" представлены в виде столбцов лишь для удобства восприятия.) В конце последней строки обратная косая черта отсутствует.

Изменения следует вносить только в последние три строки файла ppp-on-dialer. По умолчанию при составлении сценария предполагалось, что провайдер не использует PAP, поэтому в последних двух строках предусмотрена передача имени пользователя и пароля в ответ на запрос. (Имя пользователя и пароль хранятся в переменных ACCOUNT и PASSWORD; их значения задаются в сценарии ppp-on.) При необходимости вы можете удалить эти строки или поставить в начале их символы #, указывающие на то, что данные строки содержат комментарии. Если вы сделаете это, то вам также надо удалить обратную косую черту в третьей с конца строке. Удаление двух последних строк и изменение предшествующей им строки приведет к тому, что если pppd попытается использовать для аутентификации соединения протокол PAP или CHAP, chat завершит работу. Если протоколы PAP и CHAP не применяются, вам, возможно, потребуется отредактировать в последних строках сообщения, которые система ожидает получить от провайдера. Может быть, вы захотите выполнить дополнительные команды, например, запустить на компьютере провайдера программу поддержки PPP. В этом случае вам придется включить одну или несколько строк и указать в них в качестве ожидаемого сообщения приглашение для ввода команды.

Использование сценариев при установлении PPP-взаимодействия

Редактирование сценариев — наиболее трудоемкая часть работы по обеспечению PPP-взаимодействия. После того как данная задача выполнена, вам остается лишь ввести ppp-on, после чего соединение будет инициализировано. (Если сценарий ppp-on расположен в каталоге, который не учтен в переменной окружения PATH, вам придется указать полный путь к этому файлу.) Используя внешний модем, вы сможете следить за ходом установления соединения по светодиодным индикаторам на его панели; при соответствующей настройке модема вы также услышите характерный звук. Если соединение

будет установлено нормально, то через несколько секунд вы получите доступ к Internet и сможете использовать клиент-программы для взаимодействия с Internet-серверами.

В случае возникновения проблем вам надо, прежде всего, ознакомиться с содержимым файла протокола (обычно сообщения об установлении соединения и возникающих при этом ошибках записываются в файл `/var/log/messages`). В конце файла должна содержаться информация о действиях программы `pppd`, включая операции, которые завершились неудачно. Там вы найдете сведения об истечении времени тайм-аута при ожидании PAP-аутентификации, об ошибках при выполнении `chat` и другие данные. Если вы не можете понять смысл сообщений, используйте термины, найденные в файле протокола, как ключевые слова при поиске на сервере `http://groups.goggle.com`. На этом сервере хранятся архивы сообщений, отправленных в группы новостей Usenet и посвященных сетевому взаимодействию системы Linux. Среди них вы найдете обсуждение проблем, связанных с установлением PPP-соединений. Не исключено, что какое-то из сообщений будет содержать ответ на ваш вопрос, либо, по крайней мере, вы поймете, в каком направлении следует искать решение проблемы. Полезными также будут сведения, содержащиеся в документе PPP HOWTO.

Большинство дистрибутивных пакетов сконфигурировано так, что инициализировать PPP-соединение может только пользователь `root`. Многие воспринимают это как недостаток. Подобные меры принимаются для того, чтобы повысить безопасность в том случае, когда на одном компьютере работают несколько пользователей. Однако иногда такие ограничения мешают в работе. Для того чтобы частично разрешить эту проблему, для программ, предназначенных для взаимодействия по коммутируемым линиям, устанавливается бит SUID, в результате чего эти программы запускаются с привилегиями `root`. Очевидно, что при этом снова приходится решать вопросы защиты. В частности, многие администраторы сужают круг пользователей, которые имеют право запускать подобные программы. Они создают группу PPP, включают в нее тех пользователей, которым действительно необходим доступ к удаленным узлам по коммутируемым линиям, и устанавливают права доступа так, что запустить программу могут только члены группы PPP.

Многие провайдеры сообщают пользователям IP-адреса серверов DNS, которые должны использоваться при взаимодействии посредством PPP-соединения. Эти сведения можно включить в файл `/etc/resolv.conf` (структура данного файла обсуждалась выше).

Установление соединения по запросу

При работе на отдельной рабочей станции или компьютере необходимость использовать специальную программу установления взаимодействия или вручную запускать соответствующий сценарий практически не мешает работе. Если же PPP-соединением пользуется несколько компьютеров, подключенных к локальной сети, могут возникать проблемы. Часто пользователи пытаются установить соединение в то время, когда оно активно, разорвать соединение, когда другие пользователи обмениваются через него информацией с Internet, либо по окончании работы оставляют соединение активным на длительное время. Для решения этих проблем были созданы средства установления *соединения по запросу* (dial-on-demand). В системе Linux подобные функции выполняет инструмент под названием `diald`. Эта программа выявляет трафик, направленный из локальной сети к внешним узлам, и инициализирует PPP-соединение. Кроме того, если в течение определенного времени сетевая активность отсутствует, эта программа разрывает соединение. В результате клиентские программы, расположенные в локальной сети,

могут работать почти так же, как и программы, находящиеся на компьютерах, постоянно подключенных к Internet; для того чтобы установить или разорвать PPP-соединение, пользователям не приходится предпринимать никаких действий. Различия проявляются лишь в том, что с момента, когда diald обнаруживает попытку обращения к внешнему узлу и до установления PPP-соединения, проходит определенное время. Это связано с тем, что система должна обратиться к модему, а модем, в свою очередь, установить соединение с модемом провайдера. Через некоторое время после прекращения сетевой активности соединение разрывается. Если вы зададите слишком малое значение этого интервала, задержка будет возникать слишком часто, так как с момента получения Web-страницы до запроса очередного документа сетевое взаимодействие отсутствует и система может разорвать соединение. Заметьте также, что если PPP-соединение не активно, при обращении к Web-странице броузер выведет сообщение о том, что документ не доступен. Причина в том, что время тайм-аута, используемое при работе броузера, значительно меньше времени, необходимого для установления PPP-соединения.

Чтобы программа diald могла работать, необходимо установить в ядре Linux средства поддержки SLIP (вопросы настройки и компиляции ядра рассматривались в главе 1). Протокол SLIP необходим для связывания компьютера с программой diald. Эта программа поддерживает постоянно активный сетевой интерфейс, поэтому она имеет возможность выявлять сетевой трафик и устанавливать при необходимости PPP-соединение.

К сожалению, diald не входит в состав большинства дистрибутивных пакетов Linux, поэтому этот инструмент необходимо устанавливать отдельно. Исходный код программы можно найти на сервере http://diald.sourceforge.net, а для того, чтобы получить двоичные коды diald для RPM и Debian, надо обратиться соответственно по адресам http://www.rpmfind.net и http://www.debian.org/distrib.packages.

Для настройки программы diald используются три конфигурационных файла, описанных ниже.

- /etc/diald.conf. В этом файле содержатся опции, подобные тем, которые используются сценарием ppp-on, например, имя устройства, посредством которого подключен модем (device), и скорость соединения (speed). Посредством опций local и remote задаются IP-адреса для внутреннего использования в программе diald. Эти адреса должны принадлежать одному сегменту, но следует следить за тем, чтобы они не совпадали с адресами узлов вашей локальной сети. Вы можете использовать для этой цели IP-адреса, специально выделенные для внутренних сетей, например адреса, принадлежащие диапазону 192.168.x.x.

- /etc/ppp/diald-dialer. Этот файл практически идентичен рассмотренному ранее сценарию ppp-on-dialer. При настройке его содержимое необходимо изменить так же, как и ppp-on-dialer.

- /usr/lib/diald/standard. filter. В этом файле задается значение тайм-аута. Если в течение указанного интервала времени сетевая активность отсутствует, программа diald должна разорвать соединение. Если в процессе работы вы обнаружите, что соединение прекращается слишком быстро, имеет смысл вернуться к значению тайм-аута по умолчанию, первоначально заданному в файле /usr/lib/diald/standard.filter.

Если ваш провайдер использует PAP или CHAP, то кроме перечисленных выше конфигурационных файлов, вам надо также отредактировать файл /etc/ppp/pap-secrets или /etc/ppp/chap-secrets. В соответствующем файле указываются те же данные, что и при настройке PPP-соединения, устанавливаемого с помощью обычных сценариев. Вам также придется включить в файл /etc/resolv.conf адреса серверов DNS, которые сообщит провайдер. Для того чтобы запустить diald, надо задать команду /usr/sbin/diald. Сделать это может только пользователь root. После этого diald будет распознавать трафик, направленный извне, и устанавливать PPP-соединения. Первая попытка обращения к серверу Internet закончится неудачей, так как для установления соединения требуется время, превышающее время тайм-аута большинства служб Internet. Вторая попытка будет успешной.

Если вы хотите, чтобы программа diald автоматически запускалась при загрузке системы, вам надо создать сценарий запуска SysV или включить дополнительные записи в локальный сценарий (/etc/rc.d/rc.local или /etc/rc.d/boot.local). Программа diald будет нормально работать и в том случае, если ваш компьютер выполняет функции NAT-маршрутизатора.

Резюме

Для того чтобы работа в сети стала возможной, необходимо реализовать тот или иной тип сетевого соединения. В настоящее время для создания подавляющего большинства локальных сетей используется технология Ethernet. В системе Linux присутствуют надежные средства поддержки сети Ethernet. IP-адреса в сети распределяются либо вручную, либо для этой цели используются клиенты и сервер DHCP. Linux поддерживает оба способа распределения адресов. Настройка большинства локальных сетей выполняется приблизительно так же, как и настройка сетей Ethernet. Единственным исключением являются PPP-соединения. Протокол PPP обычно применяется для обеспечения сетевого взаимодействия по коммутируемым линиям. Для поддержки PPP-соединения используется программа pppd, выполняющаяся в системе Linux в режиме демона. Обращение к pppd осуществляется с помощью специальных программ с графическим интерфейсом, сценариев либо посредством программы diald. В любом случае после активизации PPP-соединения формируется интерфейс, который с программной точки зрения аналогичен Ethernet или другому сетевому интерфейсу.

Глава 3

Альтернативные стеки протоколов

Компьютерная программа — идеальный инструмент для решения тех задач, которые предполагают скрупулезное следование предписаниям. В ситуациях, не предусмотренных инструкциями, компьютер становится практически беспомощным. Поэтому для обеспечения работы сетей тщательно разработаны протоколы — подробное описание действий узлов сети при выполнении транзакций. Как было сказано в главе 1, протоколы объединяются в иерархическую систему, называемую *стеком сетевых протоколов,* или *стеком протоколов.* Наиболее часто в настоящее время используется стек протоколов TCP/IP. На базе протоколов семейства TCP/IP построена вся сеть Internet, кроме того, протоколы данного семейства широко используются при работе различных операционных систем, в частности Linux. В главе 2 была описана конфигурация системы для поддержки TCP/IP. Помимо TCPЯP, существует ряд альтернативных стеков протоколов, которые также поддерживаются в Linux.

В начале данной главы представлены обзор стеков протоколов и краткое описание TCP/IP. Далее обсуждаются три наиболее часто используемых стека: AppleTalk, IPX и NetBEUI. Эти стеки протоколов применяются в основном в локальных сетях, содержащих компьютеры Macintosh и PC под управлением Windows. С их помощью обеспечивается разделение файлов и принтеров.

Общие сведения о стеках протоколов

Для того чтобы вести предметный разговор о стеках протоколов и обсуждать их достоинства и недостатки, необходимо иметь хотя бы общее представление о том, как организован стек, какие функции выполняют протоколы, входящие в его состав, и как они реализованы. Большинство стеков протоколов действует по единому принципу и отличается лишь в деталях. Однако именно эти детали и являются основным аргументом в пользу выбора тех или иных протоколов.

Модель сетевого взаимодействия OSI

В основу работы стека протоколов положена модель OSI (Open System Interconnection — взаимодействие открытых систем). Данная модель предусматривает семь уровней сетевого взаимодействия, на каждом из которых решаются конкретные задачи. Источником данных, предназначенных для передачи, является программа, находящаяся на верхнем уровне модели OSI, который называется прикладным уровнем. Программа передает данные нижележащему представительскому уровню, и далее информация перемещается вниз по стеку. На каждом уровне выполняется определенная обработка данных. Нижним уровнем OSI является физический уровень, на котором решаются вопросы передачи электрических сигналов, использования кабелей, концентраторов, коммутаторов и т. д. Именно на физическом уровне осуществляется реальная передача данных от передающего компьютера к принимающему. (Если оба компьютера принадлежат одному сегменту сети, данные передаются достаточно просто. В противном случае передача осуществляется поэтапно, и на каждом этапе выполняется дополнительная обработка информации.) На принимающем компьютере данные перемещаются вверх по стеку протоколов и в конечном итоге достигают программы на прикладном уровне. Получив информацию, программа может передать ответ. Данные, составляющие ответ, также движутся вниз по стеку, передаются через сетевые соединения, а затем на компьютере, которому они предназначены, перемещаются вверх по стеку протоколов. На рис. 3.1 проиллюстрирован описанный выше процесс.

Несмотря на то что OSI является универсальной моделью, описывающей сетевое взаимодействие, в реальных стеках протоколов редко поддерживаются все семь уровней. Наиболее часто используемые стеки — TCP/IP, AppleTalk и NetBEUI — могут быть описаны в терминах OSI. При этом стек TCP/IP насчитывает лишь четыре уровня, однако общие принципы обработки передаваемых данных остаются неизменными.

Каждый уровень модели OSI взаимодействует только с двумя уровнями; один из них расположен непосредственно над ним, а другой — под ним. (Исключениями являются прикладной и физический уровни. Прикладная программа взаимодействует непосредственно с пользователем, а на физическом уровне решаются вопросы соединения двух компьютеров.) Для программных средств, реализующих различные уровни стека протоколов на конкретном компьютере, должен быть четко определен интерфейс межуровневого взаимодействия. Компоненты, соответствующие определенным уровням, должны допускать замену. Так, например, на прикладном уровне могут работать Web-броузеры и Web-серверы. Если вы замените один броузер или сервер другим, работа всего стека протоколов не нарушится. (В некоторых броузерах и серверах могут отсутствовать определенные возможности, например, броузер может не поддерживать средства SSL, однако на самом деле такие вопросы скорее относятся к интеграции сетевых средств.) Аналогично вы можете на физическом уровне заменять сетевые кабели, концентраторы и даже сетевые карты с драйверами. При этом средства поддержки более высоких уровней не нуждаются в модификации.

Для двух компьютеров, взаимодействующих по сети, действия на некотором уровне одного компьютера строго согласованы с действиями на этом же уровне другого компьютера. В некотором смысле можно утверждать, что результаты, получившиеся при обработке данных на определенном уровне стека протоколов передающего компьютера,

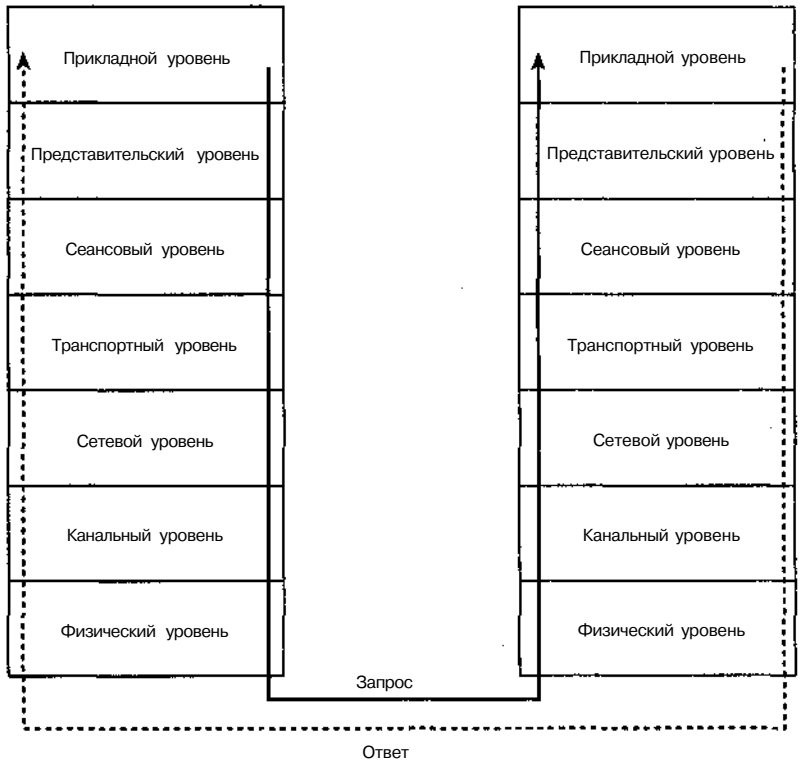


Рис. 3.1. Компоненты стека протоколов выполняют обработку данных для передачи их на другой компьютер

отменяются на этом же уровне принимающего компьютера. Главная цель работы всего стека протоколов — обеспечение взаимодействия программ прикладного уровня, поэтому каждый уровень на принимающем узле сети должен получать от нижележащего уровня в точности те же данные, которые предоставил соответствующий уровень на передающем узле. Взаимодействие компьютеров, использующих стеки протоколов, можно представить себе так, как будто некоторый уровень обменивается данными не с нижележащим и вышестоящим уровнями, а непосредственно с соответствующим уровнем другого компьютера. Очевидно, что стеки протоколов на разных компьютерах должны отвечать одному стандарту, даже если эти компьютеры работают под управлением различных операционных систем. Например, на разных уровнях стека TCP/IP, реализованного для Linux, Windows, MacOS и BeOS, выполняются одинаковые действия, несмотря на то, что коды соответствующих программных средств различаются.

Инкапсуляция и извлечение данных

Стек протоколов хорошо иллюстрирует перемещение данных между программными компонентами, поддерживающими сетевое взаимодействие, однако он не дает ответа на вопрос, какие же изменения претерпевает информация на этом пути. На различных уровнях стека протоколов выполняются инкапсуляция и извлечение данных. При инкап-

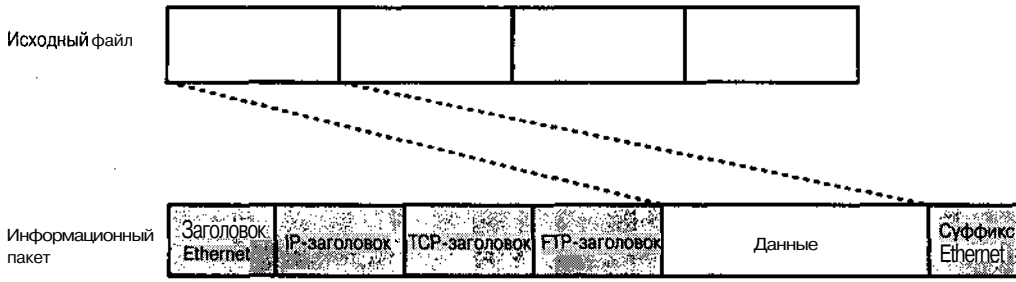


Рис. 3.2. При перемещении вниз по стеку протоколов исходные данные разбиваются на фрагменты. К каждому фрагменту добавляется служебная информация, обеспечивающая передачу пакета по сети и восстановление на принимающем узле исходных данных

суляции данные разбиваются на фрагменты, и к каждому фрагменту добавляется управляющая информация (после добавления управляющей информации фрагменты данных, в зависимости от уровня протокола, становятся пакетами или кадрами). Кроме того, информация, предназначенная для передачи, может быть модифицирована, но подобная обработка выполняется крайне редко. При извлечении данных выполняются действия, противоположные инкапсуляции.

Рассмотрим в качестве примера передачу содержимого файла с помощью протокола FTP (File Transfer Protocol — протокол передачи файлов) в сети Ethernet. При этом используется стек протоколов TCP/IP. Размер файла может превышать максимальный размер пакета данных, допустимый в TCPЯP или Ethernet. В этом случае содержимое файла должно быть разбито на несколько фрагментов. На различных уровнях стека протоколов к каждому из этих фрагментов добавляется заголовок и, возможно, суффикс (служебная информация, следующая за передаваемыми данными). Заголовок и суффикс содержат информацию, необходимую для того, чтобы система могла передавать и обрабатывать остальную часть пакета. Результат действий по инкапсуляции данных показан на рис. 3.2. На самом деле ситуация может быть более сложной. На некотором уровне стека протоколов пакет может быть разбит на несколько фрагментов меньшего размера, так, например, драйверы Ethernet могут разбить IP-пакет на два Ethernet-кадра. Подобные действия могут выполнять и маршрутизаторы. При этом заголовки IP, TCP и FTP, показанные на рис. 3.2, остаются в одном из кадров и не дублируются в другом кадре. Однако если на некотором уровне стека протоколов пакет разбивается на кадры, то на том же уровне стека на принимающем узле эти кадры обязательно объединяются в исходный пакет. Точно так же дело обстоит в том случае, если разбиение пакета выполнит маршрутизатор.

В зависимости от используемого стека протоколов и даже от состава стека, структура пакета может отличаться от представленной на рис. 3.2. Например, если для обмена данными используется Web-броузер, то вместо заголовка FTP, показанного на рис. 3.2, в составе пакета будет присутствовать заголовок HTTP. Если же для подключения компьютера к сети будут использованы сетевые карты и драйверы, отличные от Ethernet, то заголовок и суффикс Ethernet будут заменены на заголовок и суффикс, соответствующие другой сетевой технологии. Заметьте, что при передаче пакета из одной локальной сети в другую маршрутизатор реально выполняет подобную замену, т. е. удаляет существующие заголовок и суффикс и включает вместо них заголовок и суффикс другой сети.

Такая процедура выполняется несколько раз за время перемещения пакета по Internet, но данные, содержащиеся в составе пакета, доставляются в неизменном виде.

Заголовки и суффиксы содержат информацию, необходимую для доставки пакта по назначению, например, IP-адреса отправителя и получателя, номера портов, связанные с программами на прикладном уровне, расположение данных, содержащихся в пакете, в исходной битовой последовательности, и т. д. Маршрутизаторы используют эту информацию для определения маршрута пакета, а на принимающем узле эти сведения позволяют передать пакет той программе, для которой он предназначен. Адрес и порт отправителя используются при передаче ответа.

Роль стека протоколов TCP/IP в развитии сетей

В настоящее время TCP/IP является самым популярным стеком протоколов. В состав этого стека входят наиболее часто используемые протоколы, которые обсуждаются в данной книге. В большинстве приложений не реализована поддержка нескольких стеков протоколов, поэтому чаще всего приложение может работать с одним конкретным стеком. Одна из причин популярности стека протоколов TCP/IP — его гибкость. Протоколы ТСРЛР являются маршрутизируемыми протоколами, т. е. пакеты TCP/IP могут передаваться из одной локальной сети в другую. Для передачи пакетов между различными сетями не нужна единая карта Internet; при маршрутизации используется распределенная информация о структуре сети, хранящаяся на различных маршрутизаторах. Число допустимых адресов в сетях TCP/IP достаточно велико (в IPv4 адрес представляется 32 битами, а в IPv6 используются 128-битовые адреса; подробно IP-адреса рассматривались в главе 2), кроме того, в этих сетях поддерживается иерархическая структура имен. Эти положительные качества стали причиной того, что протоколы TCP/IP были выбраны в качестве основы для создания глобальной сети Internet.

Впервые протоколы TCP/IP были использованы в UNIX; система Linux "унаследовала" их. Как в Linux, так и в UNIX средства ТСРЛР используются для обеспечения работы различных компонентов системы. Сеть, в состав которой входят только компьютеры, работающие под управлением UNIX или Linux, может быть создана на основании TCP/IP, без использования других стеков протоколов.

В состав семейства TCP/IP входят HTTP, FTP, SMTP (Simple Network Mail Protocol - простой протокол передачи почтовых сообщений), NFS (Network File System — сетевая файловая система), Telnet, SSH (Secure Shell — защищенная оболочка), NNTP (Network News Transfer Protocol — протокол передачи сетевых новостей), X Window и многие другие протоколы. Широкое использование ТСРЛР привело к тому, что в инструментах, изначально ориентированных на работу с другими стеками протоколов, была реализована поддержка ТСРЛР. Например, несмотря на то, что в системе Windows используется стек протоколов NetBEUI (NetBIOS Extended User Interface — расширенный пользовательский интерфейс NetBIOS), средства поддержки протоколов SMB (Server Message Block — блок сообщений сервера) / CIFS (Common Internet Filesystem — общая межсетевая файловая система) могут взаимодействовать с ТСРЛР через NetBIOS (Network Basic Input/Output System — базовая сетевая система ввода-вывода). Начиная с Windows 95 все версии Windows поддерживают ТСРЛР. Аналогично, протоколы Apple, предназначенные для разделения файлов, могут работать не только с AppleTalk, но и с ТСРЛР.

Несмотря на свои достоинства и популярность, стек ТСРЛР не позволяет решить все задачи, возникающие при создании сетей. Так, например, в некоторых сетях могут при-

сутствовать компьютеры, не поддерживающие TCP/IP. В частности, старые компьютеры Macintosh обеспечивают обмен файлами только средствами AppleTalk, а некоторые машины под управлением DOS или Windows могут быть сконфигурированы лишь для работы с IPX или NetBEUI. Поэтому поддержка альтернативных средств протоколов является положительным качеством системы Linux.

AppleTalk

Стек протоколов создавался компанией AppleTalk параллельно с разработкой сетевого оборудования LocalTalk. Он нашел применение в компьютерах Macintosh, выпущенных в начале 1980-х. (Первоначально как аппаратные, так и программные средства назывались AppleTalk; в настоящее время это название используется лишь для обозначения программных компонентов.) С ростом популярности Ethernet Apple доработала AppleTalk для работы с аппаратным обеспечением Ethernet; этот вариант программ иногда называют EtherTalk. В системе Linux поддерживается стек протоколов AppleTalk и обеспечивается его работа как посредством аппаратуры LocalTalk, так и через Ethernet.

Как ни странно, версии Linux, ориентированные на компьютеры Macintosh, не поддерживают аппаратные средства LocalTalk, а допускают лишь работу AppleTalk через Ethernet. Поэтому для включения компьютера Macintosh под управлением Linux в сеть AppleTalk необходимо, чтобы на нем присутствовал Ethernet-адаптер.

Особенности AppleTalk

Подобно TCP/IP, AppleTalk использует 32-разрядные адреса. Подобно IP-адресу, адрес AppleTalk состоит из двух компонентов: адреса сети и адреса компьютера. В отличие от IP, длина каждого из компонентов фиксирована: 16 из 32 битов выделены для представления адреса сети, а остальные 16 битов — для идентификации компьютера. В сетях AppleTalk поддерживается процедура переговоров, предпринимаемых для получения компьютером сетевого адреса. Благодаря наличию такой процедуры администратор избавлен от необходимости явно указывать адреса. (Если вы захотите, можете задать адрес явно или запросить его из определенного диапазона, но обычно в этом нет надобности.)

Кроме адресов для идентификации компьютеров в AppleTalk-сетях существует система имен, предназначенная для того, чтобы упростить работу пользователей. Каждому компьютеру присваивается имя, кроме того, для этого компьютера определяется принадлежность к локальной группе машин, которая называется *зоной*. Полное имя состоит из имени компьютера и имени зоны. В небольших сетях информация о зоне может не использоваться, в этом случае компьютеры идентифицируются только посредством имени. Netatalk (основной пакет, предназначенный для поддержки AppleTalk в Linux) по умолчанию генерирует AppleTalk-имена на базе доменных имен TCP/IP. Так, например, если компьютеру соответствует доменное имя `larch.threeroomco.com`, Nettalk назначит ему имя `larch`. Информация о домене при этом будет утеряна. (Если сеть разбита на зоны, имя зоны также генерируется автоматически, но оно не имеет никакого отношения к имени домена TCP/IP.) Двухкомпонентные имена существенно ограничивают размеры AppleTalk-сетей, в частности, создать сеть, насчитывающую больше нескольких тысяч компьютеров, затруднительно.

Основное назначение AppleTalk — обеспечение совместного использования файлов и принтеров. Многие сетевые принтеры могут непосредственно взаимодействовать посредством протокола AppleTalk, а средства разделения файлов поддерживаются в MacOS, Windows NT и 2000, Linux, BeOS и других операционных системах. Для решения других задач AppleTalk используется лишь в сетях, состоящих из компьютеров, которые работают под управлением MacOS. В сетях, компоненты которых используют иные операционные системы, целесообразнее применять другие стеки протоколов. Если в состав сети входят различные машины, на компьютерах Macintosh устанавливают систему MacOS X, обеспечивающую работу с NFS. Пакет Netatalk (`http://nettalk.sourceforge.net`), используемый для поддержки AppleTalk в Linux, будет рассматриваться в следующем разделе.

При маршрутизации пакетов AppleTalk с помощью обычных маршрутизаторов возникают серьезные трудности. Чтобы исключить возможность взлома извне, можно запретить поддержку TCP/IP на сервере Netatalk (подобные действия имеют смысл только в том случае, если вы абсолютно уверены, что никто из пользователей локальной сети не предпримет попытку взлома, пользуясь недостатками в системе защиты Netatalk). Очевидно, что эта мера обеспечения безопасности не является единственно возможной. Средства защиты сетей TCP/IP будут подробно обсуждаться в части IV.

Программы для поддержки AppleTalk в системе Linux

Пакет Netatalk, поставляемый в составе большинства дистрибутивных пакетов Linux, предназначен для поддержки сетевого взаимодействия посредством AppleTalk. В состав этого пакета входят три основных компонента.

- **Файловый сервер AppleTalk.** Программа `afpd` обеспечивает функционирование компьютера под управлением Linux в качестве файлового сервера. В роли клиентов в данном случае могут выступать системы Macintosh. Файловый сервер поддерживает как AppleTalk, так и TCP/IP, таким образом, Linux может обслуживать даже старые компьютеры Macintosh, работающие с совместимыми аппаратными средствами. (Если соответствующие сетевые аппаратные средства не поддерживаются, можно использовать преобразователи LocalTalk — Ethernet.) Настройка сервера осуществляется с помощью файла `afpd.conf`, который обычно располагается в каталоге `/etc/atalk`. Для контроля разделяемых каталогов используется файл `AppleVolumes.default`, а файл `AppleVolumes.system` отображает расширения файлов в типы Macintosh, предназначенные для сохранения в файловой системе MacOS.

- **Сервер печати AppleTalk.** Программа papd реализует на компьютере Linux сервер печати для систем Macintosh. В сочетании с Ghostscript (компонентом стандартной очереди печати Linux) papd позволяет использовать недорогой струйный принтер как полнофункциональное PostScript-устройство и решать с его помощью достаточно сложные задачи, связанные с отображением документов. Средства, реализующие сервер печати, могут работать только с AppleTalk и не поддерживают TCP/IP.

- **Клиент печати AppleTalk.** Программа pap позволяет компьютерам под управлением Linux передавать задачи печати на принтеры, поддерживающие AppleTalk, или на

серверы печати. Эта возможность становится полезной тогда, когда система Linux работает в сети, состоящей в основном из компьютеров Macintosh, в которой используются принтеры, не поддерживающие другие протоколы. Вы даже можете обращаться с помощью данного инструмента к другим компьютерам под управлением Linux и передавать им задания на печать. Однако подобные действия часто бывают не оправданы. Как вы узнаете, прочитав главу 9, собственные средства печати Linux достаточно просты в настройке. Программа pap не использует конфигурационный файл; принтер, на который следует передать задание на печать, указывается с помощью опции -p. Так, например, команда `pap -p Laser2 sample.ps` означает, что файл sample.ps должен быть выведен на принтер Laser2.

Работа первых двух из описанных выше инструментов базируется на использовании программы `atalkd`, которая представляет компьютер в сети AppleTalk (в частности, она поддерживает AppleTalk-имя и адрес узла). Настройка этой программы производится с помощью конфигурационного файла `atalkd.conf`, который обычно располагается в каталоге `/etc/atalk`.

Netatalk не содержит клиентских программ, предназначенных для разделения файлов, поэтому из системы Linux нельзя обращаться к файлам AppleTalk. Такую возможность предоставляет версия `1.03b-alpha` пакета `afpfs`, но этот инструмент выпущен очень давно и работает ненадежно. Если вам необходимо, чтобы система Linux работала с файлами, расположенными на компьютерах Macintosh, можете воспользоваться для этого средствами NFS или SMB/CIFS, например установить в системе MacOS NFS-сервер или DAVE (`http://www.thursby.com`).

Как правило, установленные средства поддержки AppleTalk работают корректно, но по умолчанию они настроены так, что разделяемым становится только рабочий каталог пользователя. Для того чтобы изменить конфигурацию, надо внести изменения в файл `AppleVolumes.default`. Например, приведенные ниже две строки из этого файла сообщают системе о том что экспортироваться должен как рабочий каталог пользователя (запись, состоящая из символа ~), так и каталог `/mnt`.

```
~
/mnt "Mount Points" options=noadouble
```

Первая строка не содержит опций. Во второй строке указано имя, которое должно предоставляться клиенту Macintosh вместо `/mnt`, а также ключевое слово `options`, посредством которого задаются специальные опции. В данном случае указана единственная опция `noadouble`, которая означает, что файлы AppleDouble не должны создаваться, за исключением тех случаев, когда они абсолютно необходимы. (AppleDouble — специальные файлы, которые находятся в каталоге .AppleDouble и содержат данные, специфические для MacOS.)

Если Netatalk поставляется в составе дистрибутивного пакета, его компоненты, скорее всего, будут автоматически запускаться при загрузке операционной системы. Если запуск Netatalk не предусмотрен, вы можете воспользоваться SysV или локальным сценарием запуска. (Подробно процедура запуска серверов описана в главе 4.) В первую очередь следует запустить `atalkd`, а затем `afpd` и `papd`. Одна из особенностей Netatalk состоит в том, что для запуска `atalkd` требуется достаточно длительное время; при использо-

вании старых аппаратных средств оно может превышать одну минуту. Чтобы устранить задержку, надо включить в сценарий запуска после вызова программы символ &.

IPX/SPX

IPX (Internetwork Packet Exchange — межсетевой обмен пакетами) был разработан специалистами Novell как низкоуровневый транспортный протокол. При создании протокола разработчики основывались на ранних работах, выполненных компанией Xerox. Чаще всего IPX используется совместно с протоколом SPX (Sequences Packet Exchange — упорядоченный обмен пакетами). IPX и SPX составляют основы стека протоколов, возможности и популярность которого сопоставимы со стеками AppleTalk и NetBEUI. Традиционно протоколы IPX/SPX используются в продуктах NetWare, кроме того, в DOS, Windows и других операционных системах применяются программные пакеты, созданные на базе IPX/SPX. Одно из самых распространенных применений IPX/SPX — поддержка протокола NCP (NetWare Core Protocol — базовый протокол NetWare), используемого для разделения файлов и принтеров. Протоколы IPX/SPX поддерживаются в системе Linux; для этого используются как средства ядра (настройка ядра обсуждалась в главе 1), так и клиентские и серверные пакеты.

Возможности IPX/SPX

Подобно TCP/IP и AppleTalk, в IPX/SPX используются 32-разрядные адреса, которые обычно представляются в шестнадцатеричном виде, например 0x23a91002. Однако каждый адрес ставится в соответствие не одному компьютеру, а сегменту сети, который либо соединяется с другими сегментами с помощью маршрутизаторов, либо полностью изолирован от внешнего мира. В описании сети также указывается тип кадра, передаваемого на нижнем уровне; в сети IPX/SPX могут использоваться только кадры одного типа. Для идентификации отдельных компьютеров применяются аппаратные адреса узлов сети, на базе которой создается сеть IPX/SPX, например, если сеть IPX/SPX создана на основе Ethernet, то компьютеры, подключенные к ней, идентифицируются с помощью 48-разрядных адресов.

Как легко догадаться по названию и применяемой схеме адресации, протоколы IPX/SPX разработаны для организации межсетевого обмена. Такой обмен обеспечивают IPX-маршрутизаторы, которые функционируют подобно маршрутизаторам TCP/IP. (При необходимости одна система может маршрутизировать пакеты IP и IPX. В изолированной сети маршрутизатор не нужен, но программы поддержки IPX/SPX включают соответствующие средства.

Серверы IPX/SPX используют протокол SAP (Service Advertisement Protocol — протокол объявления служб). Посредством этого протокола в сети периодически объявляются имя сервера и услуги, которые он предоставляет. Эти сообщения принимаются локальными компьютерами, а IPX-маршрутизаторы передают их в другие сегменты сети. Таким образом клиенты постоянно имеют информацию о доступных серверах, но в то же время при увеличении размеров сети этот подход приводит к возрастанию сетевого трафика, так как по сети постоянно передаются широковещательные SAP-сообщения.

Программы поддержки IPX/SPX в системе Linux

Как и большинство Linux-программ, средства поддержки IPX/SPX в основном распространяются в исходных кодах. (Caldera лицензировала NetWare, и специалисты компании реализовали в Linux поддержку взаимодействия с этой системой, однако сопровождение данного порта прекращено. Версия для трех пользователей доступна по адресу `ftp://ftp.calderasystems.com/pub/old-products/netware/`,но она может работать только с ядром 2.035.) Средства поддержки IPX/SPX для Linux перечислены ниже.

- **Поддержка NCPFS ядром системы.** В состав ядра Linux входят средства поддержки файловой системы NCP. Соответствующие опции находятся в подменю Network File Systems меню File Systems. Эти средства позволяют монтировать в системе Linux разделяемые каталоги NetWare. Для монтирования используется программа ncpmount, которая обычно входит в состав пакета ncpfs.

- **LinWare.** Этот пакет обеспечивает ограниченную поддержку сервера NCP. На момент написания данной книги последней была версия 0.95 beta, ориентированная на работу с ядром 1.3.x, другими словами, пакет не обновлялся с 1996 года. Однако в будущем ситуация может измениться. В настоящее время данный пакет хранится под именем `lwared` по адресу `ftp://sunsite.unc.edu/pub/Linux/system/network/daemons/`.

- **Mars_nwe.** Этот пакет, реализующий NetWare-сервер в системе Linux, в настоящее время доступен по адресу `http://www.compu-art.de/mars_nwe/`. Информация о пакете в основном представлена на немецком языке. Английская документация ограничивается документом `Mars_nwe HOWTO`, который можно получить, обратившись по адресу `http://www.redhat.com/support/docs/tips/Netware/netware.html`. Mars_nwe поддерживает как файловый сервер, так и сервер печати. Настройка пакета осуществляется посредством конфигурационного файла `/etc/nwserv.conf` или `/etc/nwserv/nwserv.conf`. Если Mars_nwe не запускается при загрузке системы, для его запуска можно использовать команду nwserv.

Все перечисленные выше пакеты требуют, чтобы средства поддержки IPX были скомпилированы в составе ядра (о включении компонентов ядра рассказывалось в главе 1). В ряде систем используется специальный пакет `ipxutils`, в состав которого входят утилиты, предназначенные для активизации стека протоколов IPX/SPX и управления им. (В некоторых версиях эти утилиты содержатся в пакете ncpfs.)

Если вы хотите, чтобы система Linux выступала в роли сервера для клиентов NetWare, надо выполнить несложную настройку Mars_nwe. Более того, конфигурация, установленная по умолчанию, обычно обеспечивает нормальную работу данного пакета. Конфигурационные файлы снабжены подробными комментариями; прочитав их, можно получить полное представление о процессе настройки пакета. Особое внимание надо уделить следующим деталям.

- В разделе 1 конфигурационного файла определяются разделяемые тома. В терминах Linux эти тома соответствуют каталогам. В зависимости от поставки, разделяемые тома могут быть либо определены, либо нет.

- В разделе 7 содержатся опции, управляющие шифрованием пароля. Если в вашей сети отсутствует bindery-сервер NetWare, поддерживающий аутентификацию пользователей, вам необходимо задать передачу пароля в незашифрованном виде.
- В разделе 13 определены пользователи, которым разрешен доступ к серверу. В этот раздел вам придется включить имена пользователей и пароли, дублируя соответствующие настройки Linux. Пароли хранятся в незашифрованном виде, что создает угрозу безопасности системы. Если в сети присутствует bindery-сервер, после первого запуска Mars_nwe вы можете удалить эти записи, так что риск становится минимальным. Вместо того чтобы задавать имена пользователей и пароли, соответствующие отдельным учетным записям, вы можете указать в разделе 15 автоматическое выполнение этих действий. Недостаток такого подхода состоит в том, что для всех учетных записей будет установлен один и тот же пароль.

Пакет Mars_nwe содержит средства, позволяющие автоматически включать IPX-поддержку для сетевого интерфейса. Однако эти средства не действуют при работе с клиентами NetWare. Перед использованием команды ncpmount вам надо разрешить автоконфигурацию, вызвав команду `ipx_configure`. Затем вы можете монтировать том NetWare. Команды, реализующие описанную процедуру, выглядят следующим образом:

```
# ipx_configure --auto_interface=on --auto_primary=on
# ncpmount -S NW_SERV -U anne -P p4rtu3a /mnt/nwmount
```

При выполнении этих команд разрешается автоматическое определение номера локальной сети, и том на NW_SERV, связанный с пользователем anne, монтируется в точке /mnt/nwmount, при этом используется пароль p4rtu3a.

NetBEUI

NetBEUI во многом напоминает AppleTalk и IPX, однако средства NetBEUI в основном используются IBM и Microsoft для организации сетевого взаимодействия в системах DOS, Windows и OS/2. В системе Linux (по крайней мере в версиях ядра 2.4.x) стек NetBEUI не поддерживается. Тем не менее возможности NetBEUI могут быть реализованы средствами NetBIOS на базе TCP/IP, которые присутствуют в Linux (такую конфигурацию NetBIOS часто называют NBT). Кроме того, поддержка стека NetBEUI обеспечивается продуктами независимых производителей, но на сегодняшний день такие продукты находят лишь ограниченное применение.

Возможности NetBEUI

Подобно AppleTalk и IPX, стек NetBEUI был разработан для обеспечения взаимодействия в небольших сетях. Сеть под управлением NetBEUI может насчитывать не больше 256 компьютеров. В NetBEUI используются имена, подобные доменным именам TCP/IP, но, в отличие от TCP/IP, AppleTalk и IPX, числовая адресация не применяется. В NetBEUI компьютеры идентифицируются лишь с помощью имен. Имена NetBEUI состоят из двух компонентов: имени компьютера и имени группы. В зависимости от наличия централизованного сервера, предназначенного для управления регистрацией пользователя, группа компьютеров называется *рабочей группой* либо *доменом*. С момента включения компьютер, настроенный для взаимодействия посредством NetBEUI, передает широковещательные сообщения, свидетельствующие о его присутствии.

Средства NetBEUI могут работать практически в любой сетевой среде, но чаще всего сети, использующие этот стек, создаются на базе Ethernet. Подобно AppleTalk и IPX, NetBEUI может использоваться вместе с TCP/IP и другими стеками протоколов.

Средства NetBEUI очень часто применяются с протоколами SMB/CIFS, обеспечивающими разделение файлов и принтеров. По степени популярности их можно сравнить с NFS/lpd для Linux и UNIX или NCR SMB/CIFS могут работать на базе TCP/IP; такая конфигурация используется чрезвычайно часто, даже в сетях, состоящих исключительно из компьютеров под управлением Windows. Маршрутизация сообщений NetBEUI затруднена, и это обеспечивает дополнительную степень защиты сетей. В настоящее время неизвестны средства, с помощью которых злоумышленник, работающий на удаленном компьютере, мог бы незаконно получить доступ к серверу NetBEUI.

Средства поддержки NetBEUI для Linux

Компьютеры под управлением Linux редко участвуют в NetBEUI-взаимодействии, так как в стандартном ядре отсутствует поддержка этого стека. В 2000 г. силами Procom Technologies (`http://www.procom.com`) были реализованы средства поддержки NetBEUI для Linux, а также дополнения к Samba (подробно об этом рассказывается в главе 7), обеспечивающие работу Samba посредством NetBEUI. Эти дополнения не нашли широкого применения; они даже не были размещены на Web-узле Procom. Для того чтобы получить необходимые программные средства, надо обратиться в отдел технической поддержки компании. Совместно с ядрами, отличными от версий 2.0.x, средства поддержки NetBEUI могут работать некорректно (лично мне не удалось найти программы, совместимые с ядром 2.2.18). Стек NetBEUI ориентирован на работу с версиями Samba, предшествующими 2.0.7, однако были сообщения о том, что впоследствии будет реализована поддержка Samba 3.0. Для работы с NetBEUI необходимо перекомпилировать как ядро Linux, так и Samba. Таким образом, для того, чтобы устанавливать средства поддержки NetBEUI в Linux, надо иметь достаточно веские основания, например, делать это имеет смысл тогда, когда в сети запрещена поддержка стека протоколов TCP/IP. Для работы с NetBEUI вам придется использовать ядро Linux 2.0.x и Samba 2.0.6 либо более раннюю версию этого продукта.

Кроме дополнений к ядру Linux и Samba средства поддержки стека NetBEUI содержат большое количество специальных утилит, предназначенных для настройки стека и выполнения различных действий в сети NetBEUI. Настройка системы обычно не вызывает затруднений; в большинстве случаев для управления поведением NetBEUI можно использовать опции Samba. Среди утилит следует выделить `netb`; с ее помощью запускаются средства поддержки стека NetBIOS.

Использование программ поддержки NetBEUI

Пакет, предназначенный для поддержки стека NetBEUI, содержит файл README, в котором полностью описан процесс инсталляции. Установка пакета может быть выполнена двумя способами. Следуя одному из них, надо отредактировать файл Makefile, указав в нем ссылки на каталоги, содержащие коды ядра Linux и исходные тексты Samba, а также установив некоторые опции, специфические для конкретной системы. Затем необходимо скомпилировать ядро Linux и коды Samba, установить новое ядро и перезагрузить систему. Второй способ инсталляции также требует редактирования Makefile, но при этом надо предоставить системе детальные инструкции о выполнении каждого шага установ-

ки. Второй способ предпочтительнее первого, особенно если при установке возникают затруднения. В этом случае вы можете локализовать проблему и устранить ее.

Каким бы способом вы ни воспользовались, вам потребуются исходные коды как ядра Linux, так и пакета Samba. Найти их вы сможете, обратившись по адресам `http://www.kernel.org` и `http://www.samba.org`, а также на других узлах, содержащих архивы программ, например `ftp://sunsite.unc.edu`. Компиляция каждого пакета займет несколько минут. Если же при этом возникнут проблемы, то для установки средств поддержки NetBEUI вам потребуется значительно больше времени.

После окончания процесса инсталляции для включения средств поддержки NetBEUI и управления взаимодействием по сети можно воспользоваться перечисленными ниже утилитами.

- `netb`. Передавая этой утилите параметр `start`, вы можете запустить средства поддержки NetBEUI. Чтобы запретить поддержку NetBEUI, надо выполнить команду `netb stop`. Использовать NetBEUI в Linux можно лишь после вызова утилиты `netb`.
- nbview. Эта утилита сообщает сведения о текущем состоянии локального стека NetBEUI. Она считает файл `/proc/sys/netbeui`, в котором содержатся соответствующие данные, отформатирует их и отобразит в виде, удобном для восприятия.
- `nbstatus`. Утилита `nbstatus` предоставляет информацию о конкретной машине в рабочей группе. Например, по команде `nbstatus` SERVER будут выведены данные о компьютере с именем SERVER.
- `nbadmin`. Данная утилита позволяет связывать NetBEUI с конкретным сетевым интерфейсом, разрывать связь NetBEUI с интерфейсом или прекращать указанный сеанс NetBEUI-взаимодействия. Для этого используются параметры `bind`, `unbind` и `drop`. Например, соответствующая команда может иметь вид `nbadmin bind eth0` или `nbadmin drop 102`. (Получить номер сеанса можно с помощью утилиты nbview.)

В большинстве случаев необходимо задавать только команду `netb start`, а затем запускать Samba. Средства NetBEUI добавляют новые параметры к `nmbd` (поддержка имен NetBIOS), `smbd` (программа поддержки SMB) и `smbclient` (клиент Samba, работающий в текстовом режиме). Один из параметров имеет вид `-Z` <NETBEUI | TCPIP> и указывает, следует ли использовать TCP/IP или NetBEUI. Так, например, чтобы запустить smbd для поддержки NetBEUI, надо ввести команду `smbd -Z NETBEUI`. Кроме того, новый параметр `-S` ИМЯ программы smbd позволяет задать NetBEUI-имя системы.

Таким образом, вы можете превратить узел сети, работающий под управлением Linux, в сервер NetBEUI SMB/CIFS с именем NAME. Для этого надо скомпилировать ядро и коды Samba, включив в них средства NetBEUI, разработанные Procom, перезагрузить систему (возможно, для этого придется завершить работу Samba) и выполнить следующие команды:

```
# netb start
# nmbd -Z NETBEUI
# smbd -Z NETBEUI -S имя
```

Вы можете поместить приведенные выше команды в сценарий запуска системы или модифицировать соответствующим образом сценарий запуска Samba. Остальные средства Samba функционируют так, как описано в главе 7. Установив средства поддержки NetBEUI, вы добиваетесь следующих результатов. Во-первых, обеспечивается работа NetBEUI-клиентов, не поддерживающих TCP/IP, а во-вторых, снижается вероятность незаконного обращения к серверу, поскольку в нормальных условиях данные NetBEUI не маршрутизируются и не могут быть переданы по Internet. Описанные выше программы дублируют средства, обеспечивающие работу NetBIOS на базе TCPЯP. Эти средства присутствуют в последних версиях ядра Linux и Samba и для их использования не требуется устанавливать дополнительные модули и перекомпилировать программы.

Резюме

Стек протоколов является основой для работы других сетевых инструментов, например клиент-программ и серверов. Для того чтобы два компьютера могли взаимодействовать по сети, на них должны быть реализованы совместимые стеки протоколов. В настоящее время наибольшей популярностью пользуется стек протоколов TCP/IP. TCP/IP составляет базу Internet, на этом же стеке протоколов основывается работа большинства сетевых средств Linux. Однако, помимо TCP/IP, существуют и другие стеки протоколов. В предыдущие годы наиболее часто применялись AppleTalk, IPX и NetBEUI. Эти стеки в основном ориентированы на использование в локальных сетях; с их помощью, как правило, организуется совместный доступ к файлам и принтерам. Указанные стеки протоколов находят применение и в настоящее время, в частности, они используются при создании небольших сетей. В системе Linux реализована ограниченная поддержка этих стеков. Для обеспечения взаимодействия посредством NetBEUI необходимо перекомпилировать ядро Linux и коды Samba, включив в них дополнительные модули.

Глава 4

Запуск серверов

В основном данная книга (в особенности части II и III) посвящена работе различных серверов. Как правило, программы-серверы начинают работать с момента загрузки компьютера, на котором они установлены, и постоянно предоставляют свои услуги клиентам. В некоторых случаях планируются перерывы в работе серверов, связанные с необходимостью выполнения работ по обслуживанию компьютера; кроме того, доступ к тому или иному серверу может быть ограничен по соображениям безопасности. Администрируя локальную сеть, необходимо ясно представлять себе процедуру запуска серверов, в противном случае вы не сможете запустить сервер после установки или перезапустить его в случае изменения конфигурации.

Чаще всего серверы, установленные в системе Linux, начинают работать сразу же после инсталляции; в редких случаях для их запуска приходится перезагрузить компьютер. Существуют три основных способа запуска серверов: использование сценариев запуска System V (SysV), настройка *суперсервера,* например `inetd` или `xinetd`, или применение локальных сценариев запуска. В большей части дистрибутивных пакетов содержатся инструментальные средства с графическим пользовательским интерфейсом, позволяющие выполнять подобные задачи. В данной главе рассматриваются все три метода запуска серверов. Если вы не будете знать их, у вас могут возникнуть затруднения при изучении материала, изложенного в последующих главах.

Использование сценариев запуска SysV

Многие технические решения, которые используются в системе System V UNIX, разработанной AT&T, стали стандартом для современных версий UNIX и Linux. Одним из них является способ запуска системных служб, в том числе серверов. Согласно схеме загрузки SysV, каждой службе должен соответствовать специальный сценарий запуска, поддерживающий параметры `start` и `stop`. В зависимости от полученного параметра, сценарий запускает программу поддержки данной службы или завершает ее работу. Многие сценарии запуска поддерживают дополнительные параметры, например, `restart`, используемый при изменении конфигурации программы. При получении параметра `restart` сценарий завершает работу сервера, а затем снова запускает его.

Сценарии SysV используются не только для запуска сетевых серверов. В данной книге для обозначения программ, постоянно выполняющихся в операционной системе и предоставляющих некоторые услуги, но не обязательно доступных по сети, используется термин *служба,* или *демон.* Серверами называются программы, обслуживающие запросы по сети. Большинство служб, которые обсуждаются в этой книге, являются серверами. Многие специалисты не придерживаются данной терминологии, например, называют службами те программы, для обозначения которых здесь используется термин *сервер.*

Схема запуска SysV непосредственно связана с понятием *уровня выполнения* (runlevel). Каждому уровню выполнения соответствует набор сценариев запуска, который определяет службы, выполняющиеся в системе. (Посредством сценариев SysV запускаются не только серверы, но и другие службы, например, средства протоколирования, поддержки файловой системы и прочие программы.) Таким образом, настройка системы для запуска серверов с помощью сценариев SysV по сути сводится к выбору конфигурации уровней выполнения. Для решения данной задачи надо создать ссылки на требуемые сценарии и поместить их в каталог, соответствующий требуемому уровню выполнения.

Расположение сценариев запуска и соглашения по их именованию

Несмотря на то что основные принципы использования сценариев запуска SysV соблюдаются во всех системах, особенности такого использования могут различаться в зависимости от конкретного дистрибутивного пакета. В разных системах сценарии запуска размещаются в различных каталогах, имена сценариев могут различаться, но эти различия, как правило, не существенны. В табл. 4.1 описаны каталоги, используемые разными системами при работе со сценариями запуска SysV. Обратите внимание на то, что в табл. 4.1 указано размещение реальных сценариев, ссылок на эти сценарии, соответствующих определенным уровням выполнения, а также расположение локальных сценариев запуска (подробно локальные сценарии будут рассматриваться далее в данной главе). В именах ссылок на сценарии символом ? обозначается число, соответствующее уровню выполнения (от 0 до 6).

Уровень выполнения — это число от 0 до 6, которому соответствует конкретный набор действующих служб. Уровни выполнения будут более подробно описаны в одном из последующих разделов, сейчас же вам достаточно знать, что при загрузке компьютер переходит на некоторый уровень выполнения, на котором выполняется определенный набор сценариев запуска. При необходимости вы можете изменить уровень выполнения после загрузки компьютера.

Далее в этой главе будут упоминаться каталоги сценариев SysV и каталоги ссылок. Если вам потребуется изменить некоторые файлы, вы должны знать, к каким каталогам необходимо обращаться.

В некоторых системах сценарии запуска используются абсолютно одинаково. К этим системам относятся Red Hat, Mandrake, TurboLinux; несколько отличается от них Caldera. В них сценарии запуска расположены в каталоге `/etc/rc.d/init.d`, а ссылки на сценарии — в каталоге `/etc/rc.d/rc?.d`. В других системах, а особенно в Slackware,

Таблица 4.1. Сценарии запуска для основных дистрибутивных пакетов Linux

Система	Сценарий запуска	Каталог для размещения сценариев **SysV**	Каталог для размещения ссылок на сценарии SysV	Локальный сценарий запуска
Caldera OpenLinux Server 3.1	`/etc/rc.d/rc.boot`	`/etc/rc.d/init.d`	`/etc/rc.d/rc?.d`	`/etc/rc.d/rc.local`
Debian GNU/ Linux 2.2	`/etc/init.d/rcS`	`/etc/init.d`	`/etc/rc?.d`	Файлы в каталоге `/etc/rc.boot`
Linux Mandrake 8.1	`/etc/rc.d/rc.sysinit`	`/etc/rc.d/init.d`	`/etc/rc.d/rc?.d`	`/etc/rc.d/rc.local`
Red Hat Linux 7.2	`/etc/rc.d/rc.sysinit`	`/etc/rc.d/init.d`	`/etc/rc.d/rc?.d`	`/etc/rc.d/rc.local`
Slackware Linux 8.0	`/etc/rc.d/rc.S`	`/etc/rc.d`	Не используется	Различные файлы в каталоге `/etc/rc.d`
SuSE Linux 7.1	`/etc/init.d/boot`	`/etc/rc.d`	`/etc/rc.d/rc?.d`	`/etc/rc.d/boot.local`
TurboLinux 7.0	`/etc/rc.d/rc.sysinit`	`/etc/rc.d/init.d`	`/etc/rc.d/rc?.d`	`/etc/rc.d/rc.local`

сценарии расположены по-иному. Вместо того чтобы размещать ссылки на сценарии в каталогах, имена которых соответствуют уровням выполнения, Slackware использует для каждого уровня выполнения один сценарий. Так, например, сценарий `/etc/rc.d/rc.4` управляет запуском служб на уровне выполнения 4.

В большинстве дистрибутивных пакетов Linux (за исключением Slackware) действуют строгие правила по именованию содержимого каталога ссылок SysV. Имя файла ссылки имеет вид ***C##имя***, где C обозначает символ "S" или "K", ## — это число, состоящее из двух цифр, а *имя* обычно совпадает с именем соответствующего файла в каталоге сценариев. Например, сценариям network и nfs соответствуют файлы-ссылки `S10network` и K20nfs. Как нетрудно догадаться, принцип именования несет на себе определенную смысловую нагрузку. Часть имени ссылки, следующая за ***C##***, дает представление о действиях, выполняемых сценарием. Первый симол ("S" или "K") указывает, должен ли сценарий запускать программу ("S" — start) или завершать ее работу ("K" — kill) при переходе на данный уровень выполнения. Например, имя SlOnetwork означает, что сценарий network должен быть вызван для запуска соответствующих служб (в данном случае основных программ, обеспечивающих сетевое взаимодействие), а имя `K20nfs` говорит о том, что работа программ, запущенных с помощью сценария nfs (сервера NFS) должна быть завершена. Число, следующее за символом "S" или "K", определяет порядок запуска сценариев. Например, программы поддержки сетевого взаимодействия, которым соответствует ссылка SlOnetwork, будут запущены раньше, чем сервер SSH (ссылка S55sshd). Аналогично определяется последовательность активизации ссылок, имена которых начинаются с символа "K".

Имена ссылок, используемых для запуска и прекращения работы служб, могут различаться в зависимости от конкретного дистрибутивного пакета. Например, в системе Mandrake программы, поддерживающие основные сетевые средства, запускаются с помощью ссылки SlOnetwork, а в Debian для той же цели используется ссылка S35networking. Подобные различия имеют место для сценариев, запускающих конкретные серверы. В данном случае важен тот факт, что серверы, которые должны присутствовать в системе, запускаются в определенном порядке. Например, многие сетевые инструменты могут быть запущены лишь тогда, когда программы поддержки базовых сетевых средств уже начали работу. Менять порядок запуска и завершения работы служб можно лишь в случае крайней необходимости. Если вы решитесь сделать это, вы должны ясно представлять себе функции, выполняемые каждой из служб, и последствия, которые повлечет за собой изменение порядка вызова сценариев.

Говоря о различных дистрибутивных пакетах, следует отметить одну особенность системы SuSE. В этой системе для управления процессом запуска сценариев SysV используется файл `/etc/rc.config`. Разделы этого файла посвящены основным серверам, которые запускаются средствами SysV. Если сервер не указан в этом файле (о том, что сервер должен быть запущен, сообщает строка `START_`***`ИМЯ_СЕРВЕРА`***`="yes"`), он не будет выполняться в системе, даже если соответствующая ссылка начинается с символа "S". В системе Caldera для некоторых серверов используется подобная схема запуска, а запуском остальных серверов управляют файлы в каталоге `/etc/sysconfig/daemons`. Имя такого файла соответствует имени сервера. Строка ONBOOT в управляющем файле определяет, должен ли сервер выполняться в системе, однако многие из сценариев запуска Caldera не поддерживают данную опцию.

Управление сценариями запуска вручную

Если вам необходимо разрешить или запретить запуск сервера с помощью сценариев SysV, вы можете сделать это, изменяя сценарии запуска или ссылки на них. Проще всего запретить запуск сервера, удалив соответствующий сценарий из каталога сценариев SysV. Этим вы добьетесь того, что сервер не будет присутствовать ни на одном из уровней выполнения системы, но такое решение нельзя назвать элегантным. Кроме того, если вам потребуется не запретить, а разрешить выполнение сервера, вам все равно придется искать способы сделать это.

Более приемлемое решение данной задачи — переименовать ссылку на сценарий запуска в каталоге, соответствующем требуемому уровню выполнения. Например, для того, чтобы запретить выполнение сервера, надо переименовать ссылку, заменив символ "S" в начале ее имени на символ "K". Чтобы разрешить работу сервера, надо сделать обратную замену. Сложности, возникающие при этом, связаны с тем, что последовательность запуска серверов может отличаться от последовательности их завершения. Для того чтобы решить эту проблему, надо найти ссылки на этот сценарий в каталогах, соответствующих различным уровням запуска. Если хотя бы на одном уровне выполняется нужное вам действие, вы узнаете требуемый номер. Например, в результате выполнения приведенной ниже команды отображаются все ссылки на сценарий запуска системы Mandrake, соответствующий почтовому серверу Postfix.

```
$ find /etc/rc.d -name "*postfix"
/etc/rc.d/rc0.d/K30postfix
/etc/rc.d/rc1.d/K30postfix
/etc/rc.d/rc2.d/S80postfix
/etc/rc.d/rc3.d/S80postfix
/etc/rc.d/rc4.d/S80postfix
/etc/rc.d/rc5.d/S80postfix
/etc/rc.d/rc6.d/K30postfix
/etc/rc.d/init.d/postfix
```

Полученные результаты позволяют выяснить, что Postfix запускается на уровнях выполнения 2-5 и порядок запуска этого сервера определяется номером 80. Аналогично, работа сервера завершается на уровнях О, 1 и 6, и порядок завершения определяется номером 30. Если вы хотите запретить выполнение Postfix на уровне 3, вам надо переименовать ссылку S80postfix в каталоге, соответствующем этому уровню, и назначить ей имя K30postfix.

Если вам нужно временно запустить или остановить сервер, не перезагружая компьютер, либо если вы захотите перезапустить сервер после изменения его конфигурационного файла, вы можете вызвать сценарий запуска вручную и передать ему параметр `start` или `stop`. Например, для того, чтобы немедленно прекратить работу сервера Postfix в системе Mandrake, вам надо выполнить следующую команду:

```
# /etc/rc.d/init.d/postfix stop
```

Большинство версий Linux при попытке завершить работу сервера выводит специальное сообщение. Кроме того, после остановки сервера отображается дополнительное сообщение о том, насколько успешно выполнена данная операция. При запуске сценария SysV появляется сообщение об успешном запуске сервера. (Эти сообщения вы видите на экране при загрузке компьютера.)

Если вы работаете с системой Slackware, то вместо переименования, добавления и удаления сценариев запуска или ссылок на них вам надо отредактировать один файл сценария, соответствующий требуемому уровню выполнения. Так, например, чтобы изменить поведение системы на уровне 4, вам надо внести изменения в файл `/etc/rc.d/rc.4`. Заметьте, что для многих серверов сценарии запуска отсутствуют; эти серверы запускаются с помощью `/etc/rc.d/rc.inet2` (программы поддержки базовых сетевых средств, используемых этими серверами, запускаются посредством `/etc/re.d/rc.inet1`). Для того чтобы изменить набор серверов, выполняемых в системе, надо вручную отредактировать эти сценарии так, как будто вы имеете дело с локальными сценариями запуска. (Локальные сценарии запуска будут рассмотрены ниже в этой главе.)

Использование утилит управления сценариями запуска

Некоторые дистрибутивные пакеты включают специальные утилиты, которые упрощают управление сценариями запуска. Пользуясь этими утилитами, вы уменьшаете риск неправильно задать имя сценария. Так, например, изменяя набор серверов, выполняемых в системе, вручную, вы можете вместо `S80postfix` случайно задать имя `s80postfix` (т. е. вместо "S" в верхнем регистре задать "s" в нижнем регистре). При использовании специализированных утилит такая ситуация не возникнет. К сожалению, подобные утилиты присутствуют не во всех системах; чаще всего они входят в состав Red Hat и систем, созданных на ее основе, например Mandrake. Перенос утилиты из одной системы в другую не дает желаемого результата, так как в разных системах расположение и имена сценариев запуска и ссылок SysV, а также номера, определяющие порядок запуска, могут различаться.

В данном разделе рассматриваются утилиты управления сценариями запуска, предоставляющие алфавитно-цифровой интерфейс. Для выполнения подобных задач могут использоваться также утилиты с графическим интерфейсом, которые будут обсуждаться далее в этой главе.

Использование chkconfig

Инструментальное средство `chkconfig`, предназначенное для управления сценариями запуска SysV, предоставляет пользователю низкоуровневый интерфейс. Вся информация, необходимая для выполнения задачи, задается в одной командной строке. Утилита `chkconfig` вызывается следующим образом:

```
chkconfig <--list|--add|--del> [имя]
chkconfig [--level уровни] имя [on|off|reset]
```

Первый вариант вызова используется тогда, когда необходимо получить информацию о текущей конфигурации (опция `-list`), добавить или удалить ссылки из каталога ссылок SysV (соответственно опции `-add` и `-del`). Второй вариант вызова позволяет разрешить или запретить сценарий на некоторых (или на всех) уровнях выполнения (эта задача решается путем переименования ссылки SysV). Приведенные ниже примеры иллюстрируют использование данной команды.

Предположим, что вы хотите получить подробную информацию о конфигурации Postfix. Если вы знаете, что стартовый сценарий Postfix называется `postfix`, вы можете задать следующую команду:

```
# chkconfig --list postfix
postfix      0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

В результате утилита выводит информацию о состоянии Postfix на каждом из уровней выполнения. Проверить правильность полученных данных можно, воспользовавшись командой **find**. Если **chkconfig** отображает значение on, это свидетельствует о том, что имя ссылки начинается с символа "S", соответственно **off** означает, что имя ссылки начинается с символа "K".

Если вы выполните команду **chkconfig -list**, не указав имени сценария, **chkconfig** выведет информацию о состоянии всех сценариев запуска. Если в вашей системе используется **xinetd**, вы, возможно, получите также сведения о серверах, которые запускаются с помощью этого суперсервера.

Опция **-add** добавляет ссылку (если она отсутствует), а опция **-del** позволяет удалить существующую ссылку. Используя эти опции, необходимо указать имя сценария запуска. Например, команда **chkconfig -del postfix** удаляет все ссылки SysV на сценарий, ответственный за запуск сервера Postfix. В результате ее выполнения Linux не будет запускать сервер посредством сценариев SysV, а также не будет предпринимать попыток изменить состояние сервера при переходе на другой уровень выполнения. Удалять ссылки имеет смысл в том случае, если вы собираетесь запускать сервер с помощью суперсервера либо локальных сценариев запуска. Для того чтобы выполнить обратные изменения, надо воспользоваться опцией -add.

Чаще всего при работе с **chkconfig** используются параметры on, **off** и **reset**. Они позволяют разрешить или запретить запуск сервера на указанном уровне выполнения, а также восстановить исходные установки для этого уровня. Если вы не укажете опцию **-level**, то изменения будут произведены на всех уровнях выполнения. Предположим, вам необходимо запретить запуск сервера Postfix на уровне **3**. Сделать это можно с помощью следующей команды:

```
# chkconfig --level 3 postfix off
```

При выполнении этой команды не отображаются никакие данные. Проверить полученные результаты можно, вызвав утилиту **chkconfig** с опцией **-list** или просмотрев содержимое соответствующего каталога ссылок. Для того чтобы разрешить запуск сервера, надо вместо **off** указать параметр on. Если вам необходимо, чтобы действие утилиты распространялось на несколько уровней выполнения, надо указать требуемые уровни выполнения в виде одной строки. Так, например, чтобы изменения были произведены на уровнях 3-5, надо указать значение **345** опции **-level**. Если вы поэкспериментировали с установками и хотите вернуть их в исходное состояние, вам **следует** задать параметр **reset**.

```
# chkconfig postfix reset
```

Эта команда вернет ссылки на сценарии запуска для сервера Postfix в первоначальное состояние. Для того чтобы восстановить установки лишь для отдельных уровней, следует задать опцию **-level** и указать в качестве ее значения один или несколько уровней.

Несмотря на то что **chkconfig** обычно рассматривается как средство управления сценариями SysV, во многих системах эта утилита также может использоваться для настройки **xinetd**. Предположим, что **chkconfig** сконфигурирована таким образом, что она воспринимает сервер FTP как программу, запускаемую посредством суперсервера. В этом случае вы можете применять эту утилиту для изменения конфигурации FTP так,

как будто для запуска данного сервера используются сценарии SysV. При этом опция `-level` не работает, а при указании опции `-list` не отображается информация об уровнях выполнения. Любой сервер, запускаемый с помощью суперсервера, будет функционировать на тех уровнях выполнения, на которых запускается `xinetd`. Опции `-add` и `-del` действуют подобно параметрам `on` и `off`. Конфигурационные файлы `/etc/xinetd.d` не удаляются, но их использование запрещается. Подробно работа `xinetd` будет рассмотрена далее в этой главе.

При изменении конфигурации SysV посредством `chkconfig` состояние выполняющихся в системе серверов не изменяется. Например, если вы запретили запуск `sshd`, работа сервера не завершится. Чтобы это произошло, надо предпринять дополнительные действия, например, вызвать сценарий запуска SysV и передать ему опцию `stop` либо вручную остановить сервер.

Использование ntsysv

Программа `ntsysv` предоставляет пользователю интерфейс в виде текстового меню и позволяет управлять запуском серверов. Для того чтобы запустить программу, достаточно ввести ее имя, при необходимости можно задать опцию `-level` *уровни; в* качестве значения опции `-level` указывается один или несколько уровней выполнения, которые вы хотите изменить. Если вы не зададите эту опцию, `ntsysv` изменит только конфигурацию текущего уровня. Внешний вид меню `ntsysv` показан на рис. 4.1.

Программа `ntsysv` отображает сведения обо всех серверах, для которых созданы сценарии запуска SysV. Некоторые версии `ntsysv` также выводят данные о серверах, запускаемых с помощью `xinetd`. Для того чтобы разрешить или запретить запуск сервера, надо с помощью клавиш со стрелками выбрать сервер в меню и нажать клавишу пробела. Символ * слева от имени сервера указывает на то, что при переходе на данный уровень выполнения сервер будет запущен; отсутствие этого символа означает, что запуск сервера запрещен. После внесения изменений надо с помощью клавиши <Tab> выбрать кнопку

Рис. 4.1. Программа `ntsysv` предоставляет пользователю простой интерфейс для настройки сценариев SysV

OK и нажать клавишу <Enter>; в результате изменения будут сохранены, и выполнение программы завершится.

С помощью ntsysv вы можете задавать уровни выполнения, на которых будут работать серверы, запускаемые с помощью суперсервера; данная программа не позволяет лишь изменять уровни выполнения для самого суперсервера. Запрет загрузки сервера не означает, что этот сервер немедленно прекратит работу. Чтобы это произошло, вам надо вручную завершить выполнение сервера либо, если этот сервер запускается посредством суперсервера, перезапустить суперсервер.

Управление уровнями выполнения

В предыдущих разделах постоянно упоминались уровни выполнения, но из сказанного вряд ли стало ясно, что же они собой представляют. Говорилось лишь о том, что уровни выполнения и сценарии запуска SysV тесно связаны между собой. При загрузке компьютер переходит на некоторый уровень выполнения. Этому уровню соответствует каталог ссылок SysV; содержащиеся в нем ссылки указывают на сценарии запуска. Если ссылка начинается с символа "S", Linux при вызове сценария передает ему параметр start, а если имя ссылки начинается с "K", сценарию передается параметр stop.

Но как Linux узнает, на какой уровень следует перейти после загрузки? Информация об этом хранится в файле /etc/inittab, который выполняет роль конфигурационного файла для init — первого процесса, выполняющегося в системе. Процесс init является родительским для всех остальных процессов в системе. В файле /etc/inittab содержатся записи наподобие приведенной ниже.

```
id:5:initdefault:
```

Ключевое слово id, расположенное в начале, идентифицирует данную строку, а число, следующее за ним (в данном случае 5), устанавливает постоянный уровень выполнения. Если вы измените это значение и перезагрузите компьютер, система будет работать на другом уровне. Уровни О, 1 и 6 имеют специальное назначение. Уровень 0 соответствует завершению работы системы, уровень 1 — однопользовательскому режиму, а уровень 6 — перезагрузке системы. Уровни 2-5 задают нормальные режимы работы; назначение каждого из уровней может изменяться *в* зависимости от версии системы. В Caldera, Red Hat Mandrake SuSE7.3 и TurboLinux уровень 3 соответствует обычному текстовому режиму (система X Window не запускается), а уровень 5 поддерживает графический пользовательский интерфейс (система X Window запущена). В ранних версиях SuSE вместо уровней 3 и 5 для поддержки текстового режима и графического интерфейса используются уровни 2 и 3, а в Slackware для той же цели применяются уровни 3 и 4. По умолчанию в Debian на уровнях 2-5 набор серверов, запускаемых посредством сценариев SysV, существенно не отличается, но на уровнях выше третьего используется меньшее число инструментов с текстовым интерфейсом (детали настройки системы можно выяснить, просмотрев содержимое файла /etc/inittab). В большинстве систем файл /etc/inittab содержит подробные комментарии, которые описывают функциональные возможности каждого из уровней. Если вы используете версию системы, которая не обсуждается в данной книге, или если вам нужна дополнительная информация о работе системы на различных уровнях, вы можете получить требуемые сведения, просмотрев комментарии в этом файле.

ВНИМАНИЕ Не устанавливайте в качестве уровня по умолчанию уровень 0 или 6. Если вы поступите так, то сразу после загрузки работа системы будет завершена либо компьютер начнет перезагружаться. Для того чтобы изменить настройку, вам придется загрузить компьютер с другого диска.

Если вы хотите временно изменить уровень выполнения, сделайте это с помощью команды **telinit** (в некоторых системах для этого приходится вызывать **init**). Синтаксис **telinit** имеет следующий вид:

```
telinit [-t время_в_секундах] [уровень]
```

При изменении уровня выполнения некоторые процессы могут быть завершены. Для завершения процесса Linux передает ему сигнал **SIGTERM** либо SIGKILL. Сигнал **SIGTERM** обеспечивает более "мягкий" режим окончания работы; при этом программа получает возможность закрыть файлы и освободить другие ресурсы. SIGKILL принудительно завершает выполнение программы, в результате файлы, используемые в процессе его работы, могут быть повреждены. При изменении уровня выполнения **telinit** сначала пытается использовать SIGTERM. Если процесс продолжает выполняться, то через пять секунд **telinit** передает ему сигнал SIGKILL. Опция -t позволяет изменить этот интервал. В большинстве случаев значение, равное пяти секундам, вполне приемлемо.

Второй параметр, передаваемый **telinit**, задает уровень выполнения. Для указания уровня используется один символ. Результаты, которые вы получите, задавая в качестве этого параметра число, очевидны. Кроме того, вы можете передать программе другие символы. Их назначение описано ниже.

- a, b или c. Некоторые записи в файле /etc/inittab идентифицируются с помощью **символов** a, b и c. Эти символы имеют специальное назначение. Если вы передадите один из них **telinit**, программа будет обрабатывать соответствующие ему записи /etc/inittab; при этом уровень выполнения системы не изменится.
- Q или q. Если задать одно из этих значений как уровень выполнения, **telinit** повторно считает содержимое файла **/etc/inittab** и продолжит работу с учетом внесенных изменений.
- **S**, или **s**. Эта опция переводит систему в однопользовательский режим.
- U, или u. Данная опция вызывает перезагрузку процесса **init**; при этом новое содержимое файла /etc/inittab не считывается.

Зачем может понадобиться переходить на другой уровень выполнения? Изменяя уровень выполнения по умолчанию, вы можете изменять набор серверов, работающих в системе. В большинстве дистрибутивных пакетов самым важным считается сервер X Window. Одна из последних записей в файле /etc/inittab управляет запуском этого сервера; в некоторых системах эта задача решается с помощью сценариев запуска SysV. Изменение уровня выполнения позволяет быстро перейти от одного набора сервера к другому, разрешить или запретить графический режим или временно отключить X Window.

Использование inetd

В обычных условиях программа-сервер связывается с некоторым портом (ресурсом, для идентификации которого используется тип протокола и число в интервале от 1 до

65535). В зависимости от номера порта, указанного в запросе, этот запрос направляется тому или иному серверу. Например, почтовый сервер, поддерживающий SMTP (Simple Mail Transfer Protocol — простой протокол передачи почты), традиционно использует TCP-порт 25, а HTTP (Hypertext Transfer Protocol — протокол передачи гипертекстовой информации), как правило, связывается с портом 80.

Программа `inetd` является одним из суперсерверов, используемых в операционной системе Linux. Суперсервер выполняет функции посредника. Вместо набора серверов в системе запускается один суперсервер, который связывается со всеми портами, соответствующими серверам из набора. При установлении соединения суперсервер загружает сервер, порт которого указан в запросе, после чего этот сервер выполняет требуемые действия по передаче данных. Использование суперсервера обеспечивает два основных преимущества по сравнению с постоянным выполнением обычных серверов. Во-первых, при таком подходе уменьшается объем используемой оперативной памяти; в особенности это заметно, если на компьютере должно присутствовать большое количество программ-серверов. Во-вторых, перед тем как запрос будет передан обычном серверу, его получает суперсервер, который может выполнять необходимую фильтрацию данных; это повышает безопасность системы. Недостаток использования суперсервера состоит в том, что продолжительность обработки запроса увеличивается (как правило, на одну-две секунды). Это связано с тем, что для загрузки сервера требуется определенное время. Поэтому суперсервер лучше применять для серверов, которые вызываются достаточно редко. Если запросы к серверу приходят часто, лучше, если такой сервер постоянно присутствует в памяти компьютера.

Формат файла /etc/inetd.conf

Для настройки `inetd` используется конфигурационный файл `/etc/inetd.conf`. Если не принимать во внимание комментарии (строки, начинающиеся с символа #), то можно сказать, что содержимое файла `inetd.conf` представляет собой набор строк, каждая из которых определяет отдельный сервер. Пример записи, содержащейся в файле `/etc/inetd.conf`, приведен ниже.

```
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
```

Каждая строка файла состоит из нескольких полей, которые отделяются друг от друга с помощью пробелов или символов табуляции.

- Имя сервера. Первое поле в строке идентифицирует протокол, используемый сервером. Имя протокола должно соответствовать имени, указанному в файле /etc/services. Например, обратившись к этому файлу, можно выяснить, что имени `telnet` соответствует значение 23'/tcp, т. е. сервер, поддерживающий протокол `telnet`, должен использовать для взаимодействия порт 23. Для того чтобы программа `inetd` могла управлять сервером, для этого сервера должна существовать запись в файле /etc/services. Очевидно, что, планируя запуск редко встречающегося сервера посредством `inetd`, надо позаботиться о том, чтобы соответствующая запись была включена в этот файл. Подавляющее большинство серверов изначально учтено в /etc/services.

- Тип гнезда. Второе поле указывает тип гнезда, используемого при поддержке протокола. Допустимы типы `stream`, dgram, raw, `rdm` и `seqpacket`.

- **Тип протокола.** Третье поле указывает тип протокола. В данном случае речь идет о нижележащем протоколе транспортного уровня, например TCP или UDP. Допустимые протоколы указаны в /etc/protocols, однако в подавляющем большинстве случаев в этом поле указывается значение tcp или udp.

- wait/nowait. Четвертое поле записи содержит одно из двух значений: wait или nowait. Значение wait имеет смысл только для дейтаграмм (тип гнезда dgram). В остальных случаях предполагается значение nowait. Большинство серверов, поддерживающих обмен с помощью дейтаграмм, связываются с гнездом и освобождают inetd для обслуживания последующих обращений. Эти серверы называются *многопотоковыми* (multi-threaded); для них в рассматриваемом здесь поле должно содержаться значение nowait. Серверы, которые связываются с гнездом, обрабатывают все данные, а затем по истечении времени тайм-аута завершают работу, называются *однопотоковыми* (single-threaded); для них в данном поле должно содержаться значение wait. В этом поле можно также задать числовое значение, отделив его от ключевого слова wait точкой, например wait.60. Число указывает максимальное количество серверов данного типа, которые inetd может загрузить в течение одной минуты. По умолчанию принимается значение, равное 40.

- **Имя пользователя.** Программа inetd может запустить сервер с привилегиями указанного пользователя. Это позволяет существенно повысить уровень безопасности системы. Ограничив права сервера необходимым минимумом, вы сокращаете возможности злоумышленников по незаконному проникновению в систему. Так, например, серверу Apache не требуются никакие специальные привилегии, поэтому его можно запускать с правами пользователя nobody либо определить права Apache, создав для него отдельную учетную запись. В приведенном выше примере указано имя root, так как привилегии этого пользователя необходимы для выполнения процедуры регистрации, которая осуществляется в начале Telnet-сеанса. Если к имени пользователя вы добавите имя группы, разделив эти имена точкой, сервер получит привилегии группы. Например, значение nobody.nogroup указывает на то, что сервер должен быть запущен с правами пользователя nobody и группы nogroup.

- **Программа-сервер.** Шестое поле содержит имя программы-сервера, которую должен запустить inetd, приняв запрос. В приведенном примере указано имя программы /usr/sbin/tcpd. В действительности tcpd — это не сервер, а программа, реализующая TCP Wrappers (назначение TCP Wrappers будет рассмотрено ниже). В большинстве дистрибутивных пакетов, в которых используется inetd, также применяется TCP Wrappers, т. е. серверы, поддерживаемые inetd, запускаются через tcpd. Для некоторых серверов TCP Wrappers можно не использовать, но в большинстве случаев применение данного средства оправдано.

- **Параметры, передаваемые серверу.** Это поле может отсутствовать. Если же оно задано, то содержит параметры, которые должны быть переданы программе-серверу. Эти параметры могут изменять поведение сервера, указывать расположение конфигурационных файлов и предоставлять другие сведения. Если сервер запускается посредством TCP Wrappers, параметр задает имя этого сервера; в приведенном выше примере указан параметр in.telnetd. (При необходимости к имени сервера можно добавить параметры, предназначенные для него.)

Для редактирования `/etc/inetd.conf` можно использовать любой текстовый редактор. При подготовке содержимого файла следите за тем, чтобы поля записи помещались в одной строке. (Если в составе записи присутствуют длинные имена файлов или если серверу передается большое число параметров, редактор может автоматически перенести часть записи на другую строку. В этом случае при работе `inetd` возникнут проблемы.) Если вы хотите добавить новую запись для установленного вами сервера, изучите документацию на этот сервер и выясните, как должна выглядеть соответствующая запись `inetd.conf`. Часто бывает удобно создавать новые записи на основе существующих. При этом надо внимательно следить за тем, чтобы тип гнезда, тип протокола, параметры, передаваемые серверу, и другие сведения были заданы правильно, в противном случае сервер не будет работать.

Большинство дистрибутивных пакетов, в которых используется `inetd`, содержит файл `/etc/inetd.conf`, настроенный для поддержки наиболее распространенных серверов. Многие записи закомментированы. Для того чтобы сервер был активным, достаточно убрать символ комментариев из строки (очевидно, что это можно сделать только в том случае, если сервер установлен в системе). Для некоторых служб в составе `inetd.conf` присутствует несколько записей, например, в этот файл может быть включено несколько строк, описывающих различные FTP-серверы (ProFTPd и WU-FTPD). Вам, как администратору системы, необходимо проследить за тем, чтобы все такие записи, кроме одной, были закомментированы.

СОВЕТ

Инсталлировав систему Linux, вам надо внимательно просмотреть содержимое файла `/etc/inetd.conf` (или конфигурационного файла `xinetd`, который будет рассматриваться ниже) и закомментировать записи для тех серверов, которые не нужны в системе. Многие администраторы включают символы комментариев в начало тех записей, назначение которых им не понятно. Такие действия вполне допустимы, потому что ни один сервер, указанный в `inetd.conf`, не является необходимым компонентом системы; лишь некоторые из них могут потребоваться для регистрации пользователей по сети. Отключение лишних серверов повышает безопасность системы, так как сужает поле деятельности злоумышленников, пытающихся получить доступ в вашу систему из Internet. Использовать подобный подход для служб, загружаемых с помощью сценариев SysV, нельзя, поскольку многие из них жизненно важны для нормальной работы системы Linux.

Использование TCP Wrappers

Как было сказано ранее, инструмент TCP Wrappers играет роль посредника между `inetd` и целевым сервером. Средства TCP Wrappers применяются для повышения безопасности системы; они позволяют задавать правила установления соединений, защищая тем самым сервер от нежелательного взаимодействия. Предположим, что вы хотите, чтобы доступ к серверу Telnet имели только пользователи, работающие в вашей локальной сети. Программу, обеспечивающую работу сервера Telnet, можно настроить так, чтобы она отвергала попытки обращения с узлов, для обслуживания которых сервер не предназначен. Однако не все серверы предоставляют такие возможности. Передача TCP Wrappers полномочий по управлению соединением повышает гибкость системы, не требуя при этом внесения изменений в программы.

Для управления работой TCP Wrappers используются два файла: `/etc/hosts.allow` и `/etc/hosts.deny`. Эти файлы имеют одинаковый формат, но выполняют противоположные действия. В файле `hosts.allow` описываются узлы сети, которым разрешено обращаться к данному компьютеру; для всех остальных узлов доступ запрещен. Файл `hosts.deny`, напротив, содержит описания узлов, доступ с которых запрещен; все остальные узлы могут устанавливать соединение с данным компьютером. Если в системе присутствуют оба файла, приоритет имеет файл `hosts . allow`. Благодаря этому вы имеете возможность задать ограничения в файле `hosts.deny`, а затем разрешить доступ для отдельных компьютеров. Если сведения о сервере не включены ни в один из файлов (сервер может быть описан либо непосредственно, либо с помощью групповой операции), TCP Wrappers разрешает доступ к нему для всех узлов сети.

TCP Wrappers можно сравнить с локальным брандмауэром, работа которого будет рассматриваться в главе 25. При этом TCP Wrappers реализует дополнительную защиту, которая может оказаться полезной, если брандмауэр настроен неправильно, кроме того, этот инструмент предоставляет новые возможности, например, фильтрацию на основе имени группы NIS.

Подобно другим конфигурационным файлам, символ # в начале строки означает, что в данной строке содержатся комментарии. Запись в файле `hosts.allow` или `hosts . deny` имеет следующий формат:

`список_демонов : список_клиентов`

В списке демонов указывается один или несколько серверов, к которым применяется данное правило. Если в списке указано несколько серверов, их имена разделяются запятыми или пробелами. Имена серверов должны совпадать с именами, содержащимися в файле `/etc/services`. Кроме имен серверов в этом поле можно также указывать ключевое слово ALL, определяющее групповую операцию. Оно означает, что правило применяется ко всем серверам, управляемым TCP Wrappers.

ВНИМАНИЕ Не забывайте, что не все серверы запускаются с помощью TCP Wrappers. Поэтому групповая операция ALL может не включать все серверы, выполняющиеся в системе. Аналогично, указав сервер в списке демонов, вы не защитите его, если для управления им не применяются `inetd` и TCP Wrappers, либо если он не использует TCP Wrappers непосредственно.

Список клиентов определяет компьютеры, которым разрешен или запрещен доступ к серверу. Подобно списку доменов, в списке серверов может быть указан один узел либо несколько узлов. Идентификаторы узлов разделяются запятыми или пробелами. Описания узлов сети могут быть представлены в перечисленных ниже форматах.

- **IP-адрес.** В списке клиентов можно указать конкретный IP-адрес, например 10.102.201.23. Такое описание определяет только этот адрес.

- **Диапазон IP-адресов.** Задать диапазон IP-адресов можно несколькими способами. Проще всего сделать это, указав в составе адреса меньше четырех десятичных чисел; в этом случае адрес должен заканчиваться точкой. Например, значение 10.102.201. соответствует сети 10.102.201.0/24. Кроме того, можно использовать запись типа IP-адрес/маска. В файлах `hosts.allow` и `hosts.deny` также поддерживаются адреса IPv6. Они задаются в виде `[n:n:n:n:n:n:n:n]`*/длина,* где

л — значения компонентов адреса, а длина — это число битов, используемых для представления диапазона.

- Имя узла. Узел можно описывать с помощью его доменного имени, например `badcracker.threeroomco.com`. Этим способом определяется только один узел. В этом случае при получении запроса система выполняет преобразование имен, а, следовательно, если сервер DNS работает некорректно, при идентификации компьютера могут быть допущены ошибки.

- Домен. Домен можно задавать так же, как вы задается доменное имя одного компьютера. Отличие состоит лишь в том, что в данном случае имя должно начинаться с точки. Если в файле указано имя `.threeroomco.com`, оно определяет все компьютеры, принадлежащие домену `threeroomco.com`.

- Имя группы NIS. Если последовательность символов начинается со знака @, оно интерпретируется как имя группы NIS (Network Information Services — сетевая информационная служба). Этот метод предполагает, что в сети функционирует сервер NIS.

В списке клиентов могут присутствовать ключевые слова, определяющие групповые операции. Назначение этих ключевых слов описано ниже.

- ALL. Идентифицирует все компьютеры.

- LOCAL. Определяет все локальные компьютеры на основании анализа имени узла. Если в имени отсутствует точка, соответствующий узел считается локальным.

- UNKNOWN. Данное ключевое слово задает все компьютеры, чьи доменные имена не могут быть получены средствами преобразования имен.

- KNOWN. Идентифицирует компьютеры, доменные имена и IP-адреса которых известны системе.

- `PARANOID`. Определяет компьютеры, имена которых не соответствуют IP-адресам.

При использовании последних трех ключевых слов надо соблюдать осторожность, поскольку, если они присутствуют в списке клиентов, компьютер обращается к серверу DNS. Неисправность сетевого оборудования может привести к ненадежной работе сервера DNS. Если сервер DNS недоступен, получить доменное имя компьютера не удастся. Пример файла `/etc/hosts.allow`, содержащего две строки, приведен ниже.

```
telnet,ftp : 192.168.34. dino.pangaea.edu
ssh : LOCAL .pangaea.edu
```

Первая строка задает правила установления соединений для серверов Telnet и FTP, разрешая доступ к ним только из сети 192.168.34.0/24 и с компьютера `dino.pangaea.edu`. Вторая строка сообщает о том, что доступ к серверу SSH разрешен только для машин локальной сети, а также для компьютеров, принадлежащих домену `pangaea.edu`. Поскольку другие серверы не указаны в списках демонов, TCP Wrappers не блокирует доступ к ним. Например, если вы запустите через `inetd` и TCP Wrappers Apache, обратиться к этому серверу сможет каждый желающий.

Используя в списке клиентов записи типа *пользователь@компьютер*, вы можете управлять доступом отдельных пользователей, работающих на удаленных узлах. Для того чтобы это было возможно, на клиентском компьютере должен выполняться сервер ident (в некоторых системах он называется auth), который возвращает имя пользователя, работающего с конкретным сетевым портом. Компьютер, использующий TCP Wrappers, передает запрос клиентской машине и получает имя пользователя. В этом случае соединение устанавливается с некоторой задержкой, а информация о пользователе, полученная из Internet, не всегда заслуживает доверия. Поэтому данную возможность лучше использовать в локальной сети, где вы имеете возможность контролировать конфигурацию всех компьютеров.

В составе правила может присутствовать дополнительное ключевое слово EXCEPT. Оно определяет исключения из этого правила. Рассмотрим следующую запись, содержащуюся в файле /etc/hosts.deny:

```
www : badcracker.org EXCEPT goodguy@exception.badcracker.org
```

В данном случае доступ к Web-серверу запрещается для всех компьютеров, принадлежащих домену badcracker.org. Исключением являются лишь запросы, полученные от пользователя goodguy@exception.badcracker.org. Аналогичный результат можно получить, включив правило для goodguy@exception.badcracker.org в файл /etc/hosts.allow.

Если перед вами стоит задача максимально повысить безопасность системы, вы можете начать настройку с создания файла /etc/hosts.deny, содержащего следующую информацию:

```
ALL : ALL
```

Эта запись блокирует доступ ко всем серверам, поддерживаемым TCP Wrappers, с любого компьютера, независимо от его адреса. Затем можно постепенно разрешать доступ к серверам, составляя соответствующие правила и записывая их в файл /etc/hosts.allow. Возможности доступа должны ограничиваться необходимым минимумом. В частности, к серверам, чувствительным к попыткам взлома извне, например к Telnet, следует разрешить доступ только для определенных компьютеров. (Дело в том, что в процессе Telnet-взаимодействия данные, в том числе пароль, передаются в незашифрованном виде. Строго говоря, если компьютер содержит важные данные, на нем не следует вовсе устанавливать Telnet-сервер. Подробно эти вопросы будут обсуждаться в главе 13.)

Использование xinetd

Традиционно inetd был основным суперсервером, использовавшимся в системе Linux. Однако в 2000 г. наметилась тенденция перехода к альтернативному суперсерверу xinetd. Условно xinetd можно представить себе как сочетание inetd и TCP Wrappers. Но между этими программами существуют некоторые отличия. Не все возможности xinetd можно реализовать с помощью inetd и TCP Wrappers, но ряд действий, которые можно выполнить, используя inetd и TCP Wrappers, нельзя сделать посредством xinetd. При необходимости xinetd можно использовать совместно с TCP Wrappers, поэтому считается, что данный инструмент обеспечивает большую степень гибкости по сравнению с inetd. В начале 2002 г. xinetd был использован в Red Hat и Mandrake

в качестве суперсервера по умолчанию; ожидается также переход других операционных систем на xinetd.

Формат файла /etc/xinetd.conf

Поскольку возможности нового суперсервера расширены по сравнению с inetd, формат конфигурационного файла также отличается от inetd. Настройка xinetd производится с помощью файла /etc/xinetd.conf. Следует заметить, что файл xinetd.conf, поставляемый в составе дистрибутивных пакетов Red Hat и Mandrake, содержит лишь минимальный набор установок. В нем задана конфигурация серверов, а также содержится строка, которая указывает суперсерверу прочитать все файлы в каталоге /etc/xinetd.d и интерпретировать их как дополнительные конфигурационные файлы. Конфигурация xinetd напоминает конфигурацию SysV; каждому серверу соответствует собственный управляющий файл, названный по имени сервера. Например, для сервера Telnet используется файл /etc/xinetd.d/telnet. При необходимости xinetd можно настроить так, что этот суперсервер будет использовать лишь основной файл xinetd.conf, но в дистрибутивных пакетах Red Hat и Mandrake некоторые файлы запуска уже содержатся в каталоге /etc/xinetd.d.

Независимо от того, содержится ли описание сервера в /etc/xinetd.conf или в файле, находящемся в каталоге /etc/xinetd.d, оно может занимать несколько строк. Базовое определение включает те же данные, что и запись в файле inetd.conf. Например, приведенное ниже описание почти эквивалентно рассмотренной ранее записи для Telnet-сервера, находящейся в файле inetd.conf.

```
service telnet
{
        socket_type = stream
        protocol    = tcp
        wait        = no
        user        = root
        server      = /usr/sbin/in.telnetd
}
```

В конфигурационном файле xinetd каждое поле именуется. Несмотря на то что в данном примере поля расположены в том же порядке, что и в рассмотренной ранее записи inetd, порядок их следования может быть произвольным. Как нетрудно заметить, в данном примере не вызывается TCP Wrappers, однако при необходимости этот инструмент можно использовать (для того, чтобы Telnet-сервер запускался через TCP Wrappers, надо задать значение /usr/bin/tcpd поля server и добавить поле server_args, присвоив ему значение /usr/sbin/in.telnetd).

В дополнение к стандартным средствам inetd xinetd предоставляет новые опции, расширяющие возможности суперсервера. Большинство из этих опций включаются в описание сервера и помещаются в фигурные скобки. Наиболее важные опции описаны ниже.

- Средства защиты. Как упоминалось ранее, xinetd поддерживает большое количество опций, предназначенных для повышения безопасности системы. Средства, соответствующие многим из этих опций, эквивалентны средствам, предоставляе-

мым TCP Wrappers. Опции защиты подробно будут рассматриваться в следующем разделе.

- **Запрет вызова сервера.** Для того чтобы запретить вызов сервера, управляемого суперсервером inetd, надо закомментировать соответствующую строку в конфигурационном файле. В программе xinetd для этой цели используется опция disable = yes, которая помещается в описание требуемого сервера. Тот же результат можно получить, включив в раздел defaults основного файла /etc/xinetd.conf опцию disables = список_серверов, где список серверов состоит из имен серверов, разделенных пробелами. Различные инструментальные средства настройки используют оба способа. Если в описании сервера присутствует опция disable = no, это значит, что сервер активен.

- **Перенаправление.** Если вам необходимо передать запрос на другой компьютер, вы можете сделать это с помощью опции redirect = *целевой_компьютер*, где целевой компьютер (т. е, компьютер, которому должен быть передан запрос) задается с помощью доменного имени или IP-адреса. Например, если вы включите в описание сервера, содержащееся в файле /etc/xinetd.d/telnet на узле dummy.threeroomco.com, опцию redirect = 192.168.3.78, то при попытке обращения к Telnet-серверу на компьютере dummy.threeroomco.com запрос будет перенаправлен на 192.168.3.78. Эту возможность использует NAT-маршрутизатор для того, чтобы организовать обслуживание внешних запросов компьютером, принадлежащим внутренней сети. Тот же результат достигается с помощью iptables, но применяя для этой цели xinetd, вы можете использовать средства управления доступом суперсервера.

- **Протоколирование.** Используя опции log_on_success и log_on_failure суперсервера xinetd, вы можете определять, какая информация должна записываться в файл протокола в случае успешного или неудачного обращения к серверу. Значениями этих опций могут быть PID (идентификатор процесса сервера), HOST (адрес клиента), USERID (идентификатор пользователя клиентской системы, которая передала запрос), EXIT (время получения запроса и статус завершения его обработки) и DURATION (длительность сеанса). При необходимости вы можете добавлять к набору, принятому по умолчанию, или исключать из него отдельные значения, используя вместо символа = пары символов += и -=.

- **Ограничения на установление соединений.** Ограничить число соединений, поддерживаемых xinetd, можно несколькими способами. Опция per_source определяет, сколько запросов от одного источника xinetd может обработать в единицу времени. (Значение UNLIMITED этой опции позволяет обрабатывать неограниченное число запросов.) Опция instances задает максимальное количество процессов, которые xinetd может породить (это значение должно быть больше, чем значение опции per_source). При использовании опции cps ей передаются два значения, разделенные пробелом: число соединений, которые xinetd может установить в течение одной секунды, и длительность паузы (в секундах), которая должна быть выдержана, если число соединений превысит максимально допустимое. Приоритет серверов, управляемых xinetd, задается с помощью опции nice; эта опция действует подобно программе nice. И наконец, опция max_load, значением

которой является число с плавающей точкой, указывает максимальную загрузку системы, при достижении которой xinetd должен отвергать последующие запросы. При использовании этих опций снижается вероятность того, что сервер пострадает от атаки, предпринятой с целью вывода его из строя, или в результате обилия запросов, вызванных высокой популярностью сервера.

Большинство из приведенных выше опций можно использовать либо в описании сервера, либо в разделе defaults файла /etc/xinetd.conf. Помещенная в раздел defaults опция воздействует на все серверы, управляемые xinetd. Если опция присутствует и в разделе defaults, и в описании, принимается значение опции, заданное в описании сервера.

Если вы внесли изменения в файл /etc/xinetd.conf или в один из файлов, содержащихся в каталоге /etc/xinetd.d, необходимо перезапустить программу xinetd. Поскольку суперсервер xinetd чаще всего запускается посредством сценария SysV, проще всего перезапустить его с помощью команды типа /etc/rc.d/init.d/xinetd restart (в некоторых системах сценарий запуска может находиться в другом каталоге). Можно поступить и по-другому — передать xinetd сигнал SIGUSR1 или SIGUSR2, используя для этого команду kill. При получении сигнала SIGUSR1 xinetd читает содержимое нового конфигурационного файла и продолжает работу. В ответ на сигнал SIGUSR2 суперсервер делает то же самое, но при этом завершает работу тех серверов, которые согласно новому конфигурирационному файлу должны быть неактивны.

Средства управления доступом

Одно из преимуществ xinetd состоит в том, что эта программа объединяет в себе функции суперсервера и средства управления доступом, характерные для TCP Wrappers. Кроме того, настройка xinetd выполняется достаточно просто. Средства управления доступом xinetd не дублируют соответствующие функции TCP Wrappers; некоторые задачи лучше решаются с помощью xinetd, для решения других приходится применять TCP Wrappers. Настраивая xinetd, можно определять доступ либо одновременно для всех серверов, либо для каждого сервера в отдельности. Основные опции, предназначенные для управления доступом, описаны ниже.

- Ограничения для различных узлов. Для xinetd предусмотрены опции only_from и no-access, которые выполняют те же функции, что и содержимое файлов /etc/hosts.allow и /etc/hosts.deny TCP Wrappers. Эти опции могут присутствовать либо в главном конфигурационном файле, либо в файле, предназначенном для конкретного сервера. В качестве значения опции only_from задается список компьютеров, которым разрешено обращаться к серверу (для всех остальных компьютеров доступ запрещен). Аналогично, значение опции no-access представляет собой "черный список"; компьютеры, указанные в списке, не имеют права устанавливать соединение с сервером, а для остальных компьютеров доступ разрешен. Если адрес присутствует в обоих списках, приоритет имеет адрес, заданный более конкретно. Для идентификации компьютеров используются разные способы. В опциях only_from и no-access может быть указан IP-адрес узла (например, 172.23.45.67), адрес сети, оканчивающийся нулем (например, 172.23.0.0 для сети 172.23.0.0/16) или заданный с помощью маски (172.23.0.0/16), имя сети, указанное в файле /etc/networks, или доменное имя узла (например,

badguy.threeroomco.com). Если в качестве значения опции указано имя узла, xinetd выполняет преобразование имени в адрес один раз при загрузке суперсервера. Поскольку в течение работы xinetd доменное имя может измениться, данный способ установления ограничений неэффективен.

- **Ограничения по времени.** Для указания временного интервала, в течение которого сервер доступен для клиентов, используется опция access_times. Значение этой опции задается в формате *часы:минуты-часы:минуты*, например, 08:00-18:00 означает, что к серверу можно обращаться с 8 до 18 часов. Значение опции access_times влияет только на установление соединения. Например, если для Telnet-сервера задан интервал 08:00-18:00, то соединение, установленное в 17:58, может использоваться как угодно долго.

- **Ограничения на использование интерфейсов.** При необходимости вы можете связать сервер с одним сетевым интерфейсом. Для этого используется опция bind (либо опция interface, которая является синонимом bind). В качестве значения опции задается IP-адрес, связанный с интерфейсом. Например, если интерфейсу eth1 присвоен адрес 172.19.28.37 и для сервера задана опция bind = 172.19.28.37, это означает, что обращаться к этому серверу можно только через интерфейс eth1. Попытки установить соединение через ethO окончатся неудачей; результат будет такой же, как в случае, когда сервер не установлен в системе. Эту опцию удобно использовать на маршрутизаторах или на компьютерах, подключенных одновременно к нескольким сетям. Предположим, что на компьютере, обеспечивающем связь локальной сети с Internet и использующем для соединения с сервером провайдера PPP-соединение, установлены серверы Telnet и FTP. С помощью опции bind вы можете настроить xinetd так, чтобы доступ к серверам Telnet и FTP имели только компьютеры, подключенные к локальной сети.

В этом и предыдущем разделе были описаны лишь наиболее часто используемые опции xinetd. Для получения информации об остальных опциях обратитесь к справочным данным.

Использование локальных сценариев запуска

Как правило, в системе Linux большинство стандартных серверов запускается либо с помощью сценариев SysV, либо суперсервера. Исключением является сервер X, для запуска которого в файле /etc/inittab предусмотрена соответствующая запись. Сервер X запускается только на конкретном уровне выполнения. В системе Slackware для запуска основных серверов используется /etc/rc.d/rc/inet2. Большинство дистрибутивных пакетов включает локальные сценарии запуска, редактируя которые администратор имеет возможность обеспечить работу дополнительного сервера, запустить специальную утилиту или выполнить другие подобные действия. Локальные сценарии запуска для основных дистрибутивных пакетов Linux приведены в табл. 4.1.

Как правило, локальные сценарии запуска применяются для того, чтобы включить в систему сервер, для которого отсутствуют соответствующие сценарии SysV и который по каким-либо причинам нежелательно запускать посредством суперсервера. Сценарии запуска SysV ориентированы на конкретную версию системы, поэтому для серверов, которые не включены в дистрибутивный пакет, как правило, отсутствуют и сценарии

запуска. Например, если вы установите в Mandrake сервер, ориентированный на SuSE, и попытаетесь использовать для его запуска сценарии SysV, которые также предназначены для системы SuSE, сервер не будет работать. Аналогичные проблемы возникают, если автор предоставляет вам исходные коды сервера. Как правило, такие коды не настроены на конкретный дистрибутивный пакет Linux. Сервер может компилироваться без ошибок и надежно работать, но для того, чтобы обеспечить его запуск, вам придется приложить определенные усилия.

При необходимости вы можете сами написать сценарий SysV для запуска сервера. Сделать это можно, модифицируя имеющийся сценарий. Еще проще создать сценарий SysV, имея в распоряжении сценарий, выполняющий аналогичную задачу (такая ситуация может возникнуть, если вы переходите от устаревшей версии сервера к новой или если вы заменяете один из серверов, присутствующих в системе, подобным сервером независимого производителя). Однако при создании сценария SysV все же могут возникнуть проблемы. Формат сценариев запуска SysV, используемый в вашей системе, может быть незнаком для вас (обычно для написания сценариев SysV используется язык оболочки bash). Если вы не хотите тратить время на изучение формата сценариев или языка программирования, вы можете воспользоваться локальными сценариями запуска.

Для внесения изменений в локальный сценарий запуска можно использовать обычный текстовый редактор. В файл надо включить строку, формат которой совпадает с форматом команды, используемой для запуска этого сервера. Например, приведенная ниже строка позволяет запускать Telnet-сервер.

```
/usr/sbin/in.telnetd
```

По умолчанию сервер запускается в режиме, отличном от *режима демона* (т. е. он не выполняется в фоновом режиме и может управляться оболочкой). Для того чтобы указать, что сервер должен работать как фоновая программа, надо в конце командной строки указать символ &. Если вы не сделаете этого, то после вызова данного сервера сценарий запуска прекратит работу. Такую ситуацию можно условно считать допустимой *в* том случае, если строка, запускающая данный сервер, является последней строкой сценария, но в любом случае символ & нельзя забывать.

Сценарий запуска позволяет выполнять более сложные действия, чем запуск сервера посредством одной строки. Например, с помощью условных выражений языка bash можно проверить, существует ли файл сервера, или запустить сервер только при выполнении определенных условий. Подобные действия часто выполняют сценарии запуска SysV.

Необходимо помнить, что локальные сценарии запуска в разных системах могут отличаться друг от друга. Например, в SuSE сценарий boot.local запускается раньше, чем сценарий rc.local системы Red Hat, поэтому локальный сценарий запуска SuSE лучше подходит для выполнения различных действий с интерфейсами и для других задач, которые должны решаться на ранних этапах загрузки системы, а сценарий Red Hat больше пригоден для запуска сервера и других действий, которые должны выполняться тогда, когда основные средства поддержки сетевых функций уже установлены. Если задача, которую вам необходимо решить, редактируя сценарий запуска, существенно зависит от наличия в системе некоторых серверов, вам придется написать сценарий запуска SysV с номером, отражающим порядок его выполнения относительно других сценариев.

В основном локальные сценарии запуска используются для обеспечения загрузки серверов и других программ. Если работа сервера, запущенного посредством сценария SysV, может быть остановлена путем передачи сценарию параметра stop, то для сервера, кото-

рый был запущен при выполнении локального сценария, подобного способа завершения не существует. Если необходимо прекратить выполнение сервера, вам надо использовать утилиту `kill`, `killall` или другие подобные инструменты.

Использование инструментов с графическим интерфейсом

В составе многих дистрибутивных пакетов Linux поставляются инструментальные средства с графическим пользовательским интерфейсом, которые позволяют настраивать основные сетевые средства, организовывать запуск серверов и выполнять другие задачи, связанные с администрированием системы. В разных версиях Linux используются различные инструменты, но все они обладают некоторыми общими чертами. Эти программы обычно доступны с рабочего стола KDE (K Desktop Environment — среда рабочего стола K) или GNOME (GNU Network Object Model Environment — среда сетевой объектной модели GNU). Для их запуска можно также ввести команду в окне xterm. (Запускать инструменты администрирования может только пользователь `root`; в последние годы это правило строго соблюдается.) В данном разделе обсуждаются инструменты Linuxconf (используемый в Red Hat и системах, созданных на ее основе, например, Mandrake), YaST и YaST2 (применяемые в SuSE) и `ksysv` (вариант программы `ntsysv`, которая рассматривалась ранее в этой главе, снабженный графическим интерфейсом).

Существуют инструменты Webmin и SWAT, позволяющие выполнять задачи администрирования, взаимодействуя с системой средствами Web. Строго говоря, к таким инструментам можно отнести и Linuxconf, так как эта программа может размещаться не только локально, но и на удаленном компьютере; в этом случае для взаимодействия используется Web-интерфейс. Данные инструменты обсуждаются в главе 16.

Использование Linuxconf

Утилита Linuxconf представляет собой модульный инструмент конфигурирования системы. Она состоит из базовой структуры, в которую включаются модули, обеспечивающие поддержку конкретных серверов и позволяющие выполнять различные задачи по настройке системы. Linuxconf может работать в текстовом режиме (меню строятся из символов), в графическом режиме (в этом случае программа выполняется в отдельном окне), а также позволяет использовать в качестве интерфейса Web-броузер (этот режим будет рассматриваться в главе 16). Для поддержки графического интерфейса нужна не только основная программа `linuxconf`, но также дополнительный пакет (`gnome-linux-conf` или `linuxconf-gui`). Если программа Linuxconf может поддерживать графический интерфейс, она отображает его, в противном случае Linuxconf начинает работу в текстовом режиме. В данном разделе рассматривается работа утилиты в графическом режиме на локальном компьютере, но работа в текстовом режиме и через Web отличается лишь в деталях. Структура интерфейса в разных системах может быть различной. Например, в Red Hat программа отображает одно окно и выводит в нем сведения обо всех модулях, в то время как в Mandrake для каждого модуля открывается отдельное окно.

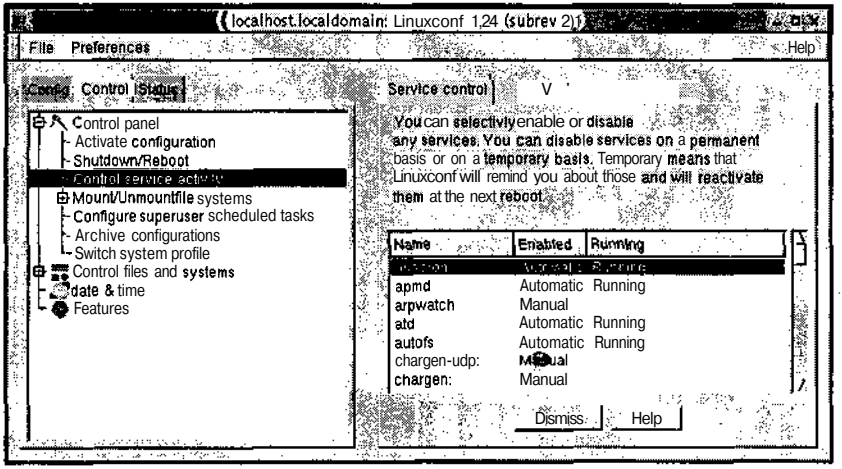


Рис. 4.2. Программу `linuxconf` можно использовать для управления работой системы Linux

Подробная информация о Linuxconf представлена на официальном Web-узле Linuxconf по адресу `http://www.solucorp.qc.ca/linuxconf/`. В настоящее время данная программа поставляется с Red Hat 7.2 и Mandrake 8.1, но не рекомендована к применению в обеих системах. Поэтому можно ожидать, что в ближайшее время вместо нее в состав дистрибутивных пакетов будет включен другой инструмент. На момент написания книги еще неизвестно, какие средства придут на замену Linuxconf: будет ли это один универсальный инструмент или набор средств, ориентированных на конкретные серверы. Несмотря на то что пакет Linuxconf предназначен для выполнения в системах Red Hat и Mandrake, существуют также версии для других систем. Информацию о них можно найти на Web-узле Linuxconf.

После запуска Linuxconf отображает информацию об областях конфигурации; данные распределены в трех вкладках: Config, Control и Status. Каждая область может включать подобласти; переходя от одной подобласти к другой, можно получить доступ к конкретному конфигурационному модулю. (В реализации Linuxconf для Mandrake после щелчка на области отображается отдельное окно, содержащее опции, доступные для этой области. Открывая таким образом новые окна, вы получаете доступ к конфигурационному модулю.) На рис. 4.2 показана реализация Linuxconf для Red Hat; информация в окне соответствует выбору модуля Control⇨Control Panel⇨Control Service Actinity. Этот модуль позволят управлять сценариями SysV и запуском сервера с помощью `xinetd`. Для того чтобы разрешить или запретить запуск сервера, выполните следующие действия.

1. Запустите Linuxconf и обратитесь к модулю Control⇨Control Panel⇨Control Service Actinity (см. рис. 4.2).

2. Выберите требуемый сервер в списке, отображаемом в правой части окна. Например, для управления сервером `sendmail` найдите пункт `sendmail` и щелкните на нем мышью. В результате в правой части окна Linuxconf отобразится новая вкладка, на которой будет отображаться текущее состояние сервера.

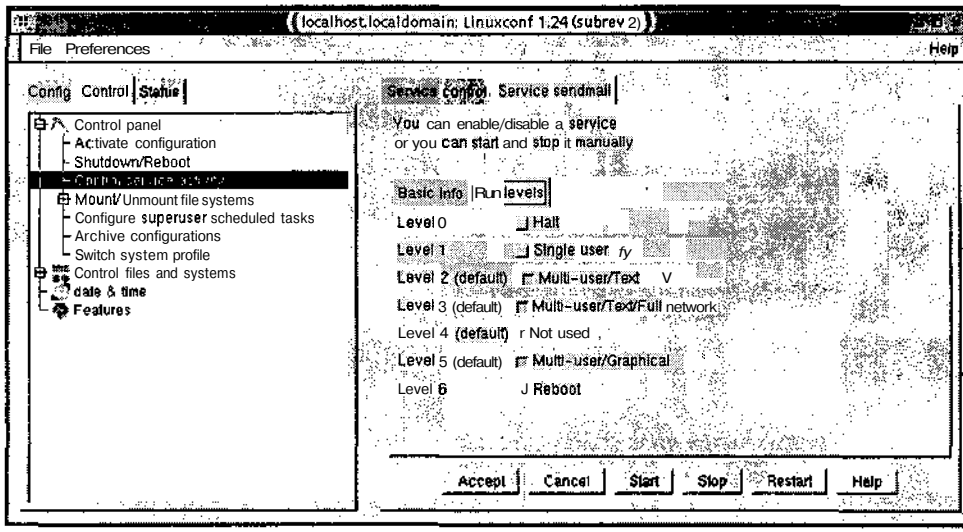


Рис. 4.3. С помощью `linuxconf` можно разрешать или запрещать запуск сервера на любом уровне выполнения

3. Выберите вкладку Run Levels. При этом окно программы должно выглядеть так, как показано на рис. 4.3. Вы можете разрешить или запретить запуск сервера на любом из уровней выполнения, для этого установите флажок опции рядом с требуемым уровнем.

4. Щелкните на кнопке Accept, а затем на кнопке Dismiss вкладки Service Control.

5. Выберите пункт меню File⇨Act/Changes, и программа выведет список возможных действий. Щелкните на пункте Do It, подтвердив тем самым внесенные изменения.

В результате этих действий система настраивается для загрузки серверов на указанном уровне выполнения. Чтобы проверить результаты, надо воспользоваться утилитой `chkconfig` или просмотреть имена файлов в каталоге ссылок SysV.

Помимо разрешения или запрета загрузки серверов, Linuxconf предоставляет возможность настройки некоторых из них. Поскольку данная программа не рекомендована к применению в Red Hat и Mandrake, многие модули настройки серверов не включаются в состав дистрибутивных пакетов. Эти модули можно найти на Web-узле Linuxconf, но они не всегда корректно работают. Причина в том, что расположение конфигурационных файлов серверов и даже их содержимое меняются в зависимости от версии системы, поэтому создать универсальный конфигурационный модуль невозможно.

Использование YaST и YaST2

В системе SuSE используются инструменты YaST (Yet Another Setup Tool) и YaST2. YaST работает в текстовом режиме, а YaST2 предоставляет графический интерфейс. Несмотря на различия в интерфейсе, обе программы обеспечивают одинаковые возможности. В этом разделе рассматривается YaST2, но если вам по каким-то причинам придется перейти на YaST, вы не будете испытывать затруднений при работе с этой программой.

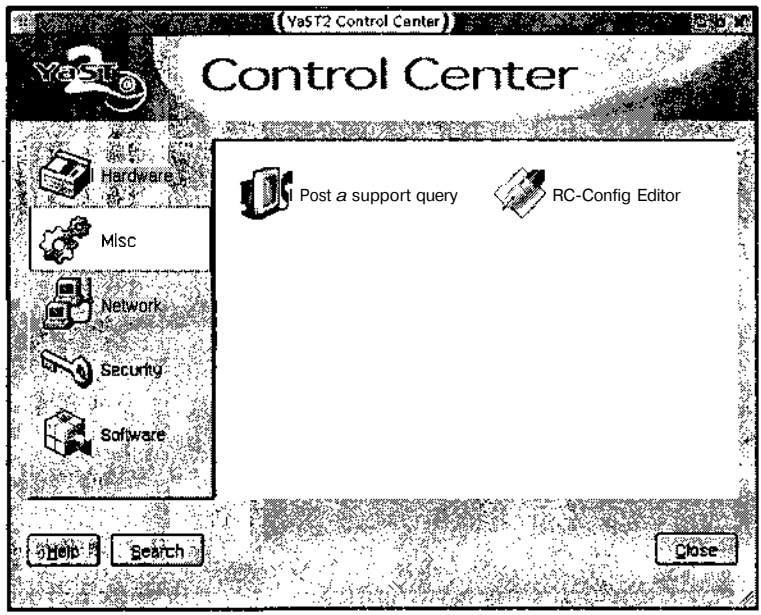


Рис. 4.4. YaST предоставляет специализированные инструменты настройки, которые объединяются в категории, называемые областями конфигурации

(Для простоты я употребляю термин YaST для обозначения обеих программ, за исключением тех случаев, когда обсуждаются различия между ними.) Для того чтобы вызвать YaST в текстовом режиме, надо ввести команду `yast`; аналогично, программа YaST2, предоставляющая графический интерфейс, вызывается по команде `yast2`. Внешний вид основного окна YaST2 показан на рис. 4.4. Категория, или область конфигурации, выбирается в левой части окна, а конкретные инструменты, предназначенные для настройки системы, отображаются справа.

В системе SuSE для настройки процедуры запуска наряду с содержимым каталога ссылок SysV используется файл `/etc/rc.config`. Для управления запуском серверов посредством YaST необходимо изменить содержимое конфигурационного файла. Для этого предусмотрен инструмент RC-Config Editor, содержащийся в области Misc. После выбора данного инструмента отображается окно, показанное на рис. 4.5. Большинство переменных, управляющих запуском сетевых серверов, расположено в области Start-Variables⇨Start-Network. Щелкните мышью на одном из пунктов списка, показанного на рис. 4.5, и YaST предоставит вам средства для установки значения переменной. Обычно программа предлагает на выбор значения Yes и No, соответствующие разрешению и запрету запуска сервера.

С помощью YaST можно задавать и другие характеристики системы. Многие важные переменные, определяющие параметры серверов, хранятся в файле `/etc/rc.config`; их значения можно задавать с помощью тех же инструментов, которые вы используете для настройки сценариев запуска SysV. Например, вы можете указать имя узла (Network⇨Network-Basics) или сообщить системе, допустима ли регистрация пользова-

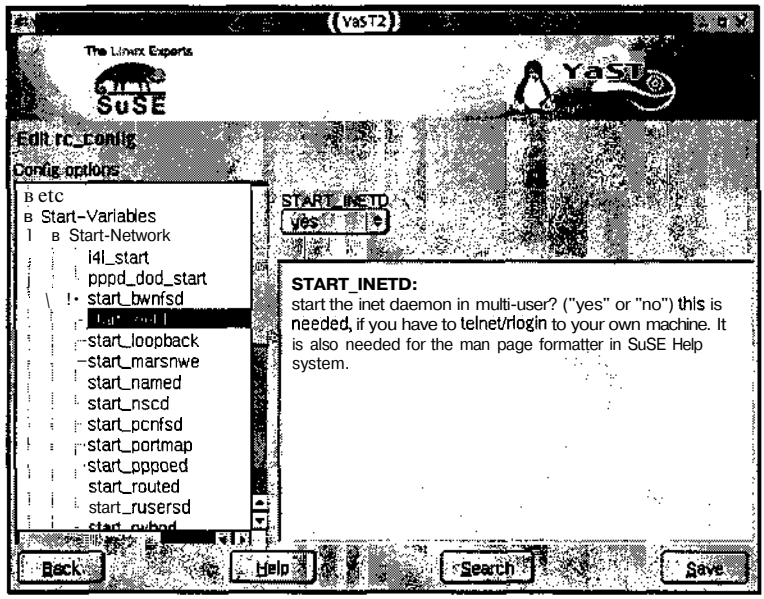


Рис. 4.5. Для разрешения или запрета запуска сервера переменной присваивается значение Yes или No

теля root с удаленного компьютера (Security⇨Security-Basics). Просмотрев содержимое различных областей, вы получите представление о возможностях YaST.

Специальные средства настройки серверов в основном находятся в области Network. Например, в этой области вы найдете инструменты NFS и Sendmail Configuration, которые используются соответственно для настройки NFS и sendmail. Вопросы настройки NFS и sendmail рассматриваются в главах 8 и 19, однако в них не уделяется внимание использованию YaST.

Использование ksysv

В одном из предыдущих разделов данной главы обсуждались инструментальные средства chkconfig и ntsysv, предназначенные для управления сценариями запуска SysV (а в некоторых случаях и серверами, загружаемыми с помощью суперсервера). Эти инструменты существенно упрощают процесс администрирования системы, но если администратор предпочитает программы с графическим интерфейсом, ему неудобно работать с ними. Существуют альтернативные программы администрирования, предоставляющие графический пользовательский интерфейс; в качестве примеров таких программ можно привести ksysv и tksysv. Программа ksysv создана в рамках проекта KDE, но может работать и в других средах. Инструмент tksysv не связан ни с какой конкретной графической средой. Обе эти программы нормально работают в Red Hat и в системах, созданных на ее основе. Окно программы ksysv показано на рис. 4.6.

Как ksysv, так и tksysv выводит слева в окне список сценариев запуска SysV; в остальной части окна отображаются списки серверов, запуск которых разрешен или запрещен на разных уровнях выполнения. После щелчка на пункте списка Available Ser-

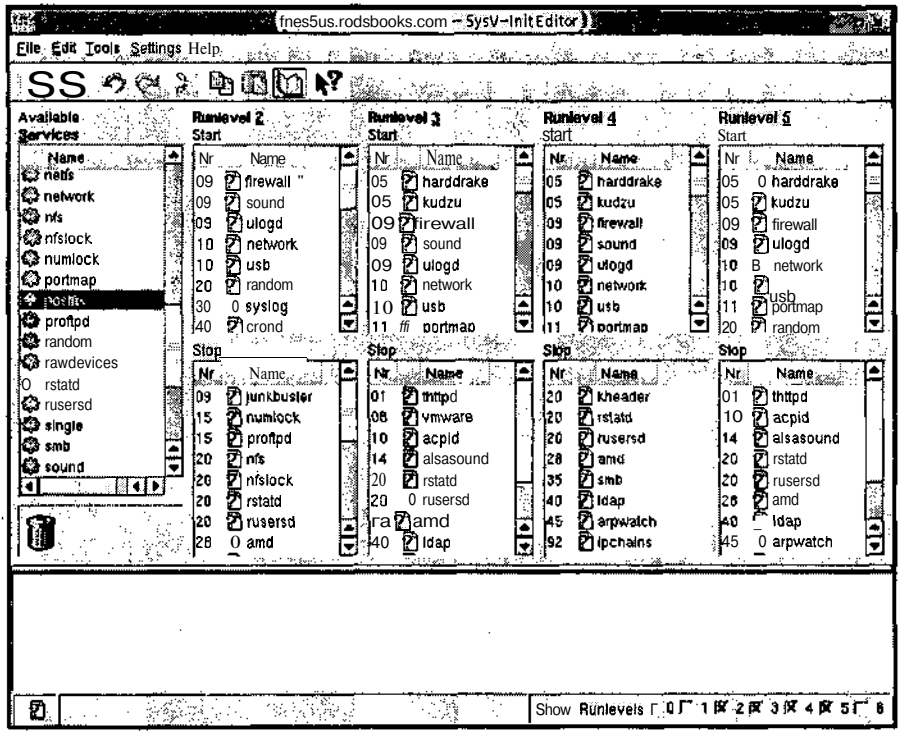


Рис. *4,6.* Для того чтобы задать особенности работы сервера, надо щелкнуть мышью на его имени

vices отображается диалоговое окно, содержащее описание сервера и управляющие элементы, посредством которого можно запустить, остановить или перезагрузить сервер, а также изменить имя файла и права доступа. Если вы щелкнете на пункте списка, соответствующего любому из уровней выполнения, будет выведено диалоговое окно, содержащее вкладки Service и Entry (рис. 4.7). На вкладке Service представлено описание сервера, а на вкладке Entry находятся инструменты, позволяющие изменить имя ссылки, указывающей на сценарий запуска и номер, определяющий порядок запуска.

Для того чтобы разрешить автоматический запуск сервера, надо перетащить мышью имя сервера из списка Stop, соответствующего определенному уровню выполнения, в список Start того же уровня. Чтобы запретить запуск сервера, надо выполнить обратное действие. Вы также можете перетащить имя сервера из списка Available Services в список, соответствующий уровню выполнения. В любом случае ksysv присвоит серверу последовательный номер, определяемый позицией, в которую вы перетащили соответствующий пункт списка. Например, если вы перетащили имя сервера и вставили его между пунктами списка с последовательными номерами 20 и 30, ksysv присвоит ему номер 25. Для того чтобы изменить этот номер, надо щелкнуть на имени сервера в списке, соответствующем уровню выполнения, и ввести новое значение в поле Sorting Number (см. рис. 4.7). Очевидно, что ksysv не имеет сведений о том, какой номер наиболее подходит для сервера в вашей системе, поэтому при настройке системы приходится уделять внимание выбору последовательных номеров. Используя ksysv, можно получить кон-

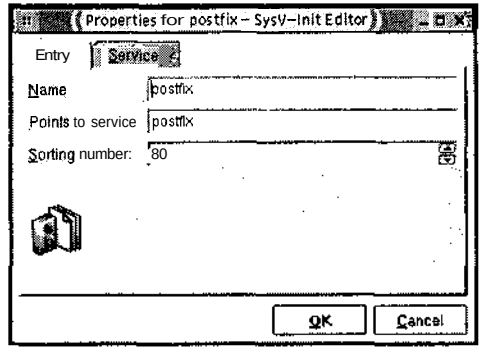


Рис. 4.7. После щелчка на имени службы ksysv отображает диалоговое окно, позволяющее редактировать сценарии запуска и ссылки SysV

фигурацию, при которой на каком-то из уровней имя сервера будет присутствовать как в списке Start, так и в списке Stop, поэтому необходимо следить за тем, чтобы подобная ситуация не возникала.

Утилиты `ksysv` и `tksysv` не обеспечивают той степени гибкости, которой позволяют добиться такие инструменты, как Linuxconf и YaST. Инструменты, предназначенные для управления сценариями SysV, в отличие от более универсальных средств, ориентированы на выполнение одной задачи. Несмотря на то что специализированные утилиты упрощают работу со сценариями SysV, использование их не освобождает вас от необходимости знать основные принципы работы системы. Вы должны представлять себе, что такое уровни выполнения, как определяется последовательность вызова сценариев, и другие особенности работы со средствами запуска SysV.

Выбор способа запуска сервера

Поскольку серверы, предназначенные для выполнения в системе, могут запускаться по-разному, возникает проблема выбора наиболее приемлемого метода запуска конкретного сервера. Большинство серверов, поставляемых в виде дистрибутивных пакетов, ориентированы на определенный метод запуска. Так, например, в состав дистрибутивного пакета может входить сценарий запуска SysV либо конфигурационный файл суперсервера, предназначенный для размещения в каталоге `/etc/xinetd.d`. В большинстве случаев изменять способ запуска, предусмотренный в дистрибутивном пакете, нет необходимости, но возможны ситуации, когда вам потребуется запустить сервер по-другому. В табл. 4.2 описаны преимущества и недостатки каждого из трех методов запуска сервера, рассмотренных в данной главе.

Как правило, большинство серверов в системе Linux (а также программ, реализующих службы, не связанные с взаимодействием по сети) запускаются посредством сценариев SysV. Обращения к некоторым серверам производятся настолько часто, что задержка, связанная с загрузкой сервера с помощью суперсервера, становится недопустимой. С точки зрения составителя дистрибутивного пакета сценарии SysV предпочтительнее локальных сценариев запуска, поскольку при использовании такого подхода можно легко добавлять

Таблица 4.2. Преимущества и недостатки различных методов запуска серверов

Способ запуска	Преимущества	Недостатки
Сценарии запуска SysV	Сервер может запускаться на различных уровнях выполнения. Сервер обрабатывает запросы без задержки. Настройка осуществляется посредством переименования файлов. Сценарии предоставляют удобные средства для запуска и остановки серверов вручную	Серверы занимают большой объем оперативной памяти. Контроль доступа из-за пределов локальной сети затруднен
Суперсервер	За счет выгрузки редко используемых серверов экономится память. Администратор имеет возможность управлять доступом извне	Большое время отклика сервера. Некоторые серверы ненадежно работают в подобной среде. Для сохранения данных между последовательными запросами приходится принимать дополнительные меры
Локальные сценарии запуска	Быстрый отклик сервера. Если сценарии SysV для сервера отсутствуют, он может быть без труда запущен с помощью локальных сценариев	Плохая интеграция с инструментальными средствами настройки (`chkconfig`, `ksysv` и т. д.). Сценарии запуска различаются в разных версиях системы

новые или удалять существующие сценарии. Кроме того, считается, что некоторые серверы, например Samba, не обеспечивают достаточной надежности, будучи загруженными с помощью суперсервера. Иногда бывает необходимо, чтобы сервер сохранял информацию о запросе, например, `nmbd` должен запоминать имена и адреса компьютеров в сети. Такой сервер нельзя загружать посредством суперсервера, так как после удаления кода программы из памяти эта информация теряется.

Помимо сценариев SysV, суперсерверы также находят широкое применение. Во многих пакетах суперсерверы используются для запуска серверов, время загрузки которых невелико (например, серверов Telnet и FTP). В некоторых случаях такой подход можно использовать и для запуска серверов с большим объемом кода, например Apache, но загрузка Apache посредством суперсервера оправдана, только если обращение к этому серверу выполняется очень редко. Некоторые системы, например Debian, позволяют при установке сервера выбирать, должен ли он запускаться с помощью сценариев SysV или суперсервера. Если же возможность выбора отсутствует, вы можете по окончании инсталляции удалить сценарий SysV данного сервера и включить запись о нем в конфигурационный файл суперсервера.

В большинстве случаев имеет смысл использовать тот суперсервер, который поставляется в составе дистрибутивного пакета Linux (`inetd` или `xinetd`). Если вам необходимы возможности, предоставляемые только `xinetd`, а в вашей системе по умолчанию используется `inetd`, вам придется заменить `inetd` на новый суперсервер. Для этого надо модифицировать сценарий SysV либо запускать новый суперсервер с помощью локального сценария запуска. Сделав это, вы должны настроить суперсервер так, как это было описано в данной главе.

Локальные сценарии запуска лучше всего подходят, если сервер первоначально был установлен для загрузки посредством сценария SysV, но оказалось, что сценарий работает некорректно. У неопытных администраторов подобная ситуация встречается тогда, когда они пытаются инсталлировать сервер, предназначенный для другой системы. В этом случае можно самостоятельно написать сценарий SysV, но если у администратора слишком мало времени или если он не имеет опыта решения подобных задач, он может воспользоваться возможностями локального сценария запуска.

Сервер можно запустить и вручную; для этого достаточно ввести в командной строке его имя. Очевидно, что такой способ применим только для проверки работы программы. После окончания отладки придется выбрать способ автоматического запуска сервера.

Резюме

Для того чтобы использовать компьютер под управлением Linux как сервер, надо знать способы запуска серверных программ. Существуют три способа запуска серверов в системе Linux: использование сценариев SysV, запуск под управлением суперсервера и применение локальных сценариев запуска. Детали запуска серверов различаются в зависимости от версии Linux, поэтому, организуя работу серверов, необходимо подробно изучить документацию на вашу операционную систему. Особенно важно знать о расположении и последовательных номерах сценариев запуска SysV, а также о том, какой суперсервер используется в системе: `inetd` или `xinetd`. Зная методы запуска серверов, вы сможете заниматься установкой как широко распространенных, так и редко встречающихся специализированных серверов, а также других сетевых средств.

ЧАСТЬ II

Серверы в локальных сетях

Глава 5

Распределение IP-адресов с помощью DHCP

В главе 2 рассматривались различные способы настройки компьютера для работы в сетях TCP/IP. Один из этих способов предполагал использование сервера DHCP (Dynamic Host Configuration Protocol — протокол динамической настройки узла). Чтобы определить расположение сервера DHCP, компьютер передает широковещательный запрос. В ответ сервер DHCP возвращает компьютеру его IP-адрес и другие сведения, необходимые для настройки. Такое взаимодействие существенно упрощает настройку компьютеров, выполняющих функции клиентов DHCP, так как исключает необходимость вводить IP-адрес машины, IP-адрес шлюза и другие конфигурационные данные. Однако система DHCP не возникает в сети сама собой. Вам необходимо сконфигурировать сервер DHCP так, чтобы он отвечал на запросы клиентов DHCP. Настройке сервера DHCP посвящена данная глава.

Прежде чем приступить к установке конфигурации сервера DHCP, вам надо убедиться в том, что он действительно нужен. Если это так, вы можете приступать к формированию конфигурационных файлов DHCP. В простейшем случае сервер DHCP распределяет IP-адреса между клиентами таким образом, что после перезагрузки адрес клиента DHCP может измениться. Более сложная конфигурация позволяет присваивать компьютеру один и тот же IP-адрес. Кроме выбора конфигурации, в данной главе рассматриваются также вопросы интеграции сервера DHCP с другими серверами, например Samba или DNS.

Эта глава не содержит исчерпывающей информации о работе сервера DHCP, а призвана лишь дать читателю основные понятия о его настройке. Тем не менее данные, приведенные здесь, часто оказываются достаточными для администрирования простых сетей. Если вам необходимо настроить сервер DHCP для выполнения сложных задач, то, возможно, потребуется дополнительная литература. В качестве дополнительных источников информации можно посоветовать книги Дромса (Droms) и Лемона (Lemon) *The DHCP Handbook: Understanding, Deploying, and Managing Automated Configuration Services* (New Riders Publishing, 1999) и Керчевала (Kercheval) *DHCP: A Guide to Dynamic TCP/IP Network Configuration* (Prentice Hall, 1999).

Использование сервера DHCP

Очевидно, что сервер DHCP имеет смысл устанавливать в том случае, если в сети присутствуют клиенты DHCP. Но при настройке клиентских машин также возникает вопрос: следует ли инсталлировать на них клиент-программы DHCP. Такая программа нужна, только если в сети имеется сервер DHCP. Круг замкнулся. Чтобы "разорвать" его, надо рассмотреть сеть в целом и выяснить, что проще: задавать для каждого компьютера статический IP-адрес или установить и настроить один сервер DHCP.

Чтобы оценить трудоемкость установки статического IP-адреса на компьютере, надо вспомнить материал, изложенный в главе 2. Необходимо также принять во внимание тот факт, что в различных операционных системах сетевые средства настраиваются поразному. Сервер DHCP, выполняющийся на компьютере под управлением Linux, может предоставлять IP-адреса клиентам, выполняющимся не только в среде Linux, но и на компьютерах, на которых установлены другие системы: различные версии UNIX, Windows, MacOS, OS/2, BeOS и т. д. Клиенты DHCP могут работать в любой системе, поддерживающей стек протоколов TCP/IP. При настройке компьютера для использования статического IP-адреса во всех системах задается приблизительно одинаковая информация; различаются лишь способы ее ввода. Практически во всех системах для инсталляции и настройки клиента DHCP требуется затратить меньше усилий, чем для установки статического IP-адреса.

Компьютер, на котором выполняется клиент DHCP, может выполнять роль сервера для других протоколов. В большинстве случаев подобная ситуация нежелательна, так как при этом могут возникать проблемы с использованием IP-адреса. Адрес компьютера, выступающего в роли сервера, должен быть неизменным. Далее в этой главе рассказывается, как настроить сервер DHCP, чтобы компьютеру каждый раз выделялся один и тот же IP-адрес. Однако в любом случае лучше отказаться от услуг сервера DHCP и использовать статический IP-адрес. Дело в том, что сервер DHCP может работать некорректно и сервер, расположенный на компьютере, который получает адрес средствами DHCP, окажется недоступным.

Для настройки сервера DHCP обычно требуется намного больше усилий, чем для установки статического IP-адреса на одном компьютере. Поэтому устанавливать сервер DHCP для обслуживания двух клиентов нецелесообразно. Работа по инсталляции и настройке сервера DHCP окупит себя только в том случае, если сеть насчитывает не меньше пяти-десяти узлов. (Если на компьютере установлены две системы, его надо считать как две машины, так как устанавливать IP-адреса придется в каждой системе.)

При оценке целесообразности использования системы DHCP надо учитывать не только трудозатраты по конфигурированию сервера DHCP и клиентов, но и квалификацию сотрудников, которые будут выполнять настройку. Если пользователи должны сами настраивать свои системы, то в этом случае почти наверняка надо отдать предпочтение DHCP. При использовании клиентов DHCP существенно снижается вероятность того, что пользователь неправильно поймет инструкцию и неверно установит адрес. Если же настройку сетевых средств на всех машинах будет выполнять квалифицированный администратор, этот фактор становится менее важен.

Несмотря на то что DHCP может значительно упростить настройку сети, в ней должен остаться по крайней мере один компьютер, использующий статический IP-адрес. Этим

компьютером является сам сервер DHCP. Очевидно, что сервер надо настроить так, чтобы он не пытался выделить свой IP-адрес клиенту.

Компьютер с несколькими сетевыми интерфейсами может выполнять роль КЛИЕНТА DHCP в одной сети и выступать в качестве сервера DHCP в другой.

Работа сети во многом зависит от надежности сервера DHCP. В случае неисправности сервера DHCP клиент не сможет вести переговоры о выделении IP-адреса, следовательно, не получит адрес при загрузке. Поэтому, занимаясь администрированием сети, необходимо предусмотреть резервный сервер DHCP. Этот сервер должен быть неактивен, но готов начать работу в любой момент, когда основной сервер DHCP выйдет из строя. По возможности надо обеспечить копирование файла основного сервера, содержащего информацию о выделенных адресах (файла аренды), на резервный сервер. Это нужно для того, чтобы резервный сервер не пытался выделить IP-адрес, который основной сервер уже присвоил другому компьютеру. Организовать периодическое копирование файла с данными о выделенных адресах можно с помощью инструмента `cron`. Следует заметить, что резервный сервер DHCP переводит существенную часть ресурсов небольшой сети в категорию накладных расходов. Работоспособность сети в случае выхода сервера DHCP из строя можно повысить, увеличивая время аренды. В этом случае работающие клиенты реже пытаются обращаться к неработающему серверу DHCP.

Сервер DHCP обязательно входит в состав дистрибутивного пакета Linux, чего нельзя сказать об остальных системах. Поэтому при формировании сети имеет смысл планировать установку сервера DHCP на компьютере под управлением Linux. В небольшой сети можно использовать *широкополосный маршрутизатор* — устройство, которое позволяет подключать офисную или домашнюю сеть к Internet посредством кабельного или DSL-соединения. Такое устройство обычно содержит сервер DHCP. Сервер на широкополосном маршрутизаторе легче настраивать, чем сервер DHCP в системе Linux, однако он обеспечивает меньшую степень гибкости. Например, сервер DHCP широкополосного маршрутизатора, как правило, нельзя настроить так, чтобы конкретному клиенту выделялся один и тот же IP-адрес.

Для запуска сервера DHCP обычно используется сценарий SysV. Благодаря этому время отклика становится минимальным, а сервер получает возможность хранить важные конфигурационные данные в памяти. (Вопросы запуска серверов рассматривались в главе 5.)

Настройка ядра и сетевых интерфейсов

Для того чтобы иметь возможность использовать сервер DHCP, надо правильно выбрать конфигурацию ядра системы, а также настроить сетевые средства. В частности, вам необходимо установить опции ядра Packet Socket и Socket Filtering. (Версия 1 dhcpd не требует Socket Filtering; эта опция нужна для новых версий программы.) Вопросы установки конфигурации ядра подробно рассматривались в главе 1.

Некоторые клиенты DHCP требуют, чтобы сервер передавал ответы на запросы по адресу 255.255.255.255. Однако по умолчанию система Linux заменяет этот адрес на широковещательный адрес локальной сети (например, 192.168.1.255). Если при работе некоторых клиентов DHCP возникает проблема (обычно это проявляется в системе Windows), ее можно устранить, включив в таблицу маршрутизации компьютера, на котором

расположен сервер DHCP, специальный маршрут. Для этого используется следующая команда:

```
# route add -host 255.255.255.255 dev ethO
```

Имя `eth0` следует заменить на имя сетевого интерфейса, используемого для подключения к вашей локальной сети. Подобную команду можно включить в сценарий запуска. Чтобы проверить установленный маршрут, надо ввести в командной строке команду `route -n`. В результате ее выполнения будут выведены все записи, содержащиеся в таблице маршрутизации. Если маршрут 255.255.255.255 содержится в таблице, он должен находиться в начале списка.

Конфигурационные файлы DHCP

Большинство дистрибутивных пакетов Linux содержит сервер DHCP, разработанный Internet Software Consortium (`http://www.isc.org/products/DHCP/`). Internet Software Consortium (ISC) в конце 2000 г. выпустил версию 3.0 DHCP, но в начале 2002 г. многие версии Linux все еще поставлялись со старой версией 2.0 сервера DHCP. Большинство параметров настройки, рассматриваемых в данном разделе, применимо к версиям 2.0 и 3.0, но версия 3.0 поддерживает некоторые новые возможности, например, средства интеграции с DNS-сервером, которые будут обсуждаться далее в этой главе.

Для настройки сервера DHCP используется конфигурационный файл `dhcpd.conf`, который обычно располагается в каталоге `/etc` или `/etc/dhdcp`. Подобно остальным конфигурационным файлам Linux, `dhcpd.conf` — это текстовый файл, для редактирования которого можно использовать обычный текстовый редактор. Кроме того, во время работы программа `dhcpd` создает собственный файл состояния `dhcp.leases`, который обычно помещается в каталог `/var/lib/dhcp`. В файле `dhcp.leases` содержится информация об аренде адресов. В системе DHCP распределением IP-адресов занимается сервер DHCP. Он сообщает клиенту DHCP, что тому на определенный период времени выделяется некоторый IP-адрес. Другими словами, адрес дается клиенту в *аренду,* а клиент должен вовремя *продлевать ее.* В файле `dhcp.leases` также содержатся сведения об Ethernet-адресах клиентов. Файл `dhcp.leases` не является конфигурационным файлом в полном смысле слова; его нельзя редактировать, но при возникновении проблем или в случае, если вам потребуется выяснить аппаратный адрес, или MAC-адрес, клиента, вы можете просмотреть содержимое `dhcp.leases`.

Строки файла `dhcpd.conf`, начинающиеся с символа #, содержат комментарии. Помимо комментариев, в этом файле находятся выражения, которые делятся на две категории.

- Параметры. Параметры сообщают серверу DHCP о том, надо ли выполнять некоторые действия (например, предоставлять адреса неизвестным клиентам), как их выполнять (например, как долго может длиться аренда адреса), и о том, какая информация должна предоставляться клиентам (например, адрес шлюза).
- **Декларации**. Декларации описывают топологию сети (адреса, связанные с конкретными сетевыми интерфейсами), определяют IP-адреса, которые должны выделяться клиентам, а также связывают набор параметров с набором деклараций.

В некоторых декларациях используется информация, заданная посредством параметров. В этом случае параметры должны предшествовать декларациям. Реальные конфи-

гурационные файлы сервера DHCP обычно начинаются с определения параметров, а за параметрами следуют декларации.

Некоторые декларации могут занимать несколько строк. В них используется группа параметров. Данные, определяемые такими декларациями, помещаются в фигурные скобки. Например, приведенный ниже фрагмент файла `dhcpd.conf` представляет собой декларацию, которая определяет конкретный компьютер и указывает, какой адрес должен выделяться ему.

```
host teela {
    hardware Ethernet 00:05:02:a7:76:da;
    fixed-address 192.168.1.2;
}
```

Конкретный смысл подобных деклараций будет рассматриваться позже. Сейчас достаточно заметить, что декларация начинается с ключевого слова `host`, за которым следует дополнительная опция (имя компьютера `teela`), а строки, содержащиеся в фигурных скобках, определяют характеристики данной декларации. Декларации, состоящие из нескольких строк, могут содержать вложенные декларации. Рекомендуется располагать вложенные декларации с отступом так, чтобы их можно было заметить с первого взгляда. Такой отступ не обязателен; программа dhcpd игнорирует лишние пробелы за исключением тех случаев, когда они указаны в кавычках.

Динамическое распределение IP-адресов

Наиболее просто устанавливается конфигурация сервера DHCP, предполагающая динамическое распределение IP-адресов. В этом случае сервер сам решает, какой адрес следует выделить компьютеру, который обратился к нему. IP-адреса для конкретных машин не резервируются, поэтому IP-адрес, который клиент получил в данный момент, в общем случае может отличаться от IP-адреса, выделенного ему при прошлом обращении к серверу. Клиент может запросить конкретный адрес; более того, большинство клиентов запрашивают именно тот адрес, который они использовали ранее, но сервер игнорирует подобную информацию в составе запроса. На практике компьютеры могут достаточно долгое время работать с одним и тем же IP-адресом. Это происходит, если компьютер редко выключается либо если он был отключен в течение короткого промежутка времени. Тем не менее наличие постоянного IP-адреса не гарантируется. Конфигурация, осуществляющая динамическое распределение адресов, удобна в сетях, содержащих большое число клиентов, однако она имеет существенный недостаток: при динамическом распределении IP-адресов затрудняется использование доменных имен, так как серверы DNS связывают имена со статическими IP-адресами. Существуют способы устранения этого недостатка; их мы рассмотрим ниже. Один из способов предполагает организацию совместной работы серверов DHCP и DNS, а другой способ состоит в выделении компьютерам фиксированных IP-адресов.

Установка глобальных параметров

В листинге 5.1 приведен пример содержимого файла `dhcpd.conf`, предназначенного для организации динамического распределения IP-адресов. Несмотря на то что данный

конфигурационный файл очень прост, его можно использовать на практике для обеспечения работы небольших сетей.

Листинг 5.1. Простой файл dhcpd.conf

```
default-lease-time 7200;
max-lease-time 10800;
option subnet-mask 255.255.255.0;
option routers 192.168.1.1;
option domain-name-servers 192.168.1.1, 172.17.102.200;
option domain-name "threeroomco.com";

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.50 192.168.1.150;
}
```

В первых шести строках файла задаются глобальные параметры, или опции; они определяют основные характеристики сети и сервера и используются при обслуживании всех клиентов. Значения, установленные с помощью глобальных параметров, могут быть переопределены в декларациях. Первые две строки (параметры default-lease-time и max-lease-time) задают длительность аренды в секундах. Подобно тому, как домовладелец и наниматель договариваются о сроках аренды квартиры, клиент DHCP и сервер договариваются о времени, в течение которого клиент может пользоваться IP-адресом. Параметр default-lease-time определяет время аренды, наиболее приемлемое для сервера DHCP. В приведенном выше примере его значение составляет 7200 секунд, или 120 минут. Если клиент запросит более длительную аренду, сервер будет исходить из значения max-lease-time; в данном случае оно равно 10800 секундам, или 180 минутам. При создании реального конфигурационного файла вы можете увеличивать или уменьшать приведенные здесь значения в зависимости от собственных потребностей. Малая длительность аренды снижает работоспособность сети при выходе сервера DHCP из строя и увеличивает нагрузку на сеть. Слишком большая длительность аренды опасна тем, что имеющиеся в наличии IP-адреса будут исчерпаны. Подобная ситуация возможна, если компьютеры будут часто включаться на короткое время; при этом сервер DHCP будет хранить информацию об аренде адресов, которые на самом деле не используются. Для тестирования сервера DHCP целесообразно устанавливать малое время аренды, например 60 секунд; в этом случае вы сможете следить за результатами изменения конфигурации сервера. Если структура сети претерпевает постоянные изменения (например, портативные компьютеры часто подключаются к сети и отключаются от нее), лучше всего подходят значения, близкие к указанным в листинге 5.1. Если же основную часть сети составляют компьютеры, которые не выключаются в течение нескольких дней, то значения параметров default-lease-time и max-lease-time могут быть порядка сотен тысяч секунд.

Следующие четыре строки листинга задают глобальные параметры, которые передаются клиенту DHCP, а именно: маска подсети, адрес маршрутизатора (шлюза), адреса серверов DNS и доменное имя. Как вы, вероятно, помните из главы 2, те же сведения используются при установке статического IP-адреса компьютера. Несмотря на то что в листинге указаны IP-адреса, вы можете также использовать доменные имена узлов, но

при этом необходимо быть уверенным, что сервер DNS работает корректно и связь с ним надежна. Если вы зададите доменные имена, то перед передачей их клиенту сервер DHCP будет преобразовывать их в IP-адреса. Для параметров, не указанных в листинге 5.1, используются значения по умолчанию. При необходимости их можно переназначить явно. Как видно на рис. 5.1, определение параметра заканчивается символом `;`. Ниже приведены некоторые параметры, которые вы, возможно, захотите включить в файл `dhcpd.conf`.

- `filename "имя_файла"`. Сервер dhcpd может выполнять функции сервера загрузки для узлов сети. Настроенный подобным образом, сервер DHCP должен предоставлять клиенту сведения о расположении исходного файла загрузки; такая информация задается с помощью параметра `filename`. После этого клиент может скопировать содержимое данного файла с помощью соответствующего протокола.

- `next-server "имя_узла"`. Этот параметр задает имя компьютера, на котором расположен исходный файл загрузки, заданный с помощью параметра `filename`. Если данный параметр не указан, то по умолчанию считается, что файл находится на том же компьютере, что и сервер DHCP.

- `server-name "имя_узла"`. Этот параметр также управляет загрузкой по сети. Он используется для того, чтобы сообщать клиенту имя сервера, на котором расположены файлы, необходимые для загрузки.

- `boot-unknown-clients` *флаг*. Как правило, значение данного флага устанавливается равным `true`, в результате чего `dhcpd` предоставляет IP-адрес любому компьютеру, который передал запрос. Если значение флага равно `false`, сервер обслуживает только те компьютеры, для которых в составе конфигурационного файла содержится декларация `host`.

- `option broadcast-address` *IP-адрес*. Если в вашей сети используется нестандартный широковещательный адрес, его можно задать с помощью данного параметра. Обычно клиенты самостоятельно определяют нужный адрес.

- `get-lease-hostnames` *флаг*. Если значение флага установлено равным `true`, dhcpd обращается к серверу DNS, определяет доменное имя и передает его клиенту вместе с IP-адресом. Это позволяет запускать на клиентских компьютерах программы, использующие доменные имена при взаимодействии по сети (например, почтовые серверы). По умолчанию для данного параметра устанавливается значение `false`.

- `use-host-decl-names` *флаг*. Этот параметр почти эквивалентен параметру `get-lease-hostnames`. Если его значение равно `true`, `dhcpd` не обращается к серверу DNS, а передает клиенту доменное имя, заданное в декларации `host`. По умолчанию принимается значение `true`.

Параметры `get-lease-hostnames` и `use-host-decl-names` определяют, должен ли сервер DHCP предоставлять клиентам доменные имена. Несмотря на то что при использовании `get-lease-hostnames` dhcpd обращается к серверу DNS для получения доменного имени, ни один из этих двух параметров не влияет на работу сервера DNS. Если вы хотите, чтобы при обращении к клиенту DHCP посредством доменного имени обеспечивалась достаточная надежность, вам надо сконфигурировать сервер DHCP так, чтобы он выделял компьютеру фиксированный IP-адрес, или принять меры для организации совместной работы серверов DHCP и DNS.

В файле `dhcpd.conf` могут присутствовать и другие параметры, многие из которых начинаются с ключевого слова `option`. Некоторые параметры указывают расположение различных серверов, например, серверов шрифтов X Window, серверов службы времени и серверов печати. Другие задают характеристики сетевых интерфейсов клиента, например, определяют, может ли клиент выполнять перенаправление IP-пакетов. Подробное описание этих параметров вы найдете на страницах справочной системы, посвященных структуре файла `dhcpd.conf`. Для использования многих параметров на клиентской машине необходимо установить дополнительные программные средства; например, стандартный клиент DHCP не поддерживает работу с серверами шрифтов X Window.

В большинстве случаев параметры указываются в начале файла `dhcpd.conf`. Кроме того, параметры могут присутствовать в декларациях, определяющих свойства подсетей или групп.

Определение диапазона адресов

В листинге 5.1 представлена чрезвычайно простая конфигурация DHCP, в которой определяется один диапазон IP-адресов. Для указания диапазона адресов используется декларация `subnet`, которая имеет следующий вид:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.50 192.168.1.150;
}
```

Данная декларация указывает на то, что сервер действует в сети 192.168.1.0/24. Очевидно, что компьютер должен иметь сетевой интерфейс, с которым связан адрес из этой сети. В пределах данной сети распространяются широковещательные запросы, передаваемые клиентами и обслуживаемые сервером DHCP. Декларация `range` определяет диапазон IP-адресов, из которого сервер выбирает адреса, предоставляемые клиенту. В данном примере это адреса 192.168.1.50-192.168.1.150. При необходимости можно использовать любой диапазон из указанной сети (192.168.1.0/24), важно, чтобы в него не попадали статические IP-адреса компьютеров, в том числе адрес самого сервера DHCP.

В файле `dhcpd.conf` может присутствовать несколько деклараций `subnet`. Если сервер обслуживает несколько сетей и соответственно содержит несколько сетевых интерфейсов, для каждого из интерфейсов должна быть указана подобная декларация. То же самое необходимо сделать, если на компьютере установлен всего один сетевой интерфейс, который связан с несколькими логическими подсетями. Работая с версиями dhcpd, предшествующими 3.0, необходимо включать декларацию `subnet` для каждого интерфейса, независимо от того, обслуживается ли соответствующая сеть сервером DHCP. Например, если два сетевых интерфейса компьютера подключены к сетям

192.168.1.0/24 и 172.20.30.0/24, но сервер DHCP обслуживает только сеть 192.168.1.0/24, в файле dhcpd.conf должна находиться пустая декларация subnet, приведенная ниже.

```
subnet 172.20.30.0 netmask 255.255.255.0 {
}
```

ВНИМАНИЕ Возможна ситуация, когда компьютер, на котором выполняется сервер DHCP, подключен к двум сетям: одну из сетей он обслуживает сам, а в другой сети используется отдельный сервер DHCP. В этом случае целесообразно настроить сервер так, чтобы он не отвечал на запросы, поступающие из второй сети. Если в одной сети присутствуют два сервера DHCP, они должны быть настроены для совместной работы, в противном случае такая конфигурация может стать источником проблем. Для блокирования входящего трафика DHCP из второй сети можно использовать брандмауэр. (Вопросы настройки брандмауэров обсуждаются в главе 25.) Еще лучше перенести сервер DHCP на компьютер, который принадлежит только одной сети.

Эта декларация указывает на то, что сервер не должен обрабатывать запросы DHCP, поступающие с соответствующего интерфейса. При использовании версии 3.0 программы dhcpd включать эту декларацию не обязательно.

Выделение фиксированных адресов

В большинстве случаев динамическое распределение IP-адресов обеспечивает нормальную работу компьютеров. Internet-соединение устанавливается по инициативе клиента, адрес сервера известен клиенту, а сервер получает адрес клиента в составе запроса. Периодическое изменение IP-адреса (учитывая, что адрес изменяется при перезагрузке компьютера) не мешает работе клиента (смена адреса в течение сеанса сделала бы обмен данными невозможным).

Тем не менее бывают ситуации, при которых необходимо настраивать систему DHCP так, чтобы клиентам выделялись фиксированные адреса. Подобным образом приходится поступать в том случае, если на компьютере, выполняющем функции клиента DHCP, работает какой-либо сервер. Использование фиксированных адресов упрощает диагностику сети; в этом случае, вызывая команду ping, не приходится выяснять, какой именно адрес присвоен в данный момент тому или иному компьютеру. При наличии фиксированных IP-адресов появляется также возможность вместо адреса указывать при обращении к компьютеру доменное имя. (Далее в этой главе будет обсуждаться способ связывания доменных имен с динамическими IP-адресами.) Программа dhcpd позволяет присваивать компьютерам фиксированные IP-адреса. Для этого надо задать MAC-адреса и сконфигурировать dhcpd так, чтобы MAC-адресу соответствовал определенный IP-адрес. Конфигурация dhcpd для выделения фиксированных адресов несколько сложнее, чем конфигурация, при которой выполняется динамическое распределение IP-адресов.

Определение MAC-адреса клиента

MAC-адрес используется для организации сетевого взаимодействия на самом низком уровне. При использовании сетевых карт Ethernet MAC-адрес состоит из шести байтов, значения которых обычно представляются шестнадцатеричными числами и разделяются двоеточиями, например 00:80:C8:FA:3B:OA. Каждый пакет, передаваемый устройством

Ethernet по сети, содержит MAC-адрес этого устройства, поэтому программа dhcpd имеет в своем распоряжении данные, идентифицирующие сетевую карту, а следовательно, и компьютер, в состав которого она входит. (Многие операционные системы содержат средства, позволяющие переопределять MAC-адреса, поэтому MAC-адрес не всегда однозначно идентифицирует устройство, однако в подавляющем большинстве случаев такой способ вполне применим.) В зависимости от типа сетевого оборудования, форматы MAC-адреса могут отличаться от формата, используемого Ethernet, но принцип применения таких адресов остается неизменным.

Первые три байта MAC-адреса Ethernet содержат код производителя сетевой карты; остальные три байта устанавливает предприятие, выпускающее Ethernet-карты. Информацию о кодах производителей можно найти по адресу http://www.coffer.com/mac_find/или http://www.cavebear.com/CaveBear/Ethernet/vendor.html. Для настройки DHCP эти сведения не требуются, но вы можете воспользоваться ими, выясняя расположение компьютеров по широковещательным запросам клиентов DHCP. Заметьте, что производитель Ethernet-карты и производитель компьютера — это, как правило, разные компании.

Для того чтобы программа dhcpd могла использовать MAC-адрес при назначении клиенту IP-адреса, надо в первую очередь выяснить MAC-адрес. В зависимости от используемых аппаратных средств и типа операционной системы, под управлением которой работает клиентский компьютер, это можно сделать различными способами. Некоторые производители снабжают свои сетевые платы наклейками с MAC-адресами. Даже если такая наклейка имеется на вашей плате, не всегда имеет смысл открывать компьютер только для того, чтобы узнать MAC-адрес Ethernet-карты. Существуют программные средства, позволяющие получить эту информацию.

Определение MAC-адреса с клиентской машины

При работе с Linux и другими подобными UNIX системами MAC-адрес можно определить с помощью команды ifconfig. Задайте команду ifconfig ethO (или укажите при вызове ifconfig другое имя), и утилита ifconfig выведет информацию об указанном сетевом интерфейсе. Отображаемые данные будут выглядеть приблизительно следующим образом:

```
ethO      Link encap:Ethernet HWaddr 00:80:C6:F9:3B:BA
```

MAC-адрес следует за ключевым словом HWaddr; в данном случае это значение 00:80:C6:F9:3B:BA. Нужная информация будет получена только в том случае, если драйвер Ethernet загружен и интерфейс активен. При этом не имеет значения, связан ли интерфейс со стеком протоколов TCP/IP.

Используя Windows 2000, вы можете выяснить MAC-адрес посредством программы IPCONFIG, которая работает подобно утилите ifconfig системы Linux. Для получения исчерпывающей информации о сетевых интерфейсах, имеющихся в системе, надо задать команду IPCONFIG /ALL. В составе отображаемых данных будет содержаться следующая строка:

```
Physical Address. . . . . : 00-50-BF-19-7E-99
```

В Windows Me используется программа WINIPCFG, выполняющая те же функции, что и IPCONFIG, и предоставляющая графический пользовательский интерфейс. При

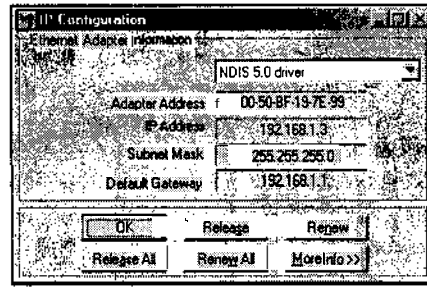


Рис. 5.1. `WINIPCFG` предоставляет информацию о сетевых интерфейсах и позволяет управлять клиентом DHCP в системе Windows 9x/Me

запуске программы выводится окно, представленное на рис. 5.1, в котором MAC-адрес отображается в поле Adapter Address.

Если клиент DHCP расположен на компьютере Macintosh и выполняется в системе MacOS Classic, вы найдете MAC-адрес в окне TCP/IP Control Panel. После щелчка на кнопке Info отобразится диалоговое окно TCP/IP Info, в котором выводится MAC-адрес. В MacOS X данная информация доступна в диалоговом окне Network, показанном на рис. 5.2.

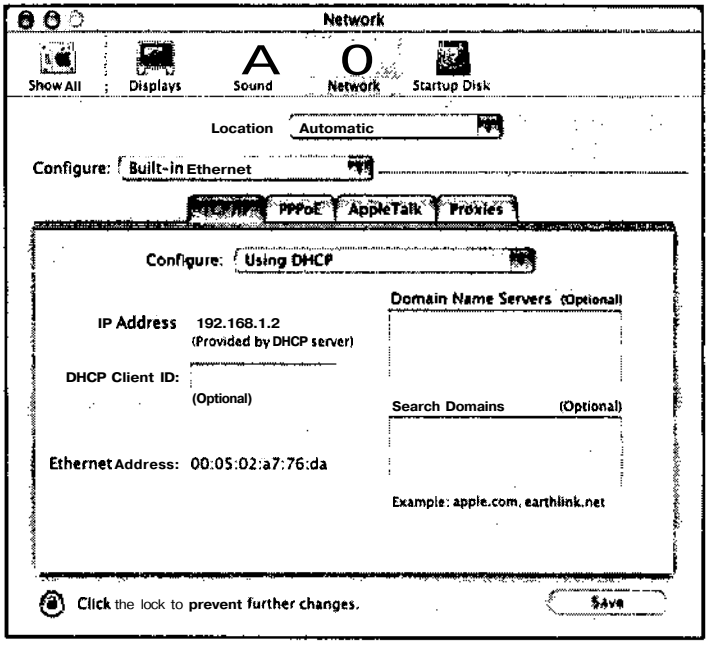


Рис. 5.2. В системе MacOS X MAC-адрес отображается в диалоговом окне Network

Аналогичные средства, позволяющие определить MAC-адрес, имеются и в других операционных системах; для того чтобы использовать соответствующие команды, надо ознакомиться с описанием конкретной системы.

Определение MAC-адреса с сервера

MAC-адрес клиента также можно определить с помощью сервера DHCP. Для этого необходимо, чтобы средства поддержки стека протоколов на клиентском компьютере работали исправно. Сконфигурируйте клиент для работы с сервером DHCP так, чтобы после загрузки клиент получил от сервера динамический IP-адрес. Затем ознакомьтесь с содержимым файла аренды (обычно это файл **/var/lib/dhcp/dhcpd.leases**). В нем вы найдете запись, которая выглядит приблизительно следующим образом:

```
lease 192.168.1.50 {
    starts 4 2002/07/19 21:37:20;
    ends 4 2002/07/19 23:17:20;
    binding state active;
    next binding state free;
    hardware ethernet 00:50:56:82:01:03;
}
```

В приведенной записи указаны IP-адрес, выделенный клиенту, время начала и завершения аренды, а также прочая информация, в том числе MAC-адрес (hardware **ethernet00:50:56:82:01:03**). Чтобы использовать содержимое файла аренды для определения MAC-адреса, вы должны знать, какой IP-адрес выделен клиенту. Это можно определить по времени аренды либо обратиться за этими сведениями на клиентскую машину.

MAC-адреса могут также содержаться в файле протокола Linux (обычно это файл **/var/log/messages**). Последнюю запись, содержащую имя dhcpd, можно найти с помощью следующей команды:

```
# grep dhcpd /var/log/messages I tail -n 1
Jul 19 18:27:38 speaker dhcpd: DHCPACK on 192.168.1.50 to
00:50:56:82:01:03 via eth0
```

Эту команду можно выполнить сразу после того, как сервер DHCP выделит IP-адрес клиенту. Если вы не знаете IP-адреса, присвоенного клиенту, вы рискуете ошибочно определить MAC-адрес. Так может случиться, если после запроса интересующего вас компьютера какой-то из клиентов обратится к серверу DHCP для того, чтобы продлить аренду. Зная IP-адрес, проверьте запись в файле протокола. Если адрес не совпадает, просмотрите другие записи, задавая значения опции -n программы **tail**, отличные от 1.

И, наконец, независимо от того, использует ли клиент статический IP-адрес или получил адрес от сервера DHCP, вы можете определить MAC-адрес с помощью команды **arp**. Вызовите команду **arp** на любом Linux-компьютере вашей сети, указав в качестве параметра IP-адрес клиента.

```
# arp 192.168.1.50
Address          HWtype   HWaddress           Flags Mask   Iface
192.168.1.50     ether    00:50:56:82:01:03   C            ethO
```

Не исключено, что перед вызовом arp вам придется инициировать обмен данными с клиентом. Используйте для передачи пакета программу ping. Соответствующая команда выглядит следующим образом: ping -c 192.168.1.50

Описание узлов с помощью MAC-адресов

Для того чтобы программа dhcpd анализировала MAC-адреса клиентов и предоставляла им фиксированные IP-адреса, надо сначала реализовать динамическое распределение адресов. Вы можете использовать в качестве шаблона код, представленный в листинге 5.1, а затем модифицировать его, например, изменить адреса сервера DNS и шлюзов либо добавить глобальные параметры. Затем необходимо добавить для каждого клиента, которому необходимо выделять фиксированный IP-адрес, декларацию host. Эта декларация может содержаться в декларации subnet либо следовать за ней. Пример декларации host приведен ниже.

```
host teela {
    hardware ethernet 00:05:02:a7:76:da;
    fixed-address 192.168.1.2;
}
```

Декларация начинается с ключевого слова host, за которым следует имя узла без указания имени домена. (Решение о том, должно ли имя домена передаваться клиенту, зависит от наличия других параметров, например use-host-decl-names.) В фигурных скобках указаны два параметра. Первый из них (hardware) задает тип сетевого интерфейса и MAC-адрес, к которому должна применяться эта декларация. В данном примере содержится запись для Ethernet-карты, а при работе в сети иного типа надо задать другой тип сетевого устройства; например, для сети Token Ring следует указать ключевое слово token-ring. Второй параметр (fixed-address) определяет IP-адрес, выделяемый клиенту. Следите за тем, чтобы этот адрес принадлежал сети, которую обслуживает сервер DHCP, и лежал за пределами диапазона, определенного с помощью параметра range, заданного в декларации subnet. В данном примере указан адрес 192.168.1.2, который не относится к диапазону 192.168.1.50-192.168.1.150, указанному в листинге 5.1, но принадлежит сети 192.168.1.0/24, указанной там же.

В файле dhcpd.conf можно определять любое число клиентов, которым должны выделяться фиксированные IP-адреса. Можно также сформировать конфигурационный файл так, что для одних компьютеров сервер DHCP будет выделять фиксированные адреса, а для других — распределять адреса динамически. Если в файле dhcpd.conf содержится и выражение range, и одна или несколько деклараций host, то компьютеры, MAC-адреса которых явно не указаны в декларациях host, будут получать адреса из диапазона, заданного с помощью декларации subnet.

Параметры для отдельных клиентов

Как было сказано ранее, в декларации, состоящей из нескольких строк, могут указываться параметры; они применимы только к текущей декларации. Параметрами являются выражения hardware и fixed-address в декларации host. Для конкретных компьютеров можно задать и другие параметры, в частности, в декларации можно указывать глобальные опции, которые были рассмотрены выше в данной главе. Часто для отдельных компьютеров указывают параметр option host-name "*имя*"; в резуль-

тате сервер DHCP будет передавать имя клиенту. В некоторых случаях этот параметр используется вместо get-lease-hostnames и use-host-decl-names. Кроме того, его можно применять для предоставления имен лишь некоторым из клиентов.

Параметры могут воздействовать на группы клиентов. Один из способов состоит в том, чтобы объединить компьютеры, принадлежащие некоторой группе, в отдельной подсети. Очевидно, что использовать такой способ крайне неудобно. Гораздо лучше создать группу узлов, объединив декларации host в составе декларации group. Соответствующий фрагмент конфигурационного файла будет иметь приблизительно следующий вид:

```
group {
    get-lease-hostnames true;
    host teela {
        hardware ethernet 00:05:02:a7:76:da;
        fixed-address 192.168.1.2;
    }
    host nessus {
        hardware ethernet 00:50:BF:19:7E:99;
        fixed-address 192.168.1.3;
    }
}
group {
    use-host-decl-names true;
    host hindmost {
        hardware ethernet 00:50:56:81:01:03;
        fixed-address 192.168.1.4;
    }
    host louiswu {
        hardware ethernet 00:e0:98:71:60:c1;
        fixed-address 192.168.1.5;
    }
}
```

Имена, предоставляемые первым двум клиентам (teela и nessus), определяются при обращении к серверу DNS. Вторым двум клиентам (hindmost и louiswu) присваиваются имена, указанные в декларации host. Путем группировки декларации можно также решать другие задачи, например, использовать разные файлы загрузки для различных компьютеров (при этом указываются параметры filename и next-server) либо задавать для некоторых узлов сети специальные установки TCP/IP (таким образом можно повысить производительность отдельных компьютеров за счет снижения эффективности работы остальной части сети).

Интеграция с другими серверами

Некоторые варианты настройки сервера DHCP предполагают, что он будет обмениваться данными с другими пакетами. Подобная ситуация возникает в тех случаях, когда клиенты DHCP должны получать дополнительные данные от сервера. Иногда другие серверы могут обращаться к серверу DHCP, чтобы получить данные, опреде-

ляющие их конфигурацию. В данной главе уже обсуждалось использование параметра `get-lease-hostnames`; в этом случае сервер DHCP вынужден обращаться за информацией к серверу DNS.

Включение информации NetBIOS

Система NetBIOS, являющаяся основой для протоколов SMB/CIFS, использует для решения стоящих перед ней задач набор структур и инструментов, действующих на базе протоколов TCP/IP. (Подробно протоколы SMB/CIFS, будут рассматриваться в главе 7.) Сервер DHCP можно настроить так, что он будет предоставлять информацию некоторым из этих структур, выполняющихся на клиентских компьютерах под управлением Windows. В результате повысится эффективность работы системы. Для этого в состав файла `dhcpd.conf` включаются следующие параметры.

- `option netbios-name-servers` *адреса_серверов*. Традиционно NetBIOS применяет систему преобразования имен, которая не имеет ничего общего с системой, с которой работает большинство TCP/IP-клиентов. Компьютеры NetBIOS могут использовать широковещательную модель, согласно которой они передают широковещательные сообщения по локальной сети, либо применять в работе систему преобразования имен, отличную от DNS. Независимые серверы имен называются NBNS (NetBIOS Name Service - служба имен NetBIOS) или WINS (Windows Internet Name Service — межсетевая служба имен Windows). Для того чтобы сервер DHCP сообщал Windows-клиентам адреса серверов, в его конфигурационном файле используются параметры `option netbios-name-servers`. Обычно в файле `dhcpd.conf` указывается один подобный сервер, но система DHCP может предоставлять адреса нескольких серверов WINS.

- `option netbios-node-type` *код_типа*. Этот параметр используется с параметром, рассмотренным выше. Он сообщает клиенту, должен ли тот реализовывать широковещательный принцип преобразования адресов или обращаться к серверу WINS. Код типа — это числовое значение в диапазоне от 1 до 8. Значения 1 и 2 сообщают соответственно об использовании широковещательного преобразования имен и сервера WINS. Значения 4 и 8 позволяют применять оба способа: 4 означает, что клиент должен сначала предпринять попытку широковещательного преобразования, а в случае неудачи обращаться к серверу WINS, а 8 указывает на то, что в первую очередь клиент должен обратиться к серверу WINS, а лишь потом использовать широковещательное преобразование. В большинстве сетей, содержащих сервер WINS, указывается значение 8, поскольку при этом снижается трафик сети и в то же время обеспечивается достаточная надежность. Если этот сервер оказывается неисправным или не содержит данные для конкретного компьютера, преобразование все же выполняется.

- `option netbios-dd-server` *адрес_сервера*. Данный параметр связан с решением самых сложных вопросов обеспечения работы NetBIOS: он задает адрес сервера NBDD (NetBIOS Datagram Distribution — распространение дейтаграмм NetBIOS). Этот сервер передает широковещательный трафик клиентам, которые в обычных условиях не получали бы эти данные. Вероятнее всего, вам не придется использовать этот параметр.

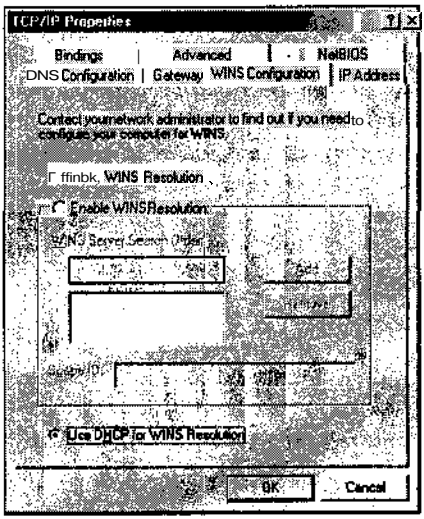


Рис. 5.3. Клиенты Windows можно непосредственно настраивать для использования NetBIOS-данных, предоставляемых сервером DHCP

- option netbios-scope *строка*. Данный параметр определяет область видимости NetBIOS — группу компьютеров, которым известно заданное имя NetBIOS. В большинстве случаев в системе NetBIOS область видимости остается неизменной, в результате любой NetBIOS-компьютер сети может распознавать имена остальных компьютеров. Если в сети устанавливается область видимости, вам придется использовать этот параметр. (При необходимости его можно включить в состав декларацииgroup.)

Если сервер DHCP предоставляет IP-адреса Windows-клиентам, целесообразно использовать первые два из рассмотренных выше параметров как глобальные. (Пакет Samba в системе Linux не использует значения, предоставляемые DHCP; они предназначены в основном для клиентов Windows). Например, вы можете включить в начало файла dhcpd.conf следующие две строки:

```
option netbios-name-servers 192.168.1.1;
option netbios-node-type 8;
```

Чтобы убедиться, что компьютеры под управлением Windows используют эту информацию, надо проверить, установлена ли опция Use DHCP for WINS Resolution в диалоговом окне TCP/IP Properties (рис. 5.3). Если вы выберете опцию Disable WINS Resolution, система вовсе не будет использовать WINS. Установив опцию Enable WINS Resolution, вы сможете вручную задать IP-адрес сервера WINS.

Взаимодействие с DNS-сервером

Если вы хотите, чтобы к клиенту DHCP мог непосредственно обратиться любой узел сети, добиться этого можно двумя способами. Вы можете сконфигурировать сервер DHCP

так, чтобы он предоставлял клиенту фиксированный IP-адрес (необходимые действия обсуждались ранее в этой главе), либо настроить серверы DHCP и DNS для взаимодействия друг с другом; в этом случае записи сервера DNS будут обновляться, отражая текущее назначение адресов клиентам DHCP. Если адреса клиентов DHCP изменяются редко, а обращение к ним выполняется только из локальной сети, второй способ предпочтительнее первого. Если к компьютеру предполагается доступ из Internet, этот способ мало пригоден. Если запись в конфигурационном файле DNS изменится, то для распространения изменений по Internet потребуется определенное время. В течение этого времени могут быть предприняты попытки обращения к компьютеру по его старому адресу.

В версиях dhcpd, предшествующих версии 3.0, отсутствовали средства для поддержки взаимодействия с сервером DNS. В версии 3 были реализованы два способа обновления записей DNS: *ad-hoc* (метод, ориентированный на конкретный узел) и *interim* (промежуточный метод). Существуют также другие способы; после принятия их в качестве Internet-стандарта они будут реализованы в последующих версиях dhcpd.

Прочие способы обновления содержимого сервера DNS реализованы в продуктах независимых производителей. Эти продукты отслеживают изменения в файле протокола DHCP и при появлении в нем записей об изменении соответствия адресов доменным именам, отражают эти изменения в конфигурации DNS.

Для организации взаимодействия с сервером DNS используется параметр `ddns-update-style`, который может принимать одно из трех значений: `ad-hoc`, `interim` и `none` (последнее значение принимается по умолчанию). Чтобы разрешить динамическое обновления DNS, надо включить параметр `ddns-update-style` в качестве глобального в начало файла `dhcpd. conf` и присвоить ему либо значение `ad-hoc`, либо `interim` (попытки задать данный параметр для отдельных клиентов не дадут ожидаемого результата).

ВНИМАНИЕ Оба метода обновления DNS работают только в том случае, если сервер DNS воспринимает информацию об обновлении. Соответствующая конфигурация сервера DNS будет обсуждаться в главе 18.

Метод обновления ad-hoc

При использовании метода ad-hoc имя узла задается одним из четырех способов. Эти способы с учетом их приоритета перечислены ниже.

1. Если в декларации `host` содержится параметр `ddns-hostname`, используется значение этого параметра.
2. Если клиент передает полностью определенное доменное имя (имя, состоящее из имени узла и имени домена), сервер DHCP принимает содержащееся в нем имя узла.
3. Если клиент передает имя узла без указании имени домена, это имя воспринимается сервером DHCP.
4. Используется имя клиента, указанное в декларации `host`.

В любом случае применяется только локальное имя компьютера. Если его невозможно определить ни одним из этих способов, сервер DHCP не предпринимает попыток обновить данные сервера DNS. Если сервер DHCP имеет в своем распоряжении имя узла, он

объединяет его с именем домена, заданным с помощью параметра ddns-domainname (если этот параметр присутствует), либо посредством параметра domain-name.

Сформированное полное доменное имя сервер DHCP использует для изменения данных сервера DNS. Сначала обновляется запись A, которая управляет прямым преобразованием (когда по заданному имени определяется IP-адрес). Если запись A создается успешно, то сервер DHCP обновляет запись PTR, используемую для обратного преобразования (когда по заданному IP-адресу определяется доменное имя узла).

Метод обновления interim

Метод `interim` во многом похож на метод ad-hoc, но он предоставляет возможность клиенту обновить запись A сервера DNS. Чтобы разрешить или запретить клиенту самостоятельно выполнять действия по обновлению записей сервера DNS, в файл `dhcpd.conf` надо включить параметр `allow client-updates` или `ignore client-updates`. По умолчанию подобные запросы клиента принимаются для обработки.

Если сервер DHCP сконфигурирован так, что клиентам разрешено обновлять данные сервера DNS, то для создания записи PTR сервер DHCP использует полное доменное имя, переданное клиентом. Если действия клиента по обновлению данных сервера DNS запрещены, сервер DHCP создает полное доменное имя из имени узла, полученного от клиента, и имени домена, заданного в файле `dhcpd.conf`. Сформированное доменное имя используется для обновления записей A и PTR.

Если клиенту запрещено изменять записи сервера DNS, то метод `interim` дает те же результаты, что и метод ad-hoc. Если вмешательство клиента в работу сервера DNS разрешено, ситуация несколько изменяется. Клиент может задать имя домена, отличное от того, которое указано при настройке сервера DHCP. Предположим, например, что в конфигурационном файле сервера DHCP указано имя домена `threeroomco.com`. При использовании метода ad-hoc клиент получает имя, принадлежащее `threeroomco.com`, даже если в его запросе был указан другой домен. Если же клиент имеет право изменять записи сервера DNS, он может задать любое имя, например `dino.pangaea.edu`. После обновления записи A по этому имени можно обращаться к клиенту. Если сервер DHCP корректно взаимодействует с сервером DNS, при создании записи PTR также будет использовано имя `dino.pangaea.edu`. Это имя будет возвращаться при обратном преобразовании. Ответить на вопрос о том, допустима ли такая конфигурация, предстоит вам. Если вашим пользователям необходимы имена, принадлежащие другим доменам, такая конфигурация может быть оправдана. Если же ваша сеть легко доступна извне, от подобных имен следует воздержаться, так как в результате выполнения обратного преобразования возможно некорректное использование имен, кроме того, хакеры получают возможность маскировать свою нелегальную деятельность.

Динамические средства DNS

Многие пользователи получают доступ к Internet через кабельные модемы и DSL-соединения. Для присвоения их компьютерам IP-адресов используются DHCP или другие протоколы. Если в подобной ситуации вам потребуется поставить в соответствие своему компьютеру фиксированное доменное имя, вы не сможете организовать совместную работу DHCP и DNS в системе Linux. В этом случае целесообразно использовать один из бесплатных пакетов динамической поддержки DNS. (Если подобный пакет распространяется на коммерческой основе, цена его обычно небольшая.)

Серверы динамической поддержки DNS обеспечивают преобразование имен узлов Internet. На своем компьютере вы можете установить клиентский пакет, который при изменении IP-адреса будет обращаться к серверу DNS провайдера. В результате вы получите возможность использовать фиксированное доменное,иш прйнадлежащее либо домену, поддерживаемому динамическими средствами DNS провайдера, либо вашему собственному домену. Этому имени будет соответствовать IP-адрес вашего компьютера. Многие провайдеры, предоставляющие услуги динамической DNS, распространяют программы, написанные на Perl или другом языке сценариев, большинство из которых могут выполняться в среде Linux.

Списки провайдеров, предоставляющие услуги динамической DNS, можно найти по адресам `http://www.technopagan.org/dynamic/`, `http://www.geocities.com/kiore_nz/` и `http://dns.highsynth.com`. Если вы не захотите поддерживать доступный извне сервер DNS, вы наверняка найдете в этих списках информацию о провайдерах, услуги которых соответствуют вашим потребностям.

Резюме

Протокол DHCP может успешно использоваться как в больших, так и в малых сетях. Сосредоточив информацию о конфигурации на одном сервере, вы существенно упростите настройку сетевых средств на клиентских машинах, работающих под управлением различных операционных систем. Следует однако помнить, что сервер DHCP также необходимо настраивать и поддерживать; если такой сервер будет сконфигурирован неверно или если он выйдет из строя, клиентские машины не смогут нормально работать в сети. Сервер DHCP можно настроить так, что он будет выделять клиентам произвольные IP-адреса из пула, либо указать при настройке, что определенным узлам сети должны предоставляться фиксированные адреса. Если вы хотите, чтобы клиенты DHCP были доступны по именам, вам необходимо использовать фиксированные адреса. Доменные имена можно связывать и с динамически выделяемым IP-адресами; для этого надо организовать совместную работу серверов DHCP и DNS. Соответствующие средства реализованы в версии 3 сервера DHCP для Linux.

Глава 6

Аутентификация средствами Kerberos

В системе Linux обычно используется *локальная аутентификация*. Пользователь вводит имя и пароль, а компьютер ищет соответствующие данные в своей базе и принимает решение о том, следует ли зарегистрировать пользователя в системе. Подобный принцип идентификации применяется и в серверах (например, в POP- и FTP-серверах), для работы с которыми необходимо указать пароль. Если пользователи должны регистрироваться на различных компьютерах, такой подход мало приемлем, так как поддержка учетных записей на различных узлах сети отнимает у администратора слишком много времени. Более того, если пароли передаются по сети (а они часто пересылаются в незашифрованном виде), появляется реальная опасность перехвата пароля и использования его для незаконного доступа к системе. Эти проблемы призвана решить система Kerberos. В этой системе поддерживается централизованная база данных о пользователях. При выполнении аутентификации компьютеры обращаются к этой базе, а кодирование информации исключает возможность перехвата секретных данных.

Название Kerberos пришло из греческой мифологии: так звали трехглавого пса, который охранял вход в царство мертвых. Несмотря на то что в английском языке имя этого мифологического персонажа пишется как Cerberus, разработчики использовали для своей системы греческий вариант имени. Изображение трехглавого пса можно найти на многих Web-страницах, посвященных системе Kerberos.

Для того чтобы использовать систему Kerberos, необходимо знать основные принципы ее работы, а также иметь представление о различных версиях Kerberos и их возможностях. Подобно другим службам, в системе Kerberos применяются клиент-программы и серверы. Вам, как администратору системы, надо уметь настраивать их, поэтому в данной главе рассматриваются оба типа программ.

Работа системы Kerberos основана на использовании протокола Kerberos. Это чрезвычайно сложный протокол; чтобы обеспечить его работу, надо сконфигурировать не только сервер Kerberos, но и другие клиенты и серверы, присутствующие в сети. Если вы не хотите ограничиваться установкой простейших вариантов системы Kerberos,

а собираетесь решать более сложные задачи, вам придется изучить дополнительные документы, специально посвященные работе системы Kerberos. Необходимую информацию можно получить, обратившись на главный Web-узел Kerberos по адресу `http://web.mit.edu/kerberos/www/`. Здесь же находится большое количество ссылок на официальные и неофициальные документы, описывающие Kerberos, а также на программные реализации данного протокола.

Использование системы Kerberos

Для защиты локальной сети от нежелательных обращений извне применяются брандмауэры, которые будут обсуждаться в главе 25 (брандмауэры также защищают внешние узлы от воздействия узлов локальной сети, при настройке которых были допущены ошибки). В отличие от брандмауэров, система Kerberos представляет собой внутреннюю систему защиты. Kerberos гарантирует, что компьютерам, с которыми взаимодействуют клиенты и серверы, действительно предоставлено право обмена данными с ними. Кроме того, эта система обеспечивает защиту паролей, передаваемых по сети, а также кодирует информацию, которой внешние клиенты обмениваются с серверами локальной сети. Помимо действий по защите, Kerberos упрощает администрирование системы. При работе этой системы создается централизованная база паролей, в результате чего пользователь, зарегистрировавшись один раз, получает доступ к почтовым серверам, FTP-серверам и другим службам сети, для работы с которыми необходимо указывать пароль.

Kerberos часто применяется в средних и больших сетях, например, в сетях колледжей, университетов и небольших корпораций. В таких сетях обычно работают почтовые серверы, серверы печати и другие службы. Зарегистрировавшись в подобной сети, пользователь может работать с разными серверами и обращаться к различным рабочим станциям.

Kerberos представляет собой межплатформенную систему. Клиенты и серверы Kerberos созданы для Linux и других UNIX-подобных систем, Windows, MacOS и т. д. (Система Kerberos, реализованная Microsoft, не всегда совместима со стандартной версией системы. На Web-странице Kerberos, поддерживаемой MIT, расположены ссылки на другие реализации Kerberos, которые могут работать в Windows и свободны от данного недостатка.) Межплатформенная совместимость является чрезвычайно важной характеристикой во многих средах.

Для эффективного использования Kerberos в приложениях должны быть специально предусмотрены средства взаимодействия с этой системой. Например, почтовый клиент и сервер, обменивающиеся данными по протоколу POP, должны поддерживать аутентификацию Kerberos (в противном случае им придется применять внутренние средства аутентификации). В данной главе рассматривается конфигурация системы Kerberos, предназначенной для работы в среде Linux. Несмотря на то что в реальной сети необходимо обеспечивать взаимодействие с другими операционными системами, здесь не будет рассматриваться настройка клиентов Kerberos для Windows, MacOS и других платформ.

Централизованные и распределенные вычисления

В конце 1960-х появились системы, позволяющие нескольким пользователям работать на одном компьютере. В особенности такая тенденция стала заметной в 1980-е годы. В частности, UNIX была создана как многопользовательская система. Многопользовательские компьютеры размещались в машинном зале, а пользователи работали на алфавитно-цифровых терминалах, а впоследствии на X-терминалах, обеспечивающих

работу с графикой. По мере удешевления оборудования рабочие станции стали размещать на рабочих местах пользователей. Этот процесс существенно ускорился при появлении персональных компьютеров x86.

В настоящее время во многих сетях используется децентрализованная распределенная модель вычислений, рабочие станции функционируют под управлением Windows, MacOS, Linux и других разновидностей UNIX. Основная нагрузка по обработке данных лежит именно на них, но в процессе работы эти компьютеры могут обращаться по сети к различным серверам. Подобные сети работают гораздо надежнее, чем системы с одним мэйнфреймом, так как при выходе из строя какого-либо из серверов остальная часть сети продолжает функционировать. Благодаря использованию распределенных вычислений каждый определенную часть обрабатывающих ресурсов, как минимум на ресурсы процессора своей рабочей станции. При работе в локальной сети каждый пользователь "привязан" к своему компьютеру, поскольку именно на нем хранится информация о пользовательском имени и пароле, используемая при аутентификации. Это — одна из проблем, решаемая с помощью Kerberos.

В настоящее время компьютеры x86 обеспечивают гораздо более высокую производительность, чем мэйнфреймы, применявшиеся двадцать лет назад. В результате появилась возможность использовать их для централизованных вычислений. Часто в системе Linux выполняется большое количество пользовательских программ. Пользователи могут работать за менее мощными компьютерами, на которых выполняются программы эмуляции терминалов (эти программы будут обсуждаться в главах 13 и 14). При таком подходе снова появляются проблемы, характерные для централизованных вычислений, например, если центральный компьютер выходит из строя, остальные устройства становятся практически бесполезными. Однако, централизованный подход существенно упрощает администрирование системы, и это может рассматриваться как дополнительный аргумент в пользу применения Kerberos.

Принцип действия Kerberos

Для того чтобы эффективно применять средства Kerberos в сети, надо инсталлировать сервер паролей Kerberos, который также называют *центром распространения ключей* (key distribution center — KDC). Кроме того, необходимо обеспечить поддержку средств Kerberos клиентскими и серверными программами. Программы, настроенные для взаимодействия с Kerberos, часто называют *керберизованными приложениями* (Kerberized application). Чтобы использовать Kerberos, необходимо понимать, как работает протокол Kerberos и как организуется взаимодействие основных компонентов системы. В данном разделе описаны основные принципы действия протокола Kerberos, а также представлена информация об основных продуктах Kerberos и изложены требования к KDC.

Взаимодействие компонентов Kerberos

Кратко Kerberos можно определить как протокол, обеспечивающий централизованную идентификацию пользователей и применяющий кодирование данных для противодействия различным видам атак. Однако такое определение нельзя называть полным.

Система шифрования Kerberos достаточно сложна и решает ряд задач. Структура сети Kerberos также должна быть рассмотрена более подробно.

Сетевые компоненты Kerberos

Основным компонентом системы Kerberos является KDC. Он отвечает за аутентификацию компьютеров в области (realm) Kerberos. Обычно область Kerberos совпадает с некоторым доменом или поддоменом Internet. Например, домен `threeroomco.com` может содержать единственную область Kerberos; в этом случае область скорее всего будет иметь имя `THREEROOMCO.COM`. В отличие от имен доменов Internet, имена областей Kerberos чувствительны к регистру символов. Для того чтобы подчеркнуть различия между доменом Internet и областью Kerberos, которая определяет тот же набор компьютеров, принято задавать имена областей символами верхнего регистра. Область Kerberos может занимать не весь домен либо включать компьютеры из нескольких доменов. Если в одном домене находятся две или более области Kerberos, то для их идентификации в начало имени области добавляются дополнительные компоненты, например `REALM1.THREEROOMCO.COM` и `REALM1.THREEROOMCO.COM`.

В процессе работы система Kerberos выдает *билеты* на использование различных служб. Подобно авиационным или театральным билетам, билеты Kerberos предоставляют право пользования некоторыми услугами. Существуют два основных типа билетов, о которых будет сказано ниже.

Сервером Kerberos будем называть либо компьютер, на котором выполняется серверная программа Kerberos, либо саму программу, т. е. KDC. *Клиент* Kerberos — это компьютер либо программа, которые получают билет от сервера Kerberos. Обычно считается, что действия системы Kerberos инициирует пользователь, который отправляет запрос на получение услуг от некоторого сервера приложения (например, сервера печати).

Kerberos предоставляет билеты *принципалам,* в роли которых выступают пользователи либо серверные программы. Для описания принципала применяется идентификатор, состоящий из трех компонентов: *основы* (primary), *экземпляра* (instance) и *области* (realm). Этот идентификатор записывается в формате *основа/экземпляр@область*.Если билет получает пользователь, основа представляет собой пользовательское имя. В роли принципала может также выступать сервер; в этом случае основой является имя сервера, например `ftp`. Экземпляр — не обязательный компонент, он применяется в тех случаях, когда одна и та же основа используется в различных целях. Предположим, что пользователю `fluffy`поставлены в соответствие два принципала: один, используемый для решения обычных задач, и второй, предназначенный для выполнения действий по администрированию системы. Для идентификации второго принципала может быть использован экземпляр `admin`. Если область имеет имя `THREEROOMCO.COM`, то идентификаторы принципалов будут иметь вид `fluffy@THREEROOMCO.COM` и `fluffy/admin@THREEROOMCO.COM`.

Задачи, выполняемые Kerberos

Для того чтобы понять работу средств Kerberos, надо рассмотреть задачи, решаемые данной системой. Эти задачи кратко описаны ниже.

- Обеспечение аутентификации в сети. Чтобы предотвратить неавторизованный доступ к службам, сервер должен иметь возможность идентифицировать пользователей. Кроме того, в некоторых средах важно, чтобы клиент мог идентифицировать

серверы. Это исключит работу пользователей с фальшивыми серверами, созданными специально для сбора важной информации.

- **Защита паролей.** Многие службы по умолчанию используют незашифрованные пароли. Это создает угрозу безопасности системы, так как незакодированные пароли могут быть перехвачены и использованы для незаконного доступа к ресурсам. В некоторых серверах предпринимается попытка решения данной проблемы путем кодирования паролей, но в Kerberos используется нестандартный подход. Вместо того чтобы передавать пароль в зашифрованном виде, система использует его в качестве ключа для кодирования передаваемых данных. При этом пароль не передается, но данные может получить только тот пользователь, который знает пароль.

Многие приложения, использующие пароль для доступа к удаленным системам, предоставляют пользователю возможность сохранения пароля. Такое поведение типично, например, для почтовых серверов (POP и IMAP). Пользоваться этой возможностью крайне нежелательно, так как в случае взлома рабочей станции пароль окажется в руках злоумышленника. При работе в сети возникает также проблема смены пароля. Новый пароль приходится задавать сразу в нескольких программах. Обе проблемы позволяет решить Kerberos.

- **Обеспечение однократной регистрации в сети.** Kerberos дает возможность пользователю работать в сети, зарегистрировавшись лишь на своем компьютере. Для обмена с керберизованными приложениями вводить пароль не требуется. В частности, достаточно зарегистрироваться один раз, чтобы получать почту с помощью керберизованной почтовой системы или обращаться к другому компьютеру, на котором выполняется керберизованный сервер регистрации. (Описанные возможности не распространяются на взаимодействие с внешними системами). Срок действия билета, который пользователь получает при вводе пароля, ограничен, поэтому при работе в системе в течение длительного времени приходится вводить пароль повторно. Однако при обращении к серверу в течение времени жизни билета задавать пароль не нужно.

Средства Kerberos должны также удовлетворять некоторым требованиям, связанным с технической реализацией системы, но основное влияние на выбор принципа работы Kerberos оказали три описанные выше задачи. При решении их Kerberos использует билеты. Процесс взаимодействия с сервером включает следующие этапы.

1. Пользователь, работающий на рабочей станции, собирается воспользоваться некоторыми услугами, для чего вводит имя и пароль.
2. Рабочая станция (клиент Kerberos) передает имя пользователя KDC и запрашивает *TGT* (ticket-granting ticket — билет на получение билета). Этот запрос обрабатывается специальным компонентом Kerberos, который называется *TGS* (ticket-granting service — служба получения билета).
3. KDC ищет имя пользователя в базе данных. Если имя присутствует в ней, KDC возвращает TGT. В этом билете содержится имя пользователя, которому он предназначен, время выдачи билета и время, в течение которого он остается действительным. KDC кодирует TGT, используя для этого пароль, содержащийся в базе данных, и передает его клиенту.

4. Клиент получает TGT и расшифровывает его с помощью пароля, введенного пользователем. Если попытка расшифровки оказывается успешной, клиент сохраняет билет для дальнейшей работы. Все действия с билетом остаются прозрачными для пользователя.

5. Используя данные, содержащиеся в билете, клиент отправляет KDC запрос на получение такого билета, который давал бы возможность взаимодействовать с требуемым сервером. Поскольку данные были успешно расшифрованы, а затем снова зашифрованы, KDC принимает их как корректные и передает серверу новый билет. Этот билет зашифрован паролем целевого сервера (его знают только сервер и KDC) и содержит имя пользователя, инициировавшего запрос, имя службы, доступ к которой должен быть предоставлен, время выдачи билета, время его действия, код сеанса и другую информацию. Код сеанса выполняет роль нового пароля; этот пароль создан KDC и предназначен для совместного использования клиентом и сервером. Для того чтобы уменьшить риск перехвата и незаконного использования информации, устанавливается малое время действия билета.

6. Клиент принимает билет на получение услуг, но не предпринимает попытки расшифровать его (действия с этим билетом также прозрачны для пользователя).

7. Клиент передает билет на получение услуг целевому серверу. Этот билет рассматривается как запрос на инициализацию сеанса передачи данных.

8. Сервер расшифровывает билет, пользуясь для этого паролем. Попытка расшифровки будет успешной только в том случае, если билет был корректно закодирован KDC. Если запрос корректен (билет получен от действительного пользователя, успешно расшифрован и т. д.), сервер использует код сеанса для кодирования ответа клиенту.

9. Клиент получает ответ сервера. Если данные корректны, клиент предполагает, что сервер аутентифицирован, завершает процедуру установления соединения и начинает передачу информации. Информация передается только в том случае, если от сервера получен допустимый ответ.

С этого момента процесс обмена данными происходит так, как будто система Kerberos не используется, за исключением того, что в составе **керберизованных** приложений действуют средства кодирования и декодирования данных (в обычных приложениях такие средства отсутствуют). Со временем срок действия TGT и билета на получение услуг истекает, но это вряд ли повлияет на обмен информацией, так как срок действия билетов обычно устанавливается равным нескольким часам. Если же такая ситуация возникла в течение сеанса, билеты должны быть обновлены.

ВНИМАНИЕ

Как видно из описанной выше процедуры, в составе билетов содержатся временные отметки. Если таймеры компьютеров, участвующих во взаимодействии, установлены по-разному, обмен данными может не состояться. В некоторых случаях несоответствие системного времени может привести к снижению уровня безопасности системы. Поэтому необходимо синхронизировать таймеры на всех компьютерах, входящих в состав сети Kerberos. Сделать это можно с помощью сервера NTP (Network Time Protocol — сетевой протокол времени), описанного в главе 10.

Требования к серверу Kerberos

Зная принципы работы системы Kerberos, можно сформулировать требования к ее компонентам. Очевидно, что с точки зрения безопасности сети KDC является чрезвычайно важным компонентом системы. Доступ к серверу (равно как и физический доступ к компьютеру, на котором он выполняется) должен иметь только администратор системы. Необходимо уделять внимание своевременному обновлению версий программного обеспечения. Поскольку KDC используется в работе многих серверов, необходимо продумать план его восстановления в случае неисправности. Создавайте резервные копии данных KDC и держите наготове компьютер, где можно было бы запустить KDC в случае выхода из строя машины, на которой выполняется основной сервер. Имеет смысл создать ведомый, или резервный KDC, который получал бы конфигурационные данные от ведущего, или основного KDC, и в случае выхода из строя последнего мог бы взять на себя обязанности по обслуживанию клиентов и серверов в сети.

Характеристики аппаратных средств, необходимых для работы KDC, определяются размерами сети, а именно числом компьютеров и пользователей. Для небольших сетей, насчитывающих порядка двух десятков узлов, для KDC подойдет низкоуровневый компьютер. Система, содержащая процессор Pentium с низким быстродействием, 32 Мбайт оперативной памяти и жесткий диск объемом в несколько сотен мегабайт, будет более чем достаточна для работы KDC в такой сети. Если же к сети подключены сотни, а тем более тысячи машин, потребуется более мощный компьютер, обладающий быстродействующим процессором и обеспечивающий высокую скорость обмена по сети. Размещая KDC в сети, надо выбрать его расположение так, чтобы обеспечивалась максимальная надежность, а время доставки данных клиентам и серверам было минимальным. Например, если сеть состоит из нескольких отдельных сегментов, желательно поместить в каждом из сегментов свой KDC. Если один KDC будет обслуживать несколько сетевых сегментов, то при сбое в работе маршрутизаторов взаимодействие клиентов с серверами станет невозможным.

Версии и разновидности Kerberos

Наибольшей популярностью пользуется пакет Kerberos, доступный по адресу `http://web.mit.edu/kerberos/www/`. На узле MIT размещены исходные тексты Kerberos V5 Release 1.2.1 и двоичные коды, подготовленные для различных операционных систем (версия для Linux отсутствует). Здесь же можно найти Kerberos V4 и версию данной системы для Windows и MacOS (как для MacOS Classic, так и для MacOS X). В версии Kerberos V5 реализованы новые возможности по сравнению с Kerberos V4, кроме того, в последних реализациях устранены недостатки, имевшие место в предыдущих версиях.

Подобно X Window, Kerberos распространяется в исходных кодах. Кроме того, существуют варианты системы, созданные на основе кода MIT, которые предоставляются на коммерческой основе. Одна из разновидностей Kerberos создана в Швеции Королевским технологическим институтом (Royal Institute of Technology) и доступна по адресу `http://www.pdc.kth.se/kth-krb/`. Этот вариант системы называется eBones, но имена файлов, входящих в состав пакета, начинаются с krb4. Официально посредством данного узла распространяются только исходные коды, но на FTP-сервере можно найти каталог `binaries`, в котором расположены двоичные пакеты, включая пакет для Linux. Если вы решите скопировать его, будьте внимательны: это может оказаться старая реализация системы. Система eBones (по крайней мере, eBones 1.1) базируется на Kerberos V4.

Центр параллельных вычислений (Center for Parallel Computers) реализовал систему Kerberos, известную под названием Heimdal; ее коды расположены по адресу `http://www.pdc.kth.se/heimdal/`. Эта разновидность системы базируется на MIT Kerberos V5. На указанном сервере находятся как исходные тесты, так и исполняемые коды для Linux, но часто двоичные коды новой версии появляются на сервере гораздо позже исходных текстов.

В настоящее время средства поддержки Kerberos включаются в состав большинства дистрибутивных пакетов Linux. В частности, Debian 2.2 комплектуется eBones и Heimdal, в состав Mandrake 8.1 и Red Hat 7.2 входит Kerberos V5, а SuSE 7.3 комплектуется Heimdal. В поставку систем Caldera 3.1, Slackware 8.0 и TurboLinux 7.0 средства Kerberos не входят, но вы можете включить их, скомпилировав исходные коды либо установив пакет, предназначенный для другой версии Linux.

В последующих разделах данной главы описывается система Kerberos V5, разработанная MIT. Версия Kerberos V4 отличается от нее некоторыми особенностями конфигурации. Некоторые особенности работы систем, созданных на базе Kerberos V5, могут не совпадать с описанными здесь. Работая над данной главой, я использовал пакет Kerberos V5 в системе Red Hat, в тексте главы приводятся ссылки на исходные коды MIT.

Настройка сервера Kerberos

Основой сети Kerberos является сервер Kerberos, или KDC. Как и для большинства серверов Linux, для настройки KDC используется текстовый файл, находящийся в каталоге /etc. Зная формат этого файла, вы можете устанавливать конфигурацию сервера Kerberos и, следовательно, воздействовать на работу других сетевых систем. В данном разделе приводятся основные сведения о конфигурации сервера. В последующих разделах обсуждается конфигурация клиентов и серверов приложений.

Некоторые действия, рассматриваемые в данном разделе, приходится выполнять при настройке клиентов Kerberos и серверов приложений. Например, для всех систем задаются области Kerberos.

Очевидно, что пакет Kerberos должен быть установлен на компьютере, предназначенном для размещения KDC. Для компиляции исходных кодов MIT и установки полученных программ надо вызвать сценарий `configure`, содержащийся в составе пакета, а затем вызвать команды `make` и `make install`. В результате этих действий будут инсталлированы сервер Kerberos, серверы приложений и клиенты. При выполнении сценария `configure` желательно задавать опцию `--enable-shared`; при этом будут созданы разделяемые библиотеки Kerberos, что позволит уменьшить размеры программ. Такая конфигурация используется многими пакетами независимых производителей. Если система поставляется в виде двоичных кодов, компоненты Kerberos часто распределяются по отдельным пакетам. Например, дистрибутивная поставка Kerberos для системы Red Hat состоит из пакетов, которые называются `krb5-libs`, `krb5-server` и `krb5-workstation`. Такой подход позволяет исключить при инсталляции ненужные пакеты.

Редактирование конфигурационных файлов сервера

Основным конфигурационным файлом сервера Kerberos является файл /etc/krb5.conf. Этот файл состоит из нескольких разделов; роль заголовка раздела выполняет ключевое слово, помещенное в квадратные скобки. Строки, следующие до появления очередного заголовка, определяют характеристики, соответствующие текущему разделу. Пример файла krb5.conf для KDC приведен в листинге 6.1.

Листинг 6.1. Пример файла krb5.conf

```
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
 ticket_lifetime = 24000
 default_realm = THREEROOMCO.COM
 dns_lookup_realm = false
 dns_lookup_kdc = false

[realms]
 THREEROOMCO.COM = {
  kdc = kerberos.threeroomco.com:88
  kdc = kerberos-1.threeroomco.com:88
  kdc = kerberos-2.threeroomco.com:88
  admin_server = kerberos.threeroomco.com:749
  default_domain = threeroomco.com
 }

[domain_realm]
 .threeroomco.com = THREEROOMCO.COM
 threeroomco.com = THREEROOMCO.COM
 outsider.threeroomco.com = PANGAEA.EDU

[kdc]
   profile = /var/kerberos/krb5kdc/kdc.conf
```

Каждая строка внутри раздела состоит из имени переменной, за которой следуют знак равенства (символ "=") и значение этой переменной. Некоторые разделы могут содержать подразделы, для обозначения которых используются фигурные скобки. Например, в разделе [realms], представленном в листинге 6.1, фигурными скобками выделены строки, связанные с областью THREEROOMCO.COM. Наличие подразделов позволяет создавать файлы, поддерживающие несколько областей.

Большинство разделов, содержащихся в рассмотренном конфигурационном файле, используется для настройки серверов приложений и клиентов Kerberos. Не нужны лишь разделы `[login]` и `[kdc]`. Для некоторых программ могут потребоваться специальные параметры, которые обычно задаются в разделе `[appdefaults]`.

KDC также использует собственный конфигурационный файл `kdc.conf`. В этом файле содержатся данные, предназначенные только для KDC, в то время как информация в файле `krb5.conf` используется также клиентами и серверами приложений. Формат файла `kdc.conf` совпадает с форматом файла `krb5.conf`.

Определение области

Как правило, для определения областей в файлы krb5.conf и `kdc.conf` включаются специальные записи. Отредактировав файл, рассмотренный выше в качестве примера, вы измените конфигурацию KDC, серверов приложений и клиентов. Для того чтобы изменения были учтены сервером Kerberos и серверами приложений, надо перезапустить эти программы.

Редактирование файла krb5.conf

В файле `krb5.conf` находится раздел `[realms]`, в котором определены основной KDC и ведомые серверы. В разделе [domain_realm] указывается взаимосвязь между областью Kerberos и доменом Internet. Оба раздела содержатся в листинге 6.1.

В рассмотренном примере раздел `[realms]` определяет основной KDC и два ведомых KDC для области `THREEROOMCO.COM`. Традиционно эти серверы называются `kerberos` и `kerberos-n`, где n — это номер ведомого сервера в домене, соответствующем области. В данном примере домен и область Kerberos имеют одинаковые имена (отличающиеся только регистром символов), но в общем случае их имена не обязательно должны совпадать. В определении каждого из KDC указан номер порта, по которому устанавливаются соединения с соответствующим сервером. Запись `admin-server` соответствует компьютеру, который используется для администрирования области. Обычно это тот же компьютер, на котором установлен KDC, но для выполнения функций администрирования используется другой порт (как правило, порт 749). Запись `default_domain` определяет имя домена, связанное с принципалами. По умолчанию в качестве такого имени используется имя области Kerberos, преобразованное в символы нижнего регистра. Очевидно, что в данном примере запись `default_domain` не обязательна.

В одном файле `krb5.conf` можно указать несколько областей. Для того чтобы сделать это, надо включить в раздел [realms] имена нескольких областей, а параметры, описывающие каждую из них, поместить в фигурные скобки.

Записи, содержащиеся в разделе `[domain_realm]`, отображают имена компьютеров и доменов в области Kerberos. За именем узла или домена (имя домена начинается с точки) следует знак равенства, после него указывается область Kerberos, к которой относится компьютер или домен. Из листинга 6.1 видно, что все компьютеры, принадлежащие домену threeroomco.com (в том числе узел сети с именем `threeroomco.com`), попадают в область `THREEROOMCO.COM`. Исключение составляет компьютер `outsider.threeroomco.com`, который попадает в область `PANGAEA.EDU`.

СОВЕТ

При настройке DNS-сервера целесообразно создать записи CNAME, определив с их помощью имена узлов, указанные в krb5.conf и в других конфигурационных файлах. Это поможет вам быстро ввести в строй резервный KDC, расположенный на другом компьютере, не редактируя конфигурационные файлы. Кроме того, для обращения к KDC можно использовать виртуальный IP-адрес. В этом случае компьютер, поддерживающий протокол NAT (Network Address Translation — преобразование сетевых адресов), определяется на DNS-сервере как KDC, но при обращении к нему он перенаправляет запрос тому компьютеру, на котором размещается реальный KDC. Это также позволяет перемещать KDC с одного компьютера на другой, не изменяя записи DNS.

Редактирование файла kdc.conf

В файле kdc.conf содержатся те же записи, что и в файле krb5.conf. Пример содержимого файла kdc.conf приведен в листинге 6.2. Информация об областях Kerberos помещается в раздел [realms]. Редактируя файл kdc.conf, необходимо обратить внимание на имя области, так как во многих конфигурационных файлах по умолчанию используется имя области EXAMPLE.COM. В разделе [realms] также содержатся записи, в которых указываются типы ключей, поддерживаемых в данной области. Эти записи можно изменять только в том случае, если вы ясно представляете себе последствия таких изменений. В разделе [kdcdefaults] указаны дополнительные конфигурационные файлы.

Листинг 6.2. Пример файла kdc.conf

```
[kdcdefaults]
 acl_file = /var/kerberos/krb5kdc/kadm5.acl
 dict_file = /usr/share/dict/words
 admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab

[realms]
 THREEROOMCO.COM = {
  master_key_type = des-cbc-crc
  supported_enctypes = des-cbc-crc:normal des3-cbc-raw:normal \
des3-cbc-sha1:normal des-cbc-crc:v4 des-cbc-crc:afs3
  }
```

Создание основного ключа

Для контроля доступа к Kerberos используется *основной ключ* (master key), который по сути представляет собой пароль и хранится в *stash-файле*. Stash-файл — это специальный файл, который читает сервер при запуске и определяет, разрешено ли его выполнение. Если stash-файл отсутствует, основной ключ необходимо вручную вводить при запуске сервера.

Поскольку stash-файл содержит чрезвычайно важную информацию, он должен храниться на том же диске, что и KDC, и быть доступным для чтения только пользователю root. Создавать резервную копию этого файла можно лишь в том случае, если носитель информации будет храниться в надежном месте. При выборе основного клю-

ча следует руководствоваться теми же правилами, что и при выборе пароля: в качестве основного ключа нельзя использовать слово ни одного из существующих языков, его нельзя создавать на основе каких-либо данных, которые злоумышленник может узнать из официальных источников. В основном ключе должны присутствовать символы верхнего и нижнего регистра, цифры и знаки пунктуации. Ключ должен напоминать случайный набор символов и в то же время достаточно легко запоминаться. Основу ключа могут составлять начальные буквы слов некоторой фразы. Например, из фразы "yesterday I went to the dentist" формируется последовательность yiwttd. Изменяя случайным образом регистр символов и добавляя цифры и знаки пунктуации, можно получить набор знаков yi9Wt%Td, который вполне подходит для использования в качестве основного ключа.

Для создания основного ключа и записи его в stash-файл используется команда kdb5_util.

```
# kdb5_util create -r THREEROOMCO.COM -s
Initializing database '/var/kerberos/krb5kdc/principal' for
realm 'THREEROOMCO.COM',
master key name 'K/M@THREEROOMCO.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:
Re-enter KDC database master key to verify:
```

Из соображений безопасности символы, вводимые в ответ на приглашение, не отображаются на экране.

В процессе выполнения утилита kdb5_util создает и инициализирует несколько файлов. Эти файлы помещаются в каталог /var/kerberos/krb5kdc либо в другой каталог, используемый системой Kerberos, например в /usr/local/var/krb5kdc. Ниже приведен перечень файлов, создаваемых с помощью утилиты kdb5_util.

- Stash-файл с именем .k5 .*имя_ облает и* или .k5stash.
- Файлы principal иprincipal. ok, содержащие базу данных Kerberos. (В некоторых системах файл principal называется principal.db.)
- Файлыprincipal.kadm5 иprincipal.kadm5.lock, предназначенные для администрирования Kerberos.

Если вы по каким-то причинам не хотите создавать stash-файл, то, вызывая kdb5_util, не надо указывать опцию -s. В этом случае при каждом запуске сервера Kerberos придется задавать основной ключ.

Администрирование области

Отредактировав конфигурационные файлы Kerberos и вызвав kdb5_util для создания основного ключа и инициализации базы данных Kerberos, можно приступать к администрированию области. Этот процесс в основном сводится к определению принципалов. Для выполнения действий по добавлению принципалов необходимо обладать полномочиями администратора.

Таблица 6.1. Коды полномочий в файле ACL

Код	Описание
a	Позволяет добавлять принципалов или политики
A	Запрещает добавлять принципалов или политики
d	Позволяет удалять принципалов или политики
D	Запрещает удалять принципалов или политики
m	Позволяет модифицировать принципалов или политики
M	Запрещает модифицировать принципалов или политики
c	Позволяет изменять пароли принципалов
C	Запрещает изменять пароли принципалов
i	Позволяет передавать запросы базе данных
I	Запрещает передавать запросы базе данных
l	Позволяет выводить списки принципалов или политик из базы данных
L	Запрещает выводить списки принципалов или политик из базы данных
x или *	Признак групповой операции

Определение базовых ACL

Информация о принципалах Kerberos хранится в формате ACL (Access Control Lists — списки контроля доступа) в файле, имя которого определяет запись `acl_file` в составе `kdc.conf`. Этот файл содержит строки, представленные в следующем формате:

Принципал Kerberos *Полномочия* *Целевой принципал*

Несмотря на то что в ACL Kerberos и ACL для файловой системы представляют механизмы контроля доступа, они несколько отличаются друг от друга. ACL файловой системы определяют, кто имеет право доступа к файлам. ACL Kerberos предоставляют права модификации базы данных Kerberos. ACL Kerberos не зависят от средств поддержки ACL файловой системы.

Первое поле (Принципал Kerberos) содержит идентификатор принципала (правила формирования идентификатора принципала были рассмотрены ранее). Любой компонент идентификатора можно заменить символом "*". Например, имя `*/admin@THREEROOMCO.COM` соответствует любой основе для экземпляра `admin` и области `THREEROOMCO.COM`. Подобное определение позволяет предоставить доступ к KDC всем администраторам.

Второе поле (Полномочия) — это код ACL, соответствующего принципалу. Типы полномочий задаются с помощью односимвольных кодов. Назначение символов описано в табл. 6.1. Объединяя разные коды, можно задать различные типы доступа. Например, код `ali` означает, что принципал может добавлять пользователей, выводить списки принципалов и передавать запросы базе данных.

Последнее поле (Целевой принципал) может отсутствовать. Оно определяет имена принципалов, к которыми применяются заданные полномочия. Например, вы можете ограничить возможности пользователя по доступу и модификации прав конкретных принципалов. Как и при определении принципала Kerberos, в идентификаторе целевого принципала можно использовать символ "*".

Рассмотрим в качестве примера следующую запись:

```
*/admin@THREEROOMCO.COM *
```

Эта запись предоставляет всем принципалам экземпляра admin полный доступ к базе данных Kerberos. Подобная запись включается в файл по умолчанию. Первое, что надо сделать, — модифицировать запись так, чтобы она соответствовала нужной вам области.

Создание принципалов

Для администрирования базы пользователей Kerberos применяются программы kadmin и `kadmin.local`. Программа kadmin позволяет администрировать KDC с удаленного компьютера; она организует обмен шифрованными сообщениями. Программа `kadmin.local` дает возможность модифицировать базу данных без применения сетевых средств. Вначале необходимо с помощью kadmin `.local` создать хотя бы одного пользователя, обладающего правами администратора, затем можно использовать kadmin для работы с удаленного узла. Очевидно, что удаленное администрирование можно осуществлять только в том случае, если на узле KDC присутствуют специализированные серверы Kerberos, предназначенные для выполнения подобных задач.

При запуске программ kadmin и `kadmin.local` можно задавать различные параметры, определяющие имя принципала, область, администрирование которой будет выполняться, и т. д. Подробную информацию о поддерживаемых параметрах можно получить, обратившись к соответствующим разделам справочной информации. После запуска программы надо ввести команды и данные, которые она запрашивает. Например, чтобы добавить принципала `admin/admin@THREEROOMCO.COM`, необходимо задать команду addprinc.

```
# kadmin.local
Authenticating as principal root/admin@THREEROOMCO.COM with
password.
kadmin.local:  addprinc admin/admin@THREEROOMCO.COM
WARNING: no policy specified for admin/admin@THREEROOMCO.COM;
defaulting to no policy
Enter password for principal "admin/admin@THREEROOMCO.COM":
Re-enter password for principal "admin/admin@THREEROOMCO.COM":
Principal "admin/admin@THREEROOMCO.COM" created.
```

Как обычно, при вводе пароля символы не отображаются на экране. Не следует использовать в качестве административного пароля основной ключ.

После создания принципала, предназначенного для администрирования, вам надо сформировать для него *ярлык* (keytab). Ярлык — это ключ, который Kerberos использует для расшифровки административных билетов. Вам нет необходимости задавать этот ключ; Kerberos сгенерирует его самостоятельно. Достаточно лишь указать системе на необходимость выполнения этого действия, для чего следует в среде `kadmin.local` вызвать команду ktadd.

```
kadmin.local:  ktadd -k /var/kerberos/krb5kdc/kadm5.keytab \
kadmin/admin kadmin/changepw
```

Для указания файла, в котором должен храниться ярлык, используется опция -k. Имя файла должно соответствовать имени, указанному посредством записи `admin_keytab` в файле `kdc.conf`. После указания значения опции -k задаются принципалы, для кото-

рых создается ярлык, в данном примере это kadmin/admin и kadmin/changepw (эти два принципала являются стандартными компонентами Kerberos; вам нет необходимости создавать их).

В дополнение к принципалу, который соответствует администратору, вы должны создать принципалов для ваших пользователей, серверов администрирования и KDC. Для этого используется описанная выше команда addprinc. Предположим, например, что вам надо добавить принципала fluffy@THREEROOMCO.COM. Для этого вы должны ввести следующую команду:

```
kadmin.local:  addprinc fluffy@THREEROOMCO.COM
```

Для серверов желательно использовать опцию -randkey, которая указывает системе на то, что для этого принципала ключ должен быть сформирован по случайному закону. Если вы не зададите эту опцию, то программа предложит вам ввести пароль. Другие опции, которые могут быть применены в данной ситуации, описаны на страницах справочной системы, посвященных kadmin.

Принципалы для серверов приложений создаются аналогичным образом, но идентификаторы обычно имеют вид *имя_сервера/имя_узла@имя_области* (именем сервера может быть, например, pop или ftp). Необходимо также создать принципала, в идентификаторе которого в качестве основы вместо имени сервера будет указано слово host. Кроме того, надо создать ярлык принципала host, используя для этого команду ktadd. Каждый ярлык должен быть помещен в отдельный файл, т. е. для разных узлов необходимо задавать различные значения опции -k. Впоследствии этот файл следует переместить на узел, на котором выполняется сервер приложения. Можно поступить и по-другому: вызвать программу kadmin с компьютера, на котором расположен сервер приложения, создать принципала для сервера и использовать команду ktadd для того, чтобы поместить ярлык в файл на этом компьютере.

Вероятнее всего, вы решите создать принципала для каждого KDC. Идентификаторы таких принципалов создаются в формате host/*имя_узла@имя_области*, например host/kerberos-1.threeroomco.com/THREEROOMCO.COM. Для ведомых KDC создавать принципалы не обязательно, но они могут быть полезны, если на соответствующем компьютере будет разрешена регистрация обычных пользователей (что категорически не рекомендуется) или если вы захотите использовать ведомый KDC вместо ведущего. Ведущему KDC этот принципал понадобится для того, чтобы передать базу данных ведомому KDC.

Окончив работу с программой kadmin, надо завершить ее выполнение путем ввода команды quit.

Запуск KDC

К этому моменту компоненты Kerberos настроены и могут быть запущены. Для запуска сервером можно использовать способы, описанные в главе 4. Если пакет Kerberos поставлялся вместе с вашей системой, для него, вероятно, существует сценарий запуска SysV. Сценарий для KDC обычно называется krb5kdc, а сценарий для сервера администрирования — kadmin.

Если сценарии SysV отсутствуют, вы можете использовать для запуска KDC программы krb5kdc и kadmin, которые входят в состав пакета Kerberos. Каждая программа порождает процесс из оболочки, поэтому вам нет необходимости указывать после ее

имени символ "&". Вызывать данные программы можно из локального сценария запуска (/etc/rc.d/rc.local).

Настройка ведомого KDC

Ведомый KDC настраивается практически так же, как и ведущий. Вам надо отредактировать файлы krb5.conf и kdc.conf, использовать kdb5_util для создания файлов базы данных, сформировать файл ACL и вызвать команду ktadd программы kadmin.local, чтобы записать ярлык в файл.

Каждому KDC требуется файл, в котором перечислены все KDC (или, точнее, принципалы, связанные со всеми KDC). Этот файл необходим для передачи данных из базы. Указанный файл имеет имя kpropd.acl и чаще всего хранится в каталоге /var/kerberos/krb5kdc либо /usr/local/var/krb5kdc. Содержимое этого файла выглядит приблизительно следующим образом:

```
host/kerberos.threeroomco.com@THREEROOMCO.COM
host/kerberos-1.threeroomco.com@THREEROOMCO.COM
```

Сформировав данный файл для каждого из KDC, надо сконфигурировать ведомый KDC для выполнения двух серверов: kpropd и klogind. Запуск этих серверов можно осуществлять с помощью суперсервера. Соответствующие записи /etc/inetd.conf могут иметь следующий вид:

```
krb5_prop stream tcp nowait root /usr/kerberos/sbin/kpropd
kpropd eklogin stream tcp nowait root \
/usr/kerberos/sbin/klogind klogind -k -c -e
```

Возможно, на вашем компьютере расположение файлов будет отличаться от указанного выше. Если же в вашей системе используется суперсервер xinetd, вам придется внести изменение в его конфигурационный файл (о настройке суперсерверов см. в главе 4). Если в файле /etc/services отсутствуют записи для krb5_prop и eklogin, вам надо добавить в файл следующие строки:

```
krb5_prop    754/tcp          # Передача данных ведомому
                              # серверу Kerberos
eklogin      2105/tcp        # Удаленная регистрация
                              # с использованием шифрования
```

Распространение данных осуществляется ведущим KDC и состоит из двух этапов: извлечение содержимого базы данных и передача его ведомым серверам. Сценарий, выполняющий обе задачи, представлен в листинге 6.3. Возможно, вам придется внести в него изменения, отражающие особенности вашей системы и структуру сети. Если в сети существует несколько ведомых серверов, вам надо вызвать kprop для каждого из них. Запуск данного сценария через определенные промежутки времени планируется с помощью инструмента cron.

Листинг 6.3. Пример сценария, предназначенного для передачи базы данных Kerberos ведомым KDC

```
#!/bin/sh
/usr/kerberos/sbin/kdb5_util dump
/usr/kerberos/var/krb5kdc/slave_datatrans
/usr/kerberos/sbin/kprop -f
/usr/kerberos/var/krb5kdc/slave_datatrans \
    kerberos-1.mi1.threeroomco.com
```

Настройка сервера приложений Kerberos

Настройка KDC — важный этап подготовки системы Kerberos к работе, но сам по себе KDC не осуществляет полезных действий. Для того чтобы система стала работоспособной, необходимо выполнить вторую часть работы — настроить серверы приложений и клиенты Kerberos для взаимодействия с KDC. Данный раздел посвящен настройке серверов приложений, а настройка клиентов будет рассматриваться в следующем разделе.

В некоторых случаях один и тот же компьютер может выступать в роли сервера приложений и клиента. Так происходит, например, если Kerberos используется в качестве протокола аутентификации при регистрации. В этом случае компьютер должен быть настроен и как сервер приложений, и как клиент.

Выбор конфигурации сервера приложения

При настройке серверов приложений совершаются многие из тех действий, которые выполнялись при выборе конфигурации KDC. В частности, настраивая сервер приложений, необходимо изменить содержимое разделов [realms] и [domain_realm] файла krb5.conf так, чтобы находящиеся в них данные отражали конфигурацию областей. Кроме того, для работы сервера приложений нужен файл, содержащий ярлык. В этом файле должны находиться данные для узла (host/*имя_узла*@*имя_области*)и для каждого из кербериэованных серверов, которые выполняются на компьютере (например, если вы собираетесь использовать кербериэованный сервер Telnet, идентификатор принципала будет выглядеть так: telnet/*имя_узла*@*имя_области*). Файл, содержащий ярлык, можно создать с помощью программы kadmin.local на том компьютере, на котором выполняется KDC. Чтобы сделать это, вам придется добавить принципалов, вызвав команду addprinc, а затем записать ключи для этих принципалов в файл. Сеанс работы с программой kadmin.local может выглядеть так:

```
kadmin.local:  addprinc \
host/gingko.threeroomco.com@THREEROOMCO.COM
kadmin.local:  addprinc \
telnet/gingko.threeroom.com@THREEROOMCO.COM
kadmin.local:  ktadd -k gingko.keytab \
host/gingko.threeroomco.com telnet/gingko.threeroomco.com
```

Файл gingko.keytab, полученный в результате описанных действий, надо переместить в каталог /etc компьютера, на котором выполняется сервер приложений, и присвоить ему имя krb5.keytab. Так как файл содержит чрезвычайно важную информацию,

для его копирования надо применять средства, исключающие утечку данных, например перенести файл на дискету или использовать scp. Записав файл по месту назначения, надо установить права, позволяющие обращаться к нему только пользователю root, и удалить файл с компьютера KDC. Данный файл можно непосредственно создать на том компьютере, на котором установлен сервер приложений. В этом случае при вызове команды ktadd не надо указывать опцию -k gingko.keytab; система самостоятельно разместит файл в нужном каталоге. Данный метод пригоден, только если средства администрирования установлены и корректно сконфигурированы; кроме того, необходимо, чтобы была правильно выбрана базовая конфигурация Kerberos.

Запуск керберизованных серверов

Как правило, в состав стандартного пакета Kerberos входят керберизованные серверы и локальные средства аутентификации, например, программы, поддерживающие протоколы Telnet и FTP, а также разновидности shell, exec и login. Керберизованные программы надо установить вместо их традиционных аналогов. Так, если в вашей системе используется inetd, вам надо включить в файл /etc/inetd.conf следующие данные:

```
klogin stream tcp nowait root /usr/kerberos/sbin/klogind \
klogind -k -c
eklogin stream tcp nowait root /usr/kerberos/sbin/klogind \
klogind -k -c -e
kshell stream tcp nowait root /usr/kerberos/sbin/kshd \
kshd -k -c -A
ftp stream tcp nowait root /usr/kerberos/sbin/ftpd \
ftpd -a
telnet stream tcp nowait root /usr/kerberos/sbin/telnetd \
telnetd -a valid
```

Приведенные выше строки заменяют соответствующие записи в составе файла.

Помимо программ, которые распространяются в составе пакета Kerberos, доступны также керберизованные варианты других серверов. Подобные серверы поставляются независимыми производителями. Прежде чем использовать такие продукты, необходимо тщательно изучить документацию и выяснить их возможности и особенности выполнения.

Настройка клиентов Kerberos

Для того чтобы пользователь мог работать с системой Kerberos, надо в первую очередь создать принципала для этого пользователя. Принципалы пользователей имеют вид *имя_пользователя*@*имя_области*, и для их создания применяются утилиты kadmin и kadmin.local. После создания принципала пользователь может обращаться к серверам с помощью клиентов, ориентированных на работу в системе Kerberos. Вам, как системному администратору, необходимо установить соответствующие клиентские программы. Пользователям также потребуются утилиты, предназначенные для получения TGT и управления ими. Если вы захотите, чтобы средства Kerberos использовались для управления регистрацией на отдельных рабочих станциях, вам надо модифицировать обычные инструменты регистрации.

Обеспечение доступа к серверам Kerberos

На первый взгляд может показаться, что для обращения к серверам приложений Kerberos достаточно иметь соответствующие клиентские программы. На самом деле ситуация выглядит несколько сложнее. Прежде всего, в процессе обмена участвуют специальные утилиты: пользователь должен иметь возможность получить TGT и при необходимости изменить пароль Kerberos. Кроме того, к самим клиент-программам Kerberos предъявляются специальные требования; без поддержки специфических возможностей клиенты Kerberos превращаются в обычные клиентские программы, не обеспечивающие повышенный уровень защиты.

Использование сетевых утилит Kerberos

В состав пакета Kerberos входят утилиты, предназначенные для управления паролями и билетами Kerberos. Наиболее важные из них перечислены ниже.

- `kinit`. Эта программа получает TGT. Ее действие можно представить себе как "регистрацию" пользователя в области Kerberos; до запуска `kinit` воспользоваться клиентами Kerberos невозможно. При выполнении программа `kinit` обращается к KDC для получения TGT. Как было сказано в начале данной главы, TGT используется керберизованными приложениями для получения разрешения на пользование серверами приложений Kerberos. По умолчанию в качестве основы идентификатора принципала применяется имя пользователя; в этом же идентификаторе указывается имя области, принятое по умолчанию. Если вам необходимо, чтобы использовался другой идентификатор принципала, его надо указать при вызове программы. Соответствующая команда может иметь вид `kinit minerva@PANGAEA.EDU`. Программа `kinit` также поддерживает и другие опции. Подробную информацию о них можно получить, обратившись к страницам справочной системы, посвященным `kinit`.

- `klist`. Данная программа выводит список билетов (в том числе TGT), выданных на текущий момент, а также информацию о сроке их действия. Если программа `kinit` еще не запущена, `klist` не выведет сведений ни об одном билете.

- kpasswd. Программа kpasswd не имеет отношения к поддержке конкретных сеансов Kerberos; она позволяет изменять пароль принципала в базе данных Kerberos. Эта программа по сути представляет собой аналог `passwd`. Для использования kpasswd необходимо иметь TGT.

- `kdestroy`. Данная программа удаляет все билеты из кэша пользователя. Обычно она вызывается перед окончанием сеанса работы в системе или в том случае, если пользователь твердо знает, что в ближайшем будущем он не будет работать с серверами Kerberos. Несмотря на то что билеты автоматически удаляются по истечении срока их действия, рекомендуется после окончания работы с билетами вызывать программу `kdestroy`. Это снизит вероятность того, что злоумышленник сможет обнаружить недостатки в системе защиты и использовать их в своих целях.

Желательно включать вызов `kdestroy` в состав `.logout`, `.xinitrc` и других стандартных файлов, которые выполняются по окончании сеанса работы пользователя в системе. При использовании модулей PAM (они будут рассматриваться далее в этой главе) `kdestroy` вызывается автоматически.

Как используются перечисленные выше утилиты на практике? В большинстве случаев программы **kinit** и **kdestroy** вызываются соответственно в начале и в конце сеанса работы. Чтобы определить текущее состояние билетов, можно использовать утилиту **klist**. Рассмотрим следующий пример:

```
$ kinit
Password for fluffy@THREEROOMCO.COM:
$ klist
Ticket cache: FILE:/tmp/krb5cc_500
Default principal: fluffy@THREEROOMCO.COM

Valid starting     Expires            Service principal
10/09/02 14:38:57  10/10/02 00:38:57  krbtgt/THREEROOMCO.COM@\
                                      THREEROOMCO.COM


Kerberos 4 ticket cache: /tmp/tkt500
klist: You have no tickets cached
$ kpasswd
Password for fluffy@THREEROOMCO.COM:
Enter new password:
Enter it again:
Password changed.
$ kdestroy
$ klist
klist: No credentials cache file found (ticket cache
FILE:/tmp/krb5cc_500)


Kerberos 4 ticket cache: /tmp/tkt500
klist: You have no tickets cached
```

Программа **kinit**, вызываемая в начале работы, получает TGT, который затем отображается при вызове утилиты **klist** как krbtgt. Программа **klist** выводит также время выдачи билета и окончания его действия; в данном случае билет действителен в течение десяти часов (в зависимости от системы это значение может изменяться). Если вызвать **klist** после использования керберизованного клиента, данная утилита выведет сведения о другом билете. Для изменения пароля надо сначала указать текущий пароль, а затем дважды ввести новый. Как и при работе с **kinit**, символы, составляющие пароль, не отображаются на экране. После вызова **kdestroy** все билеты удаляются.

Использование керберизованных клиентов

Клиент-программы, ориентированные на работу с Kerberos, выполняются так же, как и их аналоги, не обеспечивающие дополнительную защиту. В большинстве случаев для вызова клиент-программы надо ввести ее имя и имя сервера, с которым необходимо установить соединение. Иногда для использования средств Kerberos приходится задавать некоторые параметры. Например, чтобы избавить пользователя от необходимости указы-

вать имя и пароль при работе с клиентом **telnet**, этой программе надо предоставить дополнительные данные. Особенности вызова керберизованных клиентов описаны ниже.

- **telnet**. Программа **telnet**, ориентированная на работу с Kerberos, отличается от стандартной программы **telnet** лишь некоторыми опциями. Если вы вызовете программу с помощью команды **telnet** *удаленный_узел*, вам придется ввести пользовательское имя и пароль. Чтобы воспользоваться преимуществами однократной регистрации, вы должны задать опции -a (автоматическая регистрация) и -f (перенаправление билетов).

- **rlogin**. Стандартная программа **rlogin** может создавать серьезную угрозу безопасности системы (этот вопрос будет подробно рассмотрен в главе 13). Работая с керберизованной версией этой программы, можно воспользоваться средствами аутентификации Kerberos, для чего необходимо задать опцию -f. Результат получится приблизительно такой же, как при вызове **telnet** с опциями -a и **-f**.

- **ftp**. При вызове без дополнительных опций данная программа использует средства аутентификации Kerberos, но пользователю приходится подтверждать регистрационное имя (имя пользователя отображает программа, для подтверждения достаточно нажать клавишу <Enter>).

Программа **ftp**, поставляемая в составе пакета Kerberos, в процессе установления соединения отображает более подробную информацию, чем другие клиенты Kerberos. Если у вас возникли проблемы с установкой конфигурации Kerberos, эта информация поможет разрешить их.

- rsh. Данная утилита позволяет запускать на другом компьютере программы, выполняющиеся в текстовом режиме, без использования **telnet**, **rlogin** и других подобных средств. Как и при работе с прочими керберизованными инструментами, при вызове данной программы надо задавать опцию **-f**.

- **rcp**. Стандартная программа **rcp** предназначена для передачи файлов; керберизованная версия данного инструмента поддерживает средства аутентификации Kerberos.

- Прочие программы. Здесь рассмотрены лишь описания сетевых инструментов, поставляемых в составе стандартного пакета Kerberos V5. В настоящее время доступны другие керберизованные системы. Очевидно, что для работы с ними необходимо, чтобы средства Kerberos поддерживались как клиентами, так и серверами.

В справочной системе приведена дополнительная информация о керберизованных клиентских программах, в том числе сведения о различных опциях, которые задаются при их вызове. В особенности внимательно надо прочитать документацию на программы, не входящие в стандартный пакет поставки Kerberos. Некоторые из них обеспечивают минимальное взаимодействие с Kerberos (например, поддерживают аутентификацию, но не осуществляют кодирование), другие реализуют полный набор функций защиты. Говоря о средствах Kerberos, следует особо отметить кодирование данных, которое должно поддерживаться всеми стандартными клиентами Kerberos. Если при вызове такого клиента вы укажете в командной строке опцию -x, данные будут передаваться в зашифрованном

виде. Эта возможность чрезвычайно полезна в том случае, если необходимо передавать важные данные по Internet или даже по локальной сети. Например, если вы собираетесь зарегистрироваться на компьютере с помощью средств Telnet, а затем использовать команду `su` для выполнения административных функций, то, вероятно, захотите передавать данные в зашифрованном виде и защитить таким образом пароль пользователя root. Это позволит сделать команда `ksu`, которая будет рассмотрена ниже в этой главе.

При установке системы Kerberos, поставляемой в виде исходных кодов, пользовательские программы по умолчанию размещаются в каталоге `/usr/local/bin` (инструменты, предназначенные для администрирования, располагаются в `/usr/local/sbin`). Некоторые пакеты, распространяемые в виде двоичных кодов, используют другие каталоги. Например, Kerberos для Red Hat помещает пользовательские программы в каталог /usr/kerberos/bin. Для того чтобы использовать керберизованные инструменты вместо стандартных программ, надо включить каталоги, в которых размещены эти инструменты, в состав строки, задаваемой в качестве значения переменной окружения PATH, и убедиться, что они находятся перед каталогами со стандартными программами. (Значение переменной окружения PATH обычно указывается в файле `/etc/profile`, а для пользователей, работающих с оболочкой Bash, — в файле `.bashrc`.

Применение Kerberos для регистрации пользователей

При обсуждении средств поддержки сеанса Kerberos предполагалось, что пользователь уже зарегистрирован в системе. Однако подобный подход крайне неудобен, так как пользователю приходится регистрироваться дважды: один раз в начале работы на компьютере, а второй раз при работе с серверами Kerberos (при запуске программы `kinit`). Решением этой проблемы могло бы быть использование программы, одновременно решающей обе задачи. В составе пакета Kerberos обычно поставляются два подобных инструмента: `login.krb5` и `ksu`. Они предназначены для регистрации пользователя в текстовом режиме. Регистрацию можно было бы организовать и по-другому, модифицировав библиотеки Linux для работы с Kerberos. Такое решение сложнее в реализации, но обеспечивают большую гибкость.

ВНИМАНИЕ Рекомендуется перед использованием `login.krb5` сначала проверить программу `kinit` для основных учетных записей, в том числе для записи root. Если `kinit` не работает, не будет работать и `login.krb5`, следовательно, вам не удастся зарегистрироваться с консоли в текстовом режиме. Желательно также зарегистрироваться под именем root с одного из виртуальных терминалов. В случае, если у вас возникнут проблему с использованием `login.krb5`, вы сможете изменить конфигурацию программы. Аналогичному подходу необходимо следовать и при работе с другими инструментами регистрации.

Выполнение аутентификации Kerberos в текстовом режиме

Процедура регистрации в системе Linux в текстовом режиме включает использование программы getty или одной из ее разновидностей (разновидностями getty являются mingetty, mgetty и vgetty). Они запускаются из `/etc/inittab`, контролируют консольный терминал и последовательные порты и передают управление программе /bin/login. Программы поддержки некоторых сетевых протоколов, например Telnet, также вызывают /bin/login. Как следует из имени `login.krb5`, эта программа со-

здана для замены /bin/login. Прежде чем выполнять такую замену, желательно сохранить исходную программу регистрации под другим именем. Например, вы можете использовать следующие команды:

```
# mv /bin/login /bin/login-original
# cp /usr/kerberos/sbin/login.krb5 /bin/login
```

Если возникнут проблемы с использованием login.krb5, вы всегда сможете восстановить исходную программу /bin/login. После замены программы login регистрация пользователя на компьютере будет автоматически сопровождаться начальной регистрацией в системе Kerberos. Процедура начальной регистрации включает получение TGT, поэтому после нее нет необходимости в вызове kinit. Несмотря на то что описанная конфигурация предполагает наличие записи в файле /etc/passwd, рабочего каталога пользователя и прочих ресурсов, необходимых в обычных условиях для нормальной работы на компьютере, в системе будет выполняться только аутентификация Kerberos. Существуют также другие средства регистрации, которые надо модифицировать для работы с Kerberos. К ним относятся регистрация с помощью инструментов с графическим интерфейсом, а также регистрация посредством серверов, которые не используют /bin/login, например SSH.

Переход к новой учетной записи после регистрации

При использовании su для замены учетной записи в работу включается новый механизм аутентификации. В состав пакета Kerberos входит альтернативный инструмент ksu, который принимает решение о переходе к учетной записи другого пользователя и выполняет необходимые действия. Для работы этой программы должны выполняться следующие условия.

- Компьютер, на котором работает ksu, должен иметь ярлык (обычно он хранится в файле /etc/krb5 . keytab).
- Для исполняемого файла ksu должен быть установлен признак SUID, так, чтобы программа, запускаемая от имени любого пользователя, выполнялась с правами root. Во многих пакетах Kerberos этот признак не установлен, поэтому вам необходимо сделать это самостоятельно (вызвать команду chmod a+s /usr/kerberos/bin/ksu).
- Для повышения уровня защиты пользователю, учетной записью которого вы собираетесь воспользоваться, должен соответствовать файл авторизации, в котором указываются права доступа для других пользователей. В роли файла авторизации может выступать .k5login или .k5users. Особенности создания и использования этих файлов описаны ниже.

Для того чтобы один пользователь мог перейти к учетной записи другого пользователя, целевой пользователь должен создать файл авторизации. Без этого файла ksu запросит пароль, который может быть передан по сети в незашифрованном виде (это произойдет, если пользователь регистрировался посредством незащищенного протокола, например Telnet). Файл .k5login предоставляет другому пользователю полный набор привилегий. Он состоит из набора строк, в каждой из которых указан принципал Kerberos. Файл .k5users предоставляет пользователю ограниченный доступ; в нем указаны списки программ, которые этот пользователь может запускать. Каждая строка файла начинается

с идентификатора принципала Kerberos, за которым следуют имена программ, разделенных пробелами. Групповые операции обозначаются с помощью символа *. Ниже приведен пример записи, с помощью которой принципалу minerva@THREEROOMCO.COM предоставляются права на запуск программ /bin/Is и /usr/bin/zip.

```
minerva@THREEROOMCO.COM /bin/Is /usr/bin/zip
```

После настройки программа ksu работает подобно su — вы вводите имя программы, затем указываете имя пользователя, привилегии которого вы собираетесь получить. Если файлы .k5login и .k5users отсутствуют, вам придется ввести пароль для принципала. При наличии файла авторизации вводить пароль не нужно. Такой подход создает меньшую угрозу для безопасности системы, чем взаимодействие по незащищённому протоколу.

Если вы хотите непосредственно выполнить некоторую программу, вы можете сделать это с помощью опции -e *имя_программы*.Например, для того, чтобы запустить /bin/ls от имени пользователя fluffy,вам надо вызвать команду fluffy -e /bin/Is.

Использование PAM

Замена программ login и su специальными инструментами, ориентированными на работу с Kerberos, помогает решать задачи аутентификации, но существуют ситуации, в которых подобный подход не приносит желаемых результатов. Особенный интерес вызывают случаи, когда возникает необходимость контролировать процесс регистрации на рабочей станции с помощью программы с графическим интерфейсом и выполнять аутентификацию в системе Kerberos. Существуют также другие локальные средства, выполняющие аутентификацию пользователей, которые необходимо связать с Kerberos; в качестве примеров таких средств можно привести vlock и xscreensaver (эти инструменты блокируют сеанс взаимодействия, осуществляемый как в текстовом, так и в графическом режиме, до тех пор, пока пользователь не введет пароль). Существуют универсальные инструменты связывания различных программы со средствами Kerberos, но на сегодняшний день эти инструменты нельзя назвать широко распространенными, и они не поставляются в комплекте с Kerberos. Средства для решения данной задачи базируются на поддержке модулей PAM (Pluggable Authentication Module — встраиваемый модуль аутентификации) в системе Linux.

PAM выполняет роль посредника между программами, которым требуется аутентификация (например, сервером FTP, программой login, инструментами регистрации, работающими в среде X Window), и базами данных, в которых хранится информация о пользователях, в частности пароли (/etc/passwd, /etc/shadow и другие файлы стандартного пакета Linux). Процедуры аутентификации оформляются в виде отдельной библиотеки. В этом случае файлы, применяемые для аутентификации, могут быть без труда модифицированы, а программы, использующие их, остаются без изменений. Для поддержки нового формата файлов модифицируются только PAM. В этом смысле реализация средств поддержки Kerberos в виде PAM является почти идеальным решением, позволяющим обеспечить совместную работу с Kerberos многих приложений. Любые изменения не затрагивают прикладные программы, которые взаимодействуют только с PAM.

Поддержка Kerberos с помощью PAM имеет свои ограничения. В частности, если код программы построен так, что программа запрашивает имя пользователя и пароль, ее поведение не изменится при замене PAM на модуль, поддерживающий Kerberos. Полученные данные программа передаст PAM, и он предпримет попытку аутентификации с использованием базы данных Kerberos. Так, например, PAM не устраняет необходимость ввода имени пользователя и пароля при работе с FTP-сервером, но позволяет следить за текущим паролем и приводить его в соответствие с данными для области Kerberos. Если речь идет о кербериэованных программах, выполняющих регистрацию пользователя, то такая особенность не является недостатком, так как программы регистрации в любом случае запрашивают пользовательское имя и пароль.

Несмотря на то что PAM может выступать в роли универсального средства аутентификации и поддерживается во всех версиях Linux и во многих системах, отличных от Linux, кербериэованные версии PAM мало распространены. Ниже описаны некоторые из таких модулей, которые были доступны в момент написания данной книги.

- **Модуль Деррика Брешера (Derrik** Brashier). Этот модуль предназначен для использования совместно с Kerberos V4. Он расположен по адресу `ftp://ftp.dementia.org/pub/pam/`; имена файлов, содержащих этот модуль, начинаются символами pam_krb4. Выберите самый новый из файлов (во время написания данной книги все файлы датировались 1998 г.). Данный модуль распространяется в исходных кодах, поэтому перед использованием его надо скомпилировать.

- **Модуль Фрэнка Кусека** (Frank Cusack). Модуль PAM, поддерживающий MIT Kerberos V5 и Heimdal, находится по адресу `http://www.nectar.com/zope/krb/`. Этот пакет доступен в исходных кодах; изначально коды были написаны для Solaris, но после компиляции они будут работать в системе Linux.

- **Модуль Кертиса Кинга** (Curtis King). Этот модуль доступен по адресу `ftp://ftp.dementia.org/pub/pam/`; имя файла — `pam_krb5-1.1.3.tar.gz`. Данный модуль также требует компиляции, в ходе которой могут возникнуть проблемы.

- **Модуль для системы Red Hat.** Модуль PAM Kerberos V5 входит в состав дистрибутивного комплекта Red Hat под именем pam_krb5. Данный вариант PAM поставляется в виде двоичного кода в формате RPM, поэтому чрезвычайно просто устанавливается в Red Hat и других подобных системах. При подготовке материала данной главы я использовал именно этот тип модуля, хотя следует отметить, что точно так же работает модуль Фрэнка Кусека.

- **Модули для системы Debian.** В Debian и других подобных системах работа с Kerberos V5 и Heimdal поддерживается соответственно модулями `libpam-krb5` и `libpam-heimdal`. На Web-узле Debian эти пакеты найти достаточно сложно, поэтому лучше скопировать их, обратившись по адресам `http://ftp.nl.debian.org/debian/pool/non-US/main/libp/libpam-krb5/` и `http://ftp.nl.debian.org/debian/pool/non-US/main/libp/libpam-heimdal/`.

При инсталляции PAM для поддержки Kerberos вы по сути устанавливаете средства, с помощью которых PAM настраивается для выполнения конкретной задачи. В состав

модуля PAM входит одна или несколько библиотек, которые располагаются в каталогах /lib/security или /usr/lib/security. В системе Red Hat библиотеки содержатся в файлах pam_krb5.so и pam_krb5afs.so. Для работы с этими библиотеками вы должны внести изменения в конфигурационные файлы PAM, которые содержатся в каталоге /etc/pam.d. Имена конфигурационных файлов составляются на основании имен серверов и других программ, для которых необходимо выполнить аутентификацию. Например, содержимое файла /etc/pam.d/login определяет взаимодействие программы login с PAM. При редактировании конфигурационного файла PAM в нем надо изменить (или добавить) одну или несколько строк, определяющих использование нового модуля Kerberos. В пакете, предназначенном для системы Red Hat, содержится большое число файлов с примерами настройки. Эти файлы находятся в каталоге /usr/share/doc/pam_krb5-версия/pam.d, где версия — это номер версии пакета. Чтобы упростить настройку, надо скопировать соответствующие конфигурационные файлы в каталог /etc/pam.d. Файлы, которые могут потребоваться вам, перечислены ниже.

- login. Данный файл управляет взаимодействием с программой login. Настроив PAM для работы с Kerberos, вы можете отказаться от login.krb5 и продолжать работу с привычной вам программой login.
- gdm. GNOME Display Manager, или GDM, является одним из трех широко распространенных средств регистрации с графическим интерфейсом. (О настройке GDM и других подобных инструментах речь пойдет в главе 14.)
- xdm. Вторым инструментом, предоставляющим графический интерфейс для регистрации, является X Display Manager, или XDM. Конфигурационные файлы XDM может также использовать KDE Display Manager. В описании утверждается, что данный файл должен обеспечить аутентификацию Kerberos при работе указанных средств регистрации, однако при установке конфигурации в системе Mandrake возникают проблемы.
- su и sudo. Программа su, которая обсуждалась ранее, позволяет пользователю после регистрации в системе переходить к другой учетной записи. Программа ksu делает то же самое, используя аутентификацию Kerberos, однако аналогичные результаты можно получить, создав файл PAM для su. Файл sudo управляет взаимодействием с утилитой sudo.
- passwd. Этот файл настраивает PAM так, что информация об изменении пароля, выполненном с помощью программы passwd, передается KDC.
- vlock. Программа vlock блокирует консоль, не завершая при этом сеанс работы пользователя. Чтобы разблокировать консоль, необходимо ввести пароль. Как нетрудно догадаться, данный файл обеспечивает аутентификацию путем обращения программы vlock к KDC.
- xlock и xscreensaver. Программы с такими именами предназначены для блокирования сеанса X Window (т. е. они выполняют действия, аналогичные vlock). Программа xscreensaver автоматически блокирует сеанс, если в течение определенного периода времени пользователь не выполняет никаких действий.

Возможно, вы захотите настроить и другие программы для взаимодействия с Kerberos; при этом вам придется отредактировать соответствующие конфигурационные файлы PAM. Если какой-то из серверов уже сконфигурирован для работы с Kerberos, вам не надо модифицировать файлы PAM. Например, если у вас уже установлен кербиризованный FTP-сервер, вам не следует изменять файл `/etc/pam.d/ftp`. Подобные программы могут взаимодействовать с KDC, минуя PAM; при работе с ними нет необходимости вводить имя пользователя и пароль, как это приходится делать, используя программы, взаимодействующие посредством PAM.

Если некоторая программа использует PAM, то для обеспечения ее совместной работы с Kerberos вам необходимо добавить или заменить некоторые строки в конфигурационном файле PAM. В составе такого файла содержатся записи, определяющие основные действия, связанные с аутентификацией: `auth` (аутентификация пользователя), `account` (проверка корректности учетной записи), `password` (изменение пароля) и `session` (начало и завершение сеанса). В листинге 6.4 показано содержимое файла `gdm`, входящего в состав Kerberos PAM для Red Hat.

Листинг 6.4. Пример конфигурационного файла PAM для поддержки Kerberos

```
#%PAM-1.0
auth       required     /lib/security/pam_nologin.so
auth       sufficient  /lib/security/pam_unix.so shadow md5 \
nullok likeauth
auth       required     /lib/security/pam_krb5.so use_first_pass

account    required     /lib/security/pam_unix.so

password   required     /lib/security/pam_cracklib.so
password   required    /lib/security/pam_unix.so shadow md5 \
nullok use_authtok

session    required     /lib/security/pam_unix.so
session    optional     /lib/security/pam_krb5.so
session    optional     /lib/security/pam_console.so
```

В разных системах модули PAM настраиваются по-разному, поэтому содержимое конфигурационного файла может отличаться от приведенного в листинге 6.4. Часто настройка сводится к включению строки, указывающей на `pam_krb5.so`, и удалению ссылки на другой модуль.

В данном примере наиболее важны последняя запись `auth` и вторая запись `session`, которые настраивают PAM для использования Kerberos при регистрации пользователя и завершении его работы. В записи `auth` содержится параметр `use_first_pass`, который сообщает Kerberos PAM о том, что для поддержки сеанса используется первый пароль. В результате модуль действует подобно `kinit`, получая и сохраняя TGT. Аналогично могут быть настроены многие модули PAM, но для установки конфигурации некоторых из них необходимо выполнить дополнительные действия. Так, например, может

потребоваться дополнительная запись password, которая помещается после существующих и имеет следующий вид:

```
password    required    /lib/security/pam_krb5.so use_authtok
```

Это необходимо в случае, если модуль `password` используется программой `passwd`, которая изменяет пароль, но не выполняет аутентификацию пользователя. В некоторых файлах не должны присутствовать записи `session`, так как это приведет к разрушению билетов. Например, недопустимо, чтобы модули `xscreensaver` и `linuxconf` разрушали билеты; при завершении соответствующих программ возобновляется текущий сеанс, в котором билеты должны быть действительны.

В некоторых случаях приходится удалять существующие записи из файлов, размещенных в `/etc/pam.d`. В частности, если вы добавляете запись, указывающую на `pam_krb5.so`, а в файле уже есть запись такого же типа, которая ссылается на `pam_pwdb.so`, существующую запись необходимо удалить. Библиотека `pam_pwdb.so` обеспечивает непосредственный доступ к базе данных паролей, и если обе записи присутствуют, в аутентификации участвуют как локальная база паролей, так и база данных Kerberos. Возможно, в некоторых ситуациях, когда требуется чрезвычайно высокая степень защиты, такой подход оправдан, но в большинстве случаев он лишь уменьшает степень гибкости Kerberos, так как пользователь вынужден менять пароль и на KDC, и на рабочей станции. Если в конфигурационном файле PAM password предусмотрена двойная проверка, при работе с одной рабочей станцией настройки легко согласовать с помощью инструмента `passwd`, однако отслеживать изменения во всей сети достаточно сложно.

Новая конфигурация PAM становится доступна сразу после внесения соответствующих изменений; перезапускать программы PAM не требуется. Если установленная конфигурация оказывает воздействие на сервер, который уже выполняется в системе, то чтобы он работал с учетом новой конфигурации его, возможно, придется перезапустить. Если установки влияют на регистрацию в системе, они окажут требуемое воздействие лишь после завершения сеанса работы пользователя. При использовании средств регистрации с графическим интерфейсом, например GDM, может понадобиться перезапустить сервер.

Резюме

Kerberos — мощный инструмент, позволяющий централизованно выполнять аутентификацию в сети. Протокол обмена предполагает передачу шифрованных сообщений и наличие централизованной базы данных, которая используется серверами приложений, рабочими станциями и основным сервером Kerberos для аутентификации. Применяя средства Kerberos, можно добиться того, что после регистрации на некоторой рабочей станции пользователь сможет работать со всеми сетевыми службами, при этом повторный ввод пароля не потребуется. Исходный пароль никогда не передается по сети, поэтому вероятность того, что пароль попадет в руки злоумышленника, становится минимальной. Наличие централизованной базы данных существенно упрощает поддержку учетных записей.

Существует несколько реализаций Kerberos, предназначенных для применения в системе Linux; некоторые из них проще в использовании, чем другие. Для того чтобы сконфигурировать сеть Kerberos, надо установить на всех компьютерах как минимум подмножество системы Kerberos. Один из компьютеров должен выполнять функции центра распространения ключей (key distribution center — KDC); именно на этой машине располагается база данных с информацией о пользователях. На рабочих станциях и серверах необходимо установить керберизованные версии клиентских и серверных программ. Несмотря на то что настройка этих программ требует времени и усилий, достигнутый в результате использования Kerberos более высокий уровень защиты и преимущества централизованного администрирования оправдывают затраты. В особенности это заметно в сетях среднего и большого размера.

Глава 7

Совместное использование файлов и принтеров с помощью Samba

В конце 1990-х в локальных сетях все чаще стали устанавливать компьютеры под управлением Linux. Увлеченные перспективой затрачивать для решения сложных задач относительно небольшие ресурсы, администраторы использовали Linux даже тогда, когда это было совершенно не оправдано. Часто такими компьютерами заменяли дорогие и ненадежные серверы Windows, в результате пришлось решать задачу взаимодействия клиентов, работающих под управлением Windows, и Linux-серверов, т. е. возникла потребность в специальном инструменте, позволяющем поддерживать новую конфигурацию сети. Таким инструментом стал продукт Samba — сервер, обеспечивающий поддержку протокола SMB (Server Message Block — блок сообщений сервера), который в настоящее время известен также под названием CIFS (Common Internet Filesystem — общая межсетевая файловая система). SMB/CIFS — это протокол, позволяющий организовывать совместное использование файлов и принтеров и работающий на базе NetBIOS (набор протоколов, широко используемый в сетях, содержащих компьютеры под управлением Windows). Другими словами, Samba позволяет компьютеру под управлением Linux выполнять функции файлового сервера и сервера печати в сетях, содержащих компьютеры Windows. В настоящее время Samba очень хорошо справляется с этой задачей, кроме того, данный продукт постоянно дорабатывается и дополняется новыми средствами.

В начале данной главы речь пойдет о роли, которую Samba играет в современных сетях, а затем мы обсудим общие вопросы настройки Samba. В частности, здесь будут рассмотрены конфигурация Samba как контроллера домена, основного броузера и сервера имен NetBIOS; такие функции не имеют непосредственного отношения к разделению файлов, но их приходится выполнять в сетях NetBIOS. Средства Samba, предназначенные для разделения файлов и принтеров, понять не сложно; при их обсуждении будет уделено внимание ряду параметров, позволяющих изменить поведение системы. Завершается данная глава рассмотрением средств автоматизации Samba, с помощью которых решают-

ся задачи, обычно не имеющие непосредственного отношения к организации файловых серверов и серверов печати.

В простейших случаях работу сервера Samba можно организовать, не внося существенных изменений в содержимое конфигурационных файлов. Несмотря на это, Samba остается чрезвычайно сложным продуктом, на работу которого оказывают влияние многочисленные параметры. Подробное описание формата конфигурационного файла приведено в справочной системе; для того чтобы получить необходимую информацию, надо выполнить команду man `smb.conf`. Тем, кто хочет более детально ознакомиться с функционированием Samba, можно порекомендовать обратиться к специальным изданиям на эту тему, например, к моей книге *Linux Samba Server Administration* (Sybex, 2001), а также к книге Экштейна (Eckstein) и Коллиера-Брауна (Collier-Brown) *Using Samba* (O'Reilly, 1999).

Использование сервера Samba

Несмотря на то что сервер Samba может выполнять различные функции, прежде всего, он представляет собой инструмент для разделения файлов и принтеров. Разделение файлов означает, что на клиентской машине можно смонтировать часть файловой системы удаленного компьютера и обращаться к ней как к локальной. С файлами, находящимися в файловой системе удаленной машины, могут работать различные приложения на клиентском компьютере, например, некоторый файл можно загрузить в текстовый редактор, внести в него изменения и сохранить на сервере. Подобные операции часто выполняются в сетевой среде офиса. Средства монтирования каталогов удаленной машины позволяют хранить на сервере файлы с данными и коды приложений. Разделение принтеров означает, что пользователи могут выводить данные на устройства печати, подключенные к серверам. Организация пула принтеров позволяет сэкономить ресурсы.

Поскольку NetBIOS и SMB/CIFS наследуют некоторые характеристики систем DOS и Windows, очевидно, что Samba целесообразнее всего применять в сетях, использующих компьютеры под управлением именно этих операционных систем. Ряд свойств Samba упрощает обмен данными с DOS и Windows. Например, всем известно, что в именах файлов DOS и Windows не учитывается регистр символов; имена `FILE.TXT`, `file.txt` и `File.txt` представляют один и тот же файл. В Linux, напротив, имена файлов зависят от регистра символов. Обеспечивая совместную работу Windows и Linux, Samba позволяет обращаться к файлам без учета регистра. Кроме того, SMB/CIFS поддерживает такие особенности файловой системы DOS и Windows, как атрибуты скрытых файлов и файлов архивов. Скрытый файл — это файл, который в обычных условиях не доступен для пользователей, а файл архива — это файл, содержащий резервную копию данных. В файловой системе Linux скрытые файлы и файлы архивов отсутствуют, но Samba позволяет устанавливать и использовать соответствующие признаки. Подобные требования предъявляются при обмене файлами с другими операционными системами, например с системой OS/2, поэтому серверы Samba могут быть использованы и для работы с ними.

Samba применяется даже в сетях, не использующих компьютеры под управлением DOS, Windows, OS/2 и других систем, в которых основным протоколом разделения файлов является SMB/CIFS. UNIX, Macintosh, BeOS и другие системы также поддерживают SMB/CIFS. Если такая поддержка не реализована в самой системе, она обеспечивается продуктами независимых производителей. Linux позволяет работать и с другими протоко-

лами разделения файлов (в частности, Linux поддерживает средства NFS, которые будут подробно рассмотрены в главе 8), однако в некоторых случаях использование Samba предпочтительнее. Модель защиты SMB/CIFS (к которой для аутентификации применяются пользовательские имена и пароли) отличается от модели NFS (где компьютеры идентифицируются по IP-адресам, а за безопасность системы отвечают средства защиты клиента).

Настройка Samba

При настройке Samba устанавливаются опции общего назначения, а также выбирается конфигурация конкретных разделяемых объектов Samba (разделяемыми объектами считаются каталоги и принтеры, которые Samba предоставляет клиентам). Опции общего назначения чрезвычайно важны, так как если они установлены некорректно, то доступ к серверу станет невозможным. Кроме **того,** эти настройки влияют на выполнение задач, непосредственно не связанных с разделением файлов и принтеров. К таким задачам относится, например, идентификация компьютеров в сети NetBIOS.

Конфигурационный файл Samba

Для настройки Samba используется файл **smb.conf.** В большинстве дистрибутивных пакетов Linux он помещается в один из следующих каталогов: /etc, /etc/samba или **/etc/samba.d.** Подобно многим другим конфигурационным файлам Linux, в **smb.** conf для обозначения комментариев в начало строки включается символ #. Остальные строки файла разбиты на разделы, каждый из которых содержит определение разделяемого объекта. В начале раздела находится имя разделяемого объекта, помещенное в квадратные скобки:

`[имя_разделяемого_объекта]`

Последующие строки раздела определяют характеристики объекта — имя каталога, соответствующего разделяемому объекту, особенности обработки файлов и т. д. Раздел, содержащий опции общего назначения, выглядит как определение разделяемого объекта, но таковым не является. В начале этого раздела указано имя `[global]`. Некоторые из опций, указанных в этом разделе, могут быть переопределены для конкретных объектов, а другие опции могут находиться только в разделе `[global]`.

Для настройки Samba используются параметры, которые представляются в следующем формате:

`параметр = Значение`

Samba позволяет использовать при указании параметров символы как нижнего, так и верхнего регистра, но в большинстве случаев имена параметров содержат только символы нижнего регистра, а значения параметров начинаются с прописной буквы. Некоторые значения, например имена файлов Linux, зависят от регистра символов. Ряд параметров предполагает двоичные значения; в этом случае Yes, True и 1 являются синонимами (точно так же синонимами являются значения No, False и 0).

Идентификация сервера Samba

В сетях NetBIOS используется система имен, не связанная с доменными именами, применяемыми в сетях TCP/IP. Например, компьютеру `harding.threeroomco.com`

может соответствовать имя BILLY, принадлежащее домену USPRES. Для того чтобы удобнее было отличать имена TCP/IP от имен NetBIOS, последние будут представляться в данной главе символами верхнего регистра. В системе NetBIOS предусмотрены два уровня имен: имена компьютеров и имена рабочих групп или доменов. (Домены NetBIOS не имеют никакого отношения к доменам TCPЛР.) В Samba предусмотрены средства поддержки как имен компьютеров, так и имен рабочих групп и доменов.

Различие между рабочей группой и доменом NetBIOS не очень существенны. Рабочая группа — это набор компьютеров с общим именем группы. Домен отличается от рабочей группы тем, что в нем присутствует специальный компьютер, называемый контроллером домена, который обеспечивает централизованную аутентификацию и выполняет некоторые другие функции. Домен может занимать несколько сегментов сети; для рабочей группы реализовать такую конфигурацию затруднительно. При необходимости компьютер, на котором выполняется сервер Samba, может выступать в роли контроллера домена (этот вопрос будет обсуждаться далее в данной главе).

Имя рабочей группы или имя домена задается с помощью параметра workgroup:

```
workgroup = USPRES
```

Данный параметр указывает на то, что компьютер принадлежит рабочей группе USPRES. Система может взаимодействовать с членами других рабочих групп, но при выполнении ряда функций, например при просмотре сети средствами Windows, используется имя рабочей группы. Если имя группы будет задано некорректно, сервер Samba станет недоступным для броузеров Network Neighborhood и My Network Places. Эта проблема часто возникает при установке исходной конфигурации Samba.

По умолчанию Samba использует в качестве имени узла NetBIOS первый компонент доменного имени TCP/IP этого компьютера. Например, для компьютера harding.threeroomco.com Samba выберет NetBIOS-имя HARDING. Это значение можно изменить с помощью параметра netbios name. Параметр netbios aliases позволяет присвоить одному компьютеру несколько имен NetBIOS. Например, если в конфигурационном файле будут присутствовать приведенные ниже строки, то к компьютеру можно будет обращаться как по имени BILLY, так и по имени WILLIAM.

```
netbios name = BILLY
netbios aliases = WILLIAM
```

В большинстве случаев рекомендуется, чтобы имя компьютера, входящее в состав домена TCPЯР, и имя NetBIOS, совпадали. Это исключит недоразумения при обращении к узлу сети.

Защита системы

В ранних реализациях SMB/CIFS пароли передавались по сети в незашифрованном виде. Это давало возможность для перехвата их другими узлами локальной сети, а если в обмене данными участвовали маршрутизаторы, то пароль мог быть перехвачен и внешними компьютерами. В последующих реализациях SMB/CIFS были использованы средства шифрования паролей. Однако способы кодирования SMB/CIFS не совместимы со способами, которые используются для поддержки локальных паролей Linux. Закодированный пароль SMB/CIFS невозможно сравнить с записями, хранящимися в локальной

базе Linux, поэтому в Samba была реализована собственная база паролей. Эта база называется **smbpasswd**, а для управления ею используется одноименная утилита.

В системе Windows, начиная с версий Windows 95 OSR2 и Windows NT 4.0 SP3, по умолчанию используются зашифрованные пароли. Если сервер Samba настроен для передачи незакодированных паролей, взаимодействие с Windows будет невозможным. Для того чтобы устранить эту проблему, надо переконфигурировать либо сервер Samba, либо систему Windows. Чтобы изменить конфигурацию Samba, придется затратить меньше усилий, кроме того, такой подход гораздо предпочтительнее с точки зрения безопасности системы.

Обработкой зашифрованных паролей управляет параметр encrypt passwords. Для того чтобы сервер Samba выполнял проверку закодированных паролей в файле smbpasswd, следует задать значение Yes этого параметра. Чтобы включить зашифрованный пароль пользователя в файл smbpasswd, надо выполнить следующую команду:

```
# smbpasswd -a имя_пользователя
```

В процессе выполнения утилита smbpasswd запросит пароль (по требованию программы вам придется ввести его дважды). В момент вызова smbpasswd на компьютере должна существовать учетная запись пользователя с указанным именем, иначе программа smbpasswd не включит пароль в базу. При первом запуске утилита smbpasswd отобразит предупреждающее сообщение о том, что файл smbpasswd отсутствует. Однако в ходе дальнейшей работы программа автоматически создаст этот файл, поэтому на предупреждающее сообщение можно не обращать внимание.

Дополнительные меры по защите обеспечивают параметры hosts allow и hosts deny. Они действуют подобно файлам /etc/hosts.**allow** и /etc/hosts.**deny** TCPWrappers, которые рассматривались в главе 4. Данные параметры позволяют задавать список узлов, которые имеют право взаимодействовать с сервером, и список узлов, для которых такое взаимодействие запрещено. Например, приведенное ниже выражение разрешает взаимодействие с системой только для узлов 192.168.7.0/24 и algernon.**pangaea.edu**.

```
hosts allow = 192.168.7. algernon.pangaea.edu
```

Последующие разделы посвящены работе Samba в качестве сервера имен NetBIOS, основного броузера и контроллера домена. Читатели, не интересующиеся данными вопросами, могут перейти к разделу Организация файлового сервера с помощью Samba". Конфигурация, установленная по умолчанию, обеспечит работу Samba во многих сетях.

Samba как сервер имен NetBIOS

В сетях NetBIOS необходимы средства преобразования имен. Преобразование имен NetBIOS и TCP/IP выполняется различными способами; основные из них описаны ниже.

- Имена TCP/IP. Для преобразования можно использовать имена TCPЛP; в Windows 2000, Windows XP и Samba этот способ применяется по умолчанию. Следует заметить, что такое решение не имеет непосредственного отношения к NetBIOS.

- **Применение файла lmhosts**. Система может хранить информацию о соответствии имен IP-адресам в файле, который обычно называется Imhosts и выполняет те же функции, что и файл /etc/hosts в системе Linux.

- **Широковещательная передача.** Если компьютеру необходимо определить адрес другого компьютера, он передает запрос в широковещательном режиме. Средства, реализующие данный способ, просты в настройке и хорошо работают в небольших сетях. В сетях большого размера такое преобразование приводит к неоправданному увеличению трафика. Если сеть состоит из нескольких подсетей, объединенных маршрутизаторами, данный способ будет применим только в том случае, когда маршрутизаторы настроены для передачи широковещательных сообщений.

- **WINS-сервер**. Сервер NBNS (NetBIOS Name Service — служба имен NetBIOS), который также известен под названием WINS (Windows Internet Name Service — сервер Internet-имен для системы Windows), выполняет преобразование имен в IP-адреса.

Чтобы сервер Samba мог работать в качестве WINS-сервера, надо включить в раздел [global] файла smb.conf следующий параметр:

```
wins support = Yes
```

WINS-сервер не требует дополнительной настройки (по крайней мере на стороне сервера). Если клиенты и серверы NetBIOS могут обмениваться данными по сети, они регистрируются на WINS-сервере. WINS-сервер должен быть задан для каждого компьютера, подключенного к сети. В системе Windows для этого используется диалоговое окно TCP/IP Properties, показанное на рис. 7.1. Если вы установите флажок опции Use DHCP for WINS Resolution, Windows будет получать необходимую информацию от DHCP-сервера. (Вопросы настройки DHCP-сервера рассматривались в главе 5.)

Чтобы сервер Samba использовал для преобразования имен NetBIOS WINS-сервер, вам надо задать в файле smb.conf два параметра: wins server и name resolve order. В качестве значения первого параметра надо указать IP-адрес WINS-сервера. Второй параметр может принимать от одного до четырех значений, каждое из которых задает отдельный способ преобразования имен. Samba будет пытаться применить эти способы в том порядке, в котором они указаны в составе name resolve order. (Значения host и beast определяют соответственно обычный метод преобразования с использованием средств TCP/IP и широковещательную передачу, которая осуществляется для преобразования имен NetBIOS.) Пример использования указанных параметров приведен ниже.

```
wins server = 192.168.1.1
name resolve order = wins Imhosts host beast
```

Роль WINS-сервера может выполнять только одна система. Если таких серверов будет несколько, ненадежность выполнения процедуры преобразования снизится, так как некоторые компьютеры будут регистрироваться на одном сервере, а остальные — на других серверах. (В NetBIOS-сетях предусмотрен механизм использования вторичного сервера имен. Наличие вторичного сервера увеличивает надежность сети и обеспечивает работу клиентов при выходе основного сервера из строя. Однако система Samba не может работать в качестве вторичного сервера имен.)

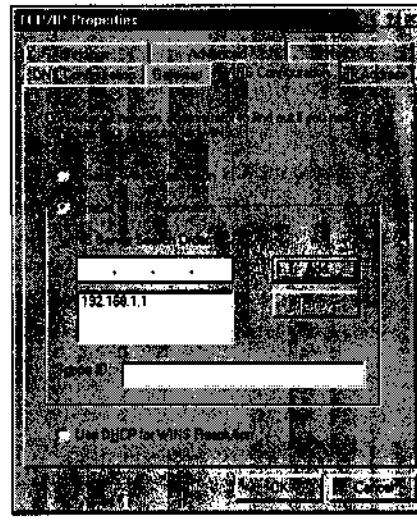


Рис. 7.1. Диалоговое окно TCP/IP Properties в системе Windows позволяет задавать адрес WINS-сервера

Samba как основной броузер

Выше я упоминал о броузерах Network Neighborhood и My Network Places. Эти программы не являются Web-броузерами. Они предназначены для просмотра данных, предоставляемых серверами SMB/CIFS, работающими в сети NetBIOS (рис. 7.2). Дважды щелкнув мышью на имени компьютера, вы увидите информацию о разделяемых объектах SMB/CIFS. Разделяемые объекты файлов отображаются в виде папок, а разделяемые объекты печати — в виде пиктограмм с изображением принтеров. Дважды щелкнув на изображении папки, вы получите доступ к разделяемым файлам и сможете работать с ними так же, как и с файлами на локальном жестком диске.

Такое взаимодействие удобно с точки зрения пользователя, но для того, чтобы оно стало возможным, необходимо иметь соответствующим образом настроенное программное обеспечение. Рассмотрим описанную ситуацию с точки зрения сетевого броузера Windows. Как он может получить информацию о том, какие компьютеры присутствуют в сети и какие объекты доступны для совместного использования? Для этого один из компьютеров в сегменте сети NetBIOS должен выполнять функции *основного локального броузера* (local master browser). Этот компьютер хранит список серверов SMB/CIFS, принадлежащих сегменту сети, и предоставляет данные клиентам SMB/CIFS в ответ на их запросы. В результате, для того, чтобы иметь сведения о разделяемых ресурсах, клиент должен знать только адрес основного броузера, а такую информацию он может получить с помощью широковещательного сообщения. Сразу после загрузки клиент передает в широковещательном режиме запрос на получение адреса основного броузера. То же делает и сервер, но при этом он регистрирует свои ресурсы. В результате на основном броузере всегда содержится самая новая информация о конфигурации сети.

В домене NetBIOS используется другой тип основного буроузера, который называется *основным броузером домена* (domain master browser) и взаимодействует с основными

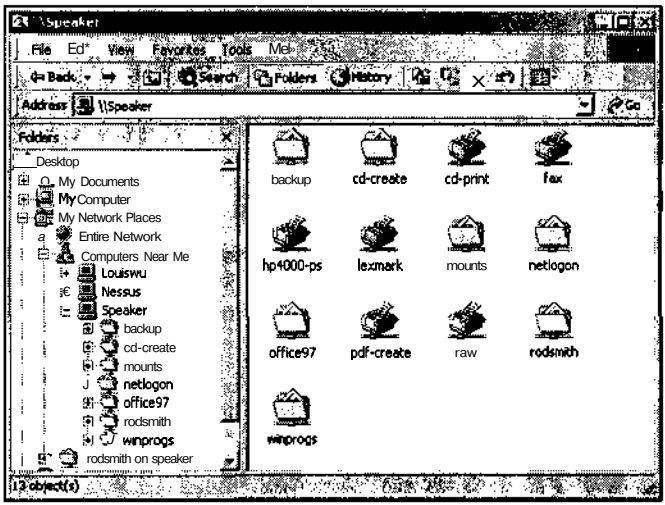


Рис. 7.2. Windows предоставляет информацию о компьютерах вашего домена или рабочей группы, а также о компьютерах, принадлежащих другим доменам или рабочим группам, которые находятся в текущем сегменте локальной сети

локальными буроузерами, расположенными в подсетях, входящих в домен. Эти броузеры обмениваются содержащимися на них списками, в результате основной броузер домена получает (с некоторой задержкой) все текущие сведения о домене.

В роли основного броузера домена обычно выступает контроллер домена. Компьютер специально настраивается для выполнения соответствующих функций. Основные локальные броузеры определяются с помощью процедуры *выборов* (election), в ходе которых компьютеры, принадлежащие сегменту сети, выступают в роли претендентов и объявляют в сети свои возможности. Один из претендентов оказывается победителем и принимает на себя обязанности основного локального броузера. Важно, чтобы основные броузеры работали надежно; если компьютер, выполняющий функции основного броузера часто выходит из строя или перезагружается, то в это время пользователи не смогут просматривать ресурсы сети. Важно также, чтобы сервер Samba не проявлял излишней "настойчивости", пытаясь стать победителем. Установка одного из параметров (он будет рассмотрен ниже) приведет к тому, что Samba будет периодически выставлять свою кандидатуру на роль основного локального броузера, в результате чего процедура просмотра ресурсов сети станет работать с перебоями.

Чтобы настроить сервер Samba для выполнения функций основного локального броузера, надо установить параметры, позволяющие ему участвовать в выборах и побеждать. Для этого в раздел `[global]` надо включить следующие строки:

```
browse list = Yes
local master = Yes
preferred master = Yes
os level = 65
```

Параметр browse list указывает на то, что сервер Samba должен подготавливать список просмотра, который может совместно использоваться другими системами. По умолчанию принимается значение Yes данного параметра, поэтому он может не указываться. Для параметра local master по умолчанию также предполагается значение Yes. Это значение сообщает серверу Samba на то, что он должен участвовать в выборах, но не гарантирует победу. Если задано значение Yes параметра preferred master, сервер сразу после запуска выставит свою кандидатуру на выборы и, если не станет победителем, периодически будет повторять подобные попытки. По умолчанию для данного параметра устанавливается значение No. Если в сети работает несколько серверов Samba, значение Yes параметра preferred master можно задавать только на одном компьютере. В противном случае работа сети будет нарушена вследствие периодически повторяющихся процедур выборов. Параметр os level задает значение основного критерия, принимающегося во внимание при проведении выборов. Чем выше это значение, тем больше вероятность победы. Значение os level = 65 позволяет серверу одержать победу над компьютерами под управлением Windows (по крайней мере над системами Windows Me и Windows 2000), но не гарантирует победу над другими серверами Samba, так как в их конфигурационных файлах могут быть указаны более высокие значения os level. Если же вы не хотите, чтобы сервер Samba стал основным броузером, вам надо задать значение os level, равное 0.

Для настройки Samba в качестве основного броузера домена надо, во-первых, установить рассмотренные выше параметры так, чтобы компьютер выиграл выборы и стал основным локальным броузером, а во-вторых, задать параметр domain master = Yes. Вы также должны сконфигурировать систему для выполнения функций главного контроллера домена. Необходимые для этого действия будут рассматриваться в следующем разделе. Как и в случае основного локального броузера, вам не следует настраивать больше одной системы (Samba или Windows) в качестве основного броузера домена. Если сеть сконфигурирована так, что компьютеры объединяются в рабочие группы, параметр domain master указывать не следует.

Samba как контроллер домена

Как вы уже знаете, компьютеры в сетях NetBIOS объединяются в рабочие группы, либо в домены. Эти структуры похожи друг на друга, но в домене степень централизации несколько выше. В домене присутствует компьютер, называемый *контроллером домена,* который выполняет аутентификацию пользователей. Вместо того чтобы предоставлять каждому серверу самостоятельно решать, разрешить ли пользователю обращаться к разделяемым объектам, эту задачу выполняет контроллер домена. Использование такого контроллера существенно упрощает администрирование сети, содержащей несколько серверов, в особенности, если администратору часто приходится создавать учетные записи для новых пользователей и удалять старые записи. Однако для настройки контроллера домена необходимо приложить определенные усилия. Контроллер домена может решать также другие задачи. Например, с его помощью можно поддерживать сценарии регистрации или создавать профили Windows. Профили — это специальные наборы установок, которые могут вместе с пользователями перемещаться от одного клиента к другому. Конфигурирование контроллера домена — достаточно сложная задача, поэтому в данном разделе приводятся лишь общие сведения о его настройке.

Существуют два типа контроллеров домена NetBIOS: *основной контроллер домена* (primary domain controller — PDC) и *резервный контроллер домена* (backup domain controller — BDC). В случае если PDC выходит из строя или становится недоступным, его функции выполняет BDC. Samba поддерживает возможности PBC, но не может выступать в роли BBC.

Чтобы настроить Samba для работы в качестве PBC, необходимо добавить в раздел `[global]` файла `smb.conf` следующие параметры:

```
security = User
encrypt passwords = Yes
domain logons = Yes
```

Параметр `security` указывает на то, что запросы на доступ к разделяемым ресурсам должны обрабатываться с использованием имен и паролей Linux. (Такая установка принята по умолчанию для Samba 2.0.0 и более поздних версий данного продукта.) Параметр `encrypt passwords` управляет шифрованием паролей. Контроллер домена должен передавать все пароли в закодированном виде. Основным параметром, определяющим действия PBC, является `domain logons`. Если значение `domain logons` равно Yes, этот параметр указывает Samba на то, что сервер должен принимать запросы на регистрацию в домене и обрабатывать их с учетом содержимого файла `smbpasswd`. В системе Windows PBC выполняет также функции основного броузера домена и WINS-сервера. Желательно настроить подобным образом и сервер Samba.

Для установки доменного имени надо использовать параметр workgroup, включив его в файл `smb.conf`.

Если в состав сети входят компьютеры под управлением Windows NT, 2000 или XP, вам надо предпринять дополнительные действия по настройке системы. Во-первых, необходимо помнить, что вереии Samba, предшествующие 2.2.0, не поддерживают взаимодействие с клиентами Windows NT, версии, выпущенные ранее 2.2.1 а, не поддерживают клиентов Windows XP и Windows 2000 Service Pack 2 (SP2). Samba 2.2.0 и более поздние версии поддерживают многие новые возможности, реализованные в клиентах NT/2000 и даже в Windows NT 4.0. Это надо учитывать, выбирая версию Samba. Во-вторых, для использования Samba в качестве контроллера домена для каждого из клиентов NT/2000/XP необходимо создать в системе Linux структуру, которая называется *доверительной учетной записью* (trust account). Сделать это можно, выполнив следующие команды:

```
# groupadd -r trust
# useradd -r -g trust -d /dev/null -s /dev/null client$
# smbpasswd -a -m client
```

Команду `groupadd` надо задать только один раз; с ее помощью создается специальная группа для доверительной учетной записи. (При необходимости вы можете использовать одну из существующих групп, но это не рекомендуется. Гораздо лучше сформировать для этой цели отдельную группу.) Команда `useradd` создает специальную учетную запись для компьютера NetBIOS с именем CLIENT. Символ $ после имени пользователя обязателен. Команда `smbpasswd` добавляет запись `client` в файл smbpasswd. Эта команда не требует указания символа $. Когда клиент Windows NT/2000/XP попытается обратиться к домену, он будет зарегистрирован с помощью доверительной учетной записи; при этом для аутентификации будет использована база паролей Samba.

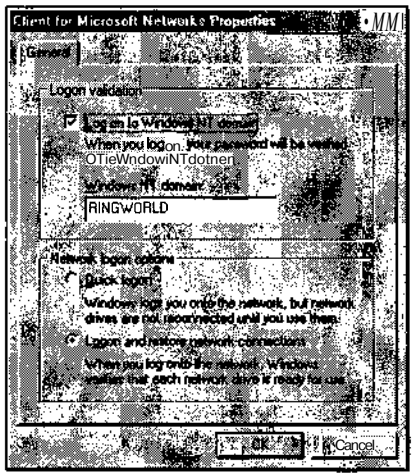


Рис. 7.3. Выбор конфигурации домена на клиентской машине под управлением Windows сводится к установке флажка опции и вводу имени домена

Конфигурация домена требует также изменить способ настройки клиента. Это можно сделать с помощью управляющей панели. В Windows 9x/Me для этого надо выбрать в окне Network пункт Microsoft Networks и щелкнуть на кнопке Properties. Система отобразит диалоговое окно Client for Microsoft Networks Properties, показанное на рис. 7.3. Установите флажок опции Log On to Windows NT Domain и введите имя домена. Чтобы сделать то же самое в системе Windows 2000, надо щелкнуть правой кнопкой мыши на пиктограмме My Computer и выбрать пункт Properties, чтобы открыть диалоговое окно System Properties. Затем необходимо выбрать Network Identification и щелкнуть на кнопке Properties. После установки конфигурации домена система запросит пользовательское имя и пароль администратора. В ответ надо, как обычно, ввести имя пользователя и пароль.

Организация файлового сервера с помощью Samba

Настройка файлового сервера для выполнения функций контроллера домена и WINS-сервера часто бывает необходима, но в подавляющем большинстве случаев сервер Samba используется как обыкновенный файловый сервер. Для того чтобы настроить Samba таким образом, надо определить разделяемые файлы, включив эти определения в файл `smb.conf` после раздела `[global]`. Для описания разделяемых файлов достаточно нескольких строк, однако не исключено, что вы захотите добавить к ним ряд важных параметров.

Описание разделяемых объектов

Описание разделяемого объекта Samba, обеспечивающего совместный доступ к файлам, выглядит следующим образом:

```
[sample]
    path = /home/samba/shared-dir
    browseable = Yes
    read only = No
```

В данном примере определяется разделяемый объект с именем [sample]. В окне броузера Windows (см. рис. 7.2) он будет отображаться как SAMPLE. Разделяемому объекту соответствует каталог /home/samba/shared-dir. Если пользователь захочет просмотреть содержимое объекта SAMPLE, он увидит файлы из этого каталога. Строка browseable = Yes не является необходимой, поскольку именно это значение параметра browseable принято по умолчанию. Значение Yes параметра browseable указывает на то, что объект должен присутствовать в списке просмотра. (Чтобы удалить разделяемый объект из этого списка, надо указать параметр browseable = No, но это не сделает данный объект недоступным. Пользователи, которые знают о наличии такого объекта, могут обратиться к нему, непосредственно указав имя объекта в строке Address броузера.) По умолчанию Samba создает разделяемые объекты, предназначенные только для чтения; клиенты не могут записывать в них данные. Чтобы объект допускал как чтение, так и запись, необходимо включить в состав описания параметр read only = No либо один из его синонимов: writeable = Yes или write ok = Yes. Информация о владельце и правах доступа к файлам, входящим в состав разделяемого объекта, остается такой же, как и в системе Linux. При необходимости Samba позволяет переопределить владельца и права (действия, необходимые для этого, описаны ниже в данной главе).

Во многих сетях серверы Samba применяются для предоставления пользователям дополнительного дискового пространства, которое необходимо им для хранения документов. Для того чтобы упростить обслуживание пользователей, в системе Samba действует специальное соглашение, связанное с использованием разделяемого объекта [homes]. Если вы опишете в конфигурационном файле объект [homes], Samba будет интерпретировать его следующим образом.

- Задавать параметр path нет необходимости. Samba использует рабочий каталог пользователя, обратившегося к разделяемому объекту.
- В списке просмотра разделяемый объект отображается под именем, совпадающим с именем пользователя (например, объект rodsmith, представленный на рис. 7.2). Если вы зададите параметр browseable = No, объект [homes] не будет отображаться в списке просмотра, но будет доступен под именем пользователя.

Во многих дистрибутивных пакетах разделяемый объект [homes] изначально присутствует в файле smb.conf. Поэтому, даже если вы не объявите новые разделяемые объекты, конфигурация, установленная по умолчанию, позволит вам использовать сервер Samba для решения многих практических задач. (Имя рабочей группы вам придется задать самостоятельно и, вероятнее всего, вам необходимо будет определить политику шифрования паролей.)

Настраивая сервер Samba, вы можете описывать как угодно много разделяемых объектов, но очевидно, что определять несколько объектов [homes] не имеет смысла. Кроме

того, если вы обнаружите, что некоторый параметр присутствует в описаниях всех разделяемых объектов, целесообразно перенести его в раздел [global]. Если вы сделаете это, значение параметра будет использоваться по умолчанию для всех объектов.

Поддержка имен файлов Windows

В системах Linux и Windows действуют разные соглашения по именованию файлов. Если в вашей сети, кроме Windows, присутствуют также клиенты DOS, то при настройке Samba следует учесть, что правила именования DOS-файлов отличаются от правил, принятых не только в Windows. Таким образом, при работе в сети с компьютерами Windows и DOS сервер Samba должен решить сложную задачу: представить файловую систему Linux в формате, совместимом с файловыми системами DOS и Windows.

Одно из самых важных отличий файловой системы Linux от системы Windows состоит в том, что имена файлов Linux чувствительны к регистру символов, т. е. имена FILE.TXT, file.txt и File.txt идентифицируют различные файлы; при помещении их в один каталог конфликт не возникает. Это также означает, что при вводе имени файла пользователь должен следить за тем, чтобы были заданы символы требуемого регистра. В отличие от Linux, Windows хранит сведения о регистре, но не учитывает их при сравнении имен файлов, поэтому два файла с именами, отличающимися только регистром символов, не могут существовать в одном и том же каталоге. Файловая система DOS нечувствительна к регистру символов; если даже пользователь задал имя файла буквами нижнего регистра, система преобразует их в прописные буквы.

Параметр case sensitive, помещаемый в конфигурационный файл, определяет, должен ли сервер Samba учитывать регистр символов в именах файлов. По умолчанию принимается значение No данного параметра, что позволяет серверу Samba работать с клиентами Windows и DOS. Если клиент запросит некоторый файл, Samba проверит все файлы, отличающиеся от указанного только регистром символов, т. е. будет имитировать поведение Windows. Недостаток подобного подхода состоит в том, что производительность системы несколько снижается. Если вы хотите добиться максимальной производительности, вам надо задать параметр sensitive = Yes, но при этом некоторые из программ Windows будут работать с ошибками. Эти ошибки, конечно же, связаны с особенностями взаимодействия с сервером Samba. Параметр sensitive = Yes целесообразно использовать при работе с клиентами, которые выполняются в тех операционных системах, в которых, подобно Linux, в именах файлов и каталогов учитывается регистр символов.

Параметры preserve case и short preserve case определяют, должен ли сервер Samba сохранять информацию о регистре символов в именах файлов. Если установлено значение Yes, Samba сохраняет имена файлов именно в том виде, в котором их задают клиенты. При установленном значении No Samba будет преобразовывать буквы в именах файлов к верхнему или нижнему регистру. Конкретный регистр зависит от параметра default case. По умолчанию для этого параметра установлено значение Lower, но при необходимости вы можете задать значение Upper. Параметр preserve case воздействует на все файлы, но для коротких имен более высокий приоритет имеет параметр short preserve case. (Короткими именами называются имена, составленные по соглашениям DOS, т. е. содержащие до 8 символов в имени файла и до 3 символов в расширении; такие имена принято также называть именами 8.3.) Если в вашей сети содержится большое количество DOS-клиентов, следует задать

параметр short preserve case = No. В результате в системе Linux имена файлов будут составляться из символов нижнего регистра, но DOS-клиентам эти же имена будут доступны преобразованными к верхнему регистру.

Средства SMB/CIFS обеспечивают доставку имен 8.3 даже в том случае, если исходные имена содержат большее количество символов. Это позволяет организовывать доступ к файлам с длинными именами для клиентов, работающих в DOS или 16-битовой системе Windows. Поскольку в Linux длина имени файла не ограничена, сервер Samba должен динамически генерировать имена 8.3. Параметр mangled names = Yes (значение по умолчанию) разрешает поддержку имен 8.3; если же вы укажете mangled names = No, создание таких имен будет запрещено.

Владелец файла и права доступа

Средства защиты Linux базируются на понятиях принадлежности файла определенному владельцу и правах доступа к нему, принятых в системе UNIX. Однако SMB/CIFS использует те же признаки несколько по-другому. Средства SMB/CIFS регистрируют пользователей, обратившихся к серверу, анализируя имена и пароли, таким образом, по умолчанию Samba использует для этой цели учетные записи Linux. Если пароли передаются в незашифрованном виде, Samba применяет стандартный механизм аутентификации Linux, а при работе с зашифрованными паролями сервер Samba самостоятельно идентифицирует пользователя. Samba позволяет проводить сеанс работы от имени различных пользователей. В частности, параметры force user и force group позволяют настроить Samba так, что все обращения к некоторому разделяемому объекту будут интерпретироваться так, как будто бы они поступают от другого пользователя или от пользователя, принадлежащего другой группе. Рассмотрим следующее описание разделяемого объекта:

```
[jekyl]
    path = /home/samba/jekyl
    read only = No
    force user = hyde
```

Каждый пользователь, обратившийся к данному объекту, будет выполнять любые действия так, как будто их выполняет пользователь, которому в Linux соответствует учетная запись hyde. Если к разделяемому объекту обратится пользователь muriel и создаст в нем файл, владельцем этого файла будет hyde. То же самое произойдет, если файл создаст пользователь henry. Любой пользователь, работающий с данным объектом посредством Samba, может читать файлы, расположенные в соответствующем каталоге, даже если ему запрещено делать это при обычной регистрации в системе Linux. Доступ предоставляется даже тем пользователям, которым в системе Linux запрещено просматривать содержимое каталога /home/samba/jekyl. Пользователи работают с файлами в составе разделяемого объекта так, как будто это делает пользователь hyde. Подобным образом действует параметр force group, но он учитывает не владельца файла, а принадлежность файла группе.

Обращаясь к разделяемому объекту, для которого указан параметр force user, пользователь указывает собственный пароль.

Параметры force user и force group существенно упрощают ряд действий по настройке Samba. Предположим, например, что вы хотите создать разделяемый объект,

в пределах которого пользователи смогут свободно обмениваться документами. Для этого вам достаточно задать параметр `force user`; в результате все файлы, созданные в соответствующем каталоге, будут принадлежать одному владельцу, и их сможет прочитать любой пользователь, который получит доступ к данному объекту. Специально для этой цели вы можете создать отдельную учетную запись. О том, как ограничить круг пользователей, имеющих право обращаться к разделяемому объекту, рассказывается далее в этой главе.

Несмотря на то что и Linux, и SMB/CIFS поддерживают пользовательские имена, механизмы обеспечения доступа, используемые ими, не совпадают. Если Samba применяется для взаимодействия с клиентами DOS и Windows 9x/Me, клиент вовсе не использует сведения о правах доступа. Поэтому вы можете устанавливать для файлов, созданных этими клиентами, любые права. Сделать это можно с помощью параметров `create mask` и `directory mask`, которые позволяют задавать права доступа к файлам и каталогам. В качестве значения каждого из параметров задается трех- или четырехзначное восьмеричное число; оно представляет признаки доступа, которые могут быть установлены для файла, создаваемого средствами Samba. По умолчанию для параметра `create mask` принимается значение 0744, а для параметра `directory mask` — значение 0755. Оба значения разрешают владельцу файла чтение и запись, а членам группы и всем остальным пользователям — только чтение. Вы можете задавать значения данных параметров по-своему, но при этом необходимо учитывать, что биты прав доступа используются для представления признаков DOS, определяющих скрытые, системные файлы и файлы архивов.

В системах DOS и Windows используются три атрибута файлов, которые не поддерживаются в файловой системе Linux. Samba отображает эти атрибуты в биты, определяющие права запуска файлов на выполнение. В Samba предусмотрены параметры, позволяющие управлять отображением атрибутов файлов в права доступа. Эти параметры перечислены ниже.

- `map archive`. Если для данного параметра установлено значение Yes (значение по умолчанию), признак файла архива DOS отображается в бит, определяющий права владельца на выполнение файла. Клиенты DOS и Windows устанавливают этот признак при создании нового файла и сбрасывают его, если с помощью специальных программ создается резервная копия данного файла. По этой причине файлы, созданные по инициативе клиентов Samba, выглядят как исполняемые. Данный эффект наблюдается только в том случае, когда используется значение параметра `create mask`, принятое по умолчанию.

- `map system`. Если для данного параметра установлено значение Yes, признак системного файла DOS отображается в бит, определяющий права группы на выполнение файла. По умолчанию используется значение No, т. е. отображение не осуществляется. В DOS и Windows этим признаком помечаются файлы специального назначения. Вероятность того, что такие файлы будут сохранены на сервере Samba, очень мала.

- `map hidden`. Если для данного параметра установлено значение Yes, признак скрытого файла DOS отображается в бит, определяющий права любого пользователя системы Linux на выполнение файла. По умолчанию принимается значение No. В системах DOS и Windows файлы, для которых установлен данный признак,

остаются доступными, но они не отображаются при выводе содержимого каталогов и отсутствуют в диалоговых окнах, предназначенных для выбора файлов.

По умолчанию Samba устанавливает признак скрытого файла для файлов Linux, имена которых начинаются с точки. Linux интерпретирует подобные файлы приблизительно так же, как DOS — скрытые файлы. Если такое поведение системы вас не устраивает, вы можете изменить настройку, задав параметр `hide dot files = No`.

Значение параметра `create mask`, используемое по умолчанию, отражает значения параметров `map archive`, `map system` и `map hidden`. Если вы хотите изменить параметр `map system` или `map hidden`, вам надо изменить параметр `create mask`, добавив единицу соответственно к предпоследней или последней цифре его значения.

В системах Windows NT/2000/XP используется более сложная модель защиты, чем в Windows 9x/Me. В Windows NT/2000/XP поддерживаются ACL (Access Control Lists — списки контроля доступа), которые позволяют непосредственно определить уровень доступа конкретного пользователя к файлу. Samba поддерживает отображение прав доступа для владельца, группы и всех пользователей Linux в Windows ACL. Это позволяет хранить на сервере Samba информацию о правах, установленных в системе Windows NT/2000/XP. Поддержка ACL задается с помощью параметра `nt acl support`; по умолчанию принимается значение Yes. Если вы не хотите поддерживать соответствие между правами в Windows NT/2000/XP и Linux, задайте для данного параметра значение No.

В файловых системах DOS и Windows 9x/Me существует признак, который разрешает или запрещает запись в файл. Этот признак воздействует на права владельца, группы и пользователя, поэтому для его поддержки необходимо соответствующим образом изменить значения параметра `create mask` или `directory mask`. ACL обеспечивают полный контроль доступа к файлам, в частности, позволяют разрешить или запретить запись в файл.

Ограничение доступа к разделяемым объектам

Samba использует различные средства контроля доступа к серверу. В качестве примера можно привести уже упоминавшиеся параметры `hosts allow` и `hosts deny` и, конечно же, модель аутентификации, согласно которой пользователь должен указывать имя и пароль. Samba также предоставляет средства контроля доступа к отдельным разделяемым объектам. Наиболее важными средствами управления доступом являются параметры `valid users` и `invalid users`. В качестве значений этих параметров задаются списки пользователей, которым соответственно разрешено или запрещено обращаться к разделяемому объекту. Если вы используете параметр `valid users`, задайте перечень имен пользователей, разделенных пробелами. Эти пользователи получат право доступа к объекту, а для остальных доступ будет запрещен. Аналогично, параметр `invalid users` используется для создания "черного списка". Даже если пользователи, указанные в нем, имеют право обращаться ко всем остальным объектам, доступ к данному объекту для них будет закрыт.

Помимо `valid users` и `invalid users`, для контроля доступа могут также быть использованы параметры `write list` и `read list`. Эти параметры позволяют переопределять для отдельных пользователей установки, разрешающие чтение и запись или

только чтение. Предположим, что вы создали разделяемый объект и поместили в него программные файлы. Очевидно, что подавляющее большинство пользователей не должны вносить изменения в содержимое этого объекта, поэтому данный разделяемый объект целесообразно определить как предназначенный только для чтения. Однако некоторые пользователи будут заниматься обновлением программ, поэтому им надо предоставить право записи. Этих пользователей можно определить с помощью параметра `write list`.

В качестве примера применения описанных выше параметров рассмотрим следующий фрагмент конфигурационного файла:

```
[control]
    path = /home/samba/control
    read only = Yes
    invalid users = thomas susan
    write list = gertrude henry
```

Для большинства пользователей данный объект допускает только чтение. Двум пользователям (thomas и susan) доступ к объекту полностью запрещен, а пользователи `gertrude` и `henry` имеют право не только читать данные, но и записывать их.

Организация сервера печати с помощью Samba

Разделяемые объекты, представляющие принтеры, используются по такому же принципу, как и объекты, предназначенные для организации совместного доступа к файлам. Для того чтобы вывести данные на печать, клиент передает файл разделяемому объекту принтера. Сервер обрабатывает этот файл и выводит его содержимое на устройство печати. Многие из параметров, которые используются при описании разделяемых объектов файлов, применимы и для объектов принтеров. Так, например, в описания разделяемых объектов принтеров можно включать параметры, управляющие доступом пользователей. С другой стороны, в определении таких объектов неуместны параметры, которые управляют использованием в именах файлов символов верхнего и нижнего регистра.

Трудности, возникающие при поддержке разделяемых принтеров, в основном связаны с обработкой заданий на печать, передаваемых Windows-клиентами. В зависимости от типа очереди Linux, такие задания должны либо модифицироваться, либо оставаться неизменными. В некоторых случаях приходится создавать различные конфигурации для PostScript-принтеров и принтеров, в которых отсутствует поддержка данного языка.

При рассмотрении вопросов взаимодействия с устройством печати предполагается, что принтер уже подключен к компьютеру и может выводить данные, которые передаются ему в системе Linux. (Для некоторых принтеров отсутствуют драйверы Ghostscript, но эти принтеры могут успешно использоваться посредством Samba.)

Создание разделяемого объекта принтера

Основное различие между разделяемыми объектами файлов и принтеров состоит в том, что в описании последнего присутствует параметр `printable = Yes` или `print ok = Yes` (эти параметры являются синонимами). Каталог, указанный в определении

объекта, представляет собой временный каталог спулинга (он не должен совпадать с каталогом спулинга системы Linux, в качестве которого обычно используется /var/spool/lpd). По умолчанию принимается каталог /tmp, но в некоторых дистрибутивных пакетах Linux для этой цели создается каталог /var/spool/samba. При определении прав доступа для этого каталога должен быть установлен бит, запрещающий пользователям удалять из каталога файлы, которые были созданы другими пользователями. Как правило, к каталогу спулинга применяется команда chmod 1777 */каталог* или chmod o+t */каталог*. (Значение 1777 позволяет всем пользователям записывать данные в каталог. Такие права доступа рекомендованы для каталогов, используемых для поддержки спулинга.) Один и тот же каталог может применяться для организации нескольких очередей к принтерам. Ниже приведен пример разделяемого объекта принтера.

```
[laser]
    comment = Laser printer in Room 7
    path = /var/spool/samba
    printable = Yes
```

Параметр comment задает комментарии для разделяемого объекта LASER. (Этот параметр также может быть использован в разделяемых объектах файлов.) Данное описание можно применять только в том случае, если в системе имеется локальная очередь печати с именем laser. Если имя очереди отличается от имени разделяемого объекта, то надо указать его с помощью параметра name. Так, например, параметр name = lp сообщает системе о том, что необходимо использовать локальную очередь с именем lp.

В различных дистрибутивных пакетах Linux используются разные системы печати. В настоящее время чаще всего применяется система BSD, но системы LPRng и CUPS (Common Unix Printing System — общая система печати UNIX) также приобретают популярность. Разные системы печати отличаются друг от друга синтаксическими правилами, что необходимо учитывать при настройке Samba. Для этой цели предусмотрен параметр printing, который позволяет задавать систему печати вашего компьютера. Этот параметр чаще всего принимает значения BSD, LPRng или CUPS (другие значения в системе Linux задаются крайне редко). Если пакет Samba входит в состав дистрибутивного пакета Linux, то, вероятнее всего, он уже сконфигурирован для работы с соответствующей системой печати. Если в вашей версии Linux используется устаревшая система печати, вы можете применить параметр print command, который позволяет согласовать команды печати, используемые Samba с вашей системой. При этом переменной %s присваивается имя файла, указанное в задании на печать. После передачи файла системе печати его следует удалить. Параметр print command обеспечивает дополнительную степень гибкости и позволяет использовать Samba для решения специфических задач.

Выше в этой главе рассматривался вопрос создания разделяемого объекта файлов, обеспечивающего доступ всех пользователей к своим рабочим каталогам. Точно так же при работе с разделяемыми объектами принтеров вы можете создать один объект, обеспечивающий доступ ко всем принтерам, присутствующим в системе. Этот объект имеет имя [printers]. При наличии объекта с таким именем Samba просматривает содержимое файла /etc/printcap и для каждого принтера создает свой разделяемый объект. Подобно объекту [homes], объект [printers] обычно содержит параметр browseable = No, в результате чего разделяемый объект PRINTERS не отображается клиент-программами Windows. Если для этого параметра установлено значение Yes, в списке просмотра будут отображаться конкретные имена принтеров.

Совместное использование PostScript-принтеров

В ходе предыдущего обсуждения не затрагивался вопрос об использовании драйверов. Этот вопрос чрезвычайно важен для разделения принтеров в системе Samba; драйверы принтеров часто становятся источником проблем. В системе Windows драйверы принтеров взаимодействуют с прикладными программами и генерируют данные, совместимые с конкретной моделью принтера. Таким образом, формат файла, который приходит разделяемому объекту печати, расположенному на сервере Samba, зависит от того, какой драйвер принтера был установлен на клиентской машине. В отличие от Windows, в системе Linux обычно используются очереди печати, которые настраиваются для приема данных в формате PostScript. В зависимости от типа принтера, подключенного к компьютеру, данные могут передаваться ему в неизменном виде либо обрабатываться фильтром печати. Фильтр печати преобразует PostScript-код в формат, подходящий для данного принтера. Различия в моделях печати иногда приводят к возникновению конфликтов. Если клиент Windows предоставляет серверу Samba файл в формате, отличном от PostScript, информация, предназначенная для вывода на печать, может быть искажена. Возможна также ситуация, когда данные, сгенерированные PostScript-драйвером Windows, воспринимаются на сервере Samba как информация, представленная в другом формате. В этом случае при выводе на печать также будут получены некорректные результаты. Существует ряд правил и параметров Samba, которые позволяют разрешить данную проблему, но необходимо испробовать различные средства, чтобы найти наиболее подходящие из них.

Наиболее просто конфигурация Samba устанавливается в том случае, если разделяемый принтер поддерживает язык PostScript. В этом случае PostScript-код может быть непосредственно передан на устройство печати. Однако поддержка PostScript обычно реализуется в высокоуровневых лазерных принтерах и некоторых устройствах печати среднего уровня. Низкоуровневые лазерные принтеры и струйные принтеры, поддерживающие PostScript, встречаются крайне редко. Для того чтобы выяснить, может ли ваш принтер обрабатывать PostScript-код, необходимо ознакомиться с документацией на это устройство.

ВНИМАНИЕ Некоторые принтеры объявляются в системе как устройства, поддерживающие PostScript, однако данное свойство принтера эмулируется с помощью интерпретатора, выполняющегося в системе Windows. При организации совместного доступа к подобным принтерам они должны рассматриваться как устройства печати, не поддерживающие PostScript. Это могут быть высококачественные принтеры, но без эмулятора они не способны обрабатывать PostScript-код.

Язык PostScript и соответствующий интерпретатор были первоначально разработаны компанией Adobe (`http://www.adobe.com`). Многие компании, занимающиеся выпуском принтеров, также разработали свои версии интерпретаторов PostScript для включения в состав устройств печати. Так, поддержка PostScript реализована во многих принтерах Hewlett Packard и Lexmark. Первые интерпретаторы PostScript были далеки от совершенства, но присущие им ошибки были исправлены в более поздних версиях. В настоящее время при установке конфигурации Samba совершенно не важно, поддерживает ли ваш принтер версию PostScript, разработанную Adobe, или в нем используется один из интерпретаторов, выпущенных другими компаниями.

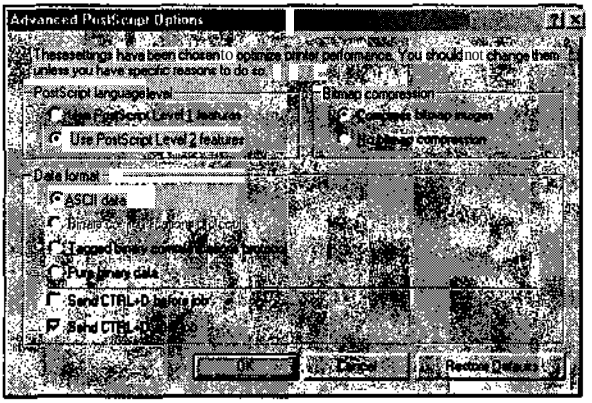


Рис. 7.4. Если наличие <Ctrl+D> приводит к возникновению проблем при выводе документа на принтер в системе Linux, вы можете запретить генерацию этого символа

Если ваш принтер непосредственно поддерживает PostScript, вы можете установить PostScript-драйвер в системе Windows. Такие драйверы входят в поставку операционной системы, а также распространяются производителями принтеров. (Компания Adobe также предоставляет драйверы PostScript, но они предназначены для принтеров, в которых используется PostScript-интерпретатор, разработанный Adobe.) Этот драйвер генерирует данные в формате PostScript, которые Samba передает в очередь на устройство печати; затем данные приходят на конкретный принтер. В большинстве случаев информация корректно выводится на печать.

Иногда при использовании описанной конфигурации возникают проблемы, которые чаще всего связаны с тем, что PostScript-драйвер, установленный в системе Windows, предваряет вывод данных символом <Ctrl+D>. Большинство PostScript-принтеров игнорирует этот символ, но он может влиять на поведение очереди в системе Linux. Фильтр печати ожидает встретить в начале файла, содержащего задание на печать, идентификатор формата PostScript. Обнаружив символ <Ctrl+D>, фильтр печати интерпретирует его как признак того, что в файле содержится ASCII-текст. Поскольку ASCII-текст не может быть непосредственно передан на PostScript-принтер, этот файл повторно преобразуется в PostScript-формат. В результате вместо текста документа на принтер выводится его PostScript-код. Разрешить эту проблему можно двумя способами.

Вы можете отключить опцию драйвера печати Windows, которая управляет выводом символа <Ctrl+D>. Эту опцию можно найти в диалоговом окне Properties принтера либо в окне, которое отображается после щелчка на кнопке Advanced в этом же окне необходимо сбросить флажок опции Send CTRL+D Before Job. (Опция Send CTRL+D After Job обычно не приводит к возникновению проблем.) Такое решение хорошо подходит в том случае, если вы используете одну очередь для обслуживания PostScript-принтеров и устройств печати, поддерживающих другие языки описания документов. Недостаток подобного подхода состоит в том, что для настройки многочисленных Windows-клиентов придется затратить много времени.

Чтобы разрешить проблему, возникающую из-за наличия в начале задания на печать символа <Ctrl+D>, необходимо установить параметр `postscript = Yes`. При этом

Samba будет непосредственно добавлять идентификатор PostScript-кода в начало задания. Получив такие данные, принтер проигнорирует как второй идентификатор PostScript, так и символ <Ctrl+D>. Данное решение хорошо подходит в том случае, если в сети присутствует большое количество Windows-клиентов. Если же вы захотите выводить задания на печать, сгенерированные драйверами другого типа, вам придется формировать вторую очередь, а это приведет к усложнению списка разделяемых объектов.

Совместное использование принтеров, не поддерживающих PostScript

Существуют два способа настройки Samba для работы с принтерами, не поддерживающими PostScript. Первый способ заключается в использовании PostScript-драйвера на клиентской машине и настройке очереди печати Linux для преобразования PostScript-данных в формат, поддерживаемый конкретным принтером. Согласно второму способу, следует установить на клиентском компьютере драйвер, генерирующий данные на языке принтера, а в системе Linux создать *очередь без обработки* (raw queue). Данные, помещенные в такую очередь, передаются на принтер в неизменном виде. Оба способа имеют свои преимущества и недостатки, и для реализации каждого из них необходимо определенным образом сконфигурировать Samba.

Использование Ghostscript

Предположим, что вам необходимо вывести на принтер, не поддерживающий PostScript, сгенерированные Linux-приложением PostScript-данные. В этом случае система должна быть сконфигурирована так, чтобы данные, предназначенные для печати, обрабатывались программой Ghostscript. Ghostscript (`http://www.cs.wisc.edu/~ghost/`) — это PostScript-интерпретатор, выполняющийся не на принтере, а на компьютере. GNU-версия Ghostscript, ориентированная на работу с различными типами принтеров, поставляется в составе большинства дистрибутивных пакетов Linux. Если какой-то из принтеров не поддерживается данной реализацией Ghostscript, можно применить специальные драйверы либо воспользоваться более современным продуктом Ghostscript производства Aladdin. Информацию о совместимости различных принтеров с интерпретаторами Ghostscript можно найти, обратившись по адресу `http://www.linuxprinting.org/printer_list.cgi`.

В очереди печати Linux, настроенной для совместной работы с Ghostscript, присутствует фильтр печати, который распознает тип файла. В отличие от фильтра, который используется в очереди, обслуживающей PostScript-принтер, данный фильтр передает данные на вход программы Ghostscript. В этом случае необходимо использовать инструментальные средства настройки принтеров, поставляемые в составе дистрибутивного пакета, а также следовать инструкциям по установке конфигурации фильтра. Настроенная подобным образом очередь работает почти идентично очереди PostScript-принтера (вопросы выбора конфигурации Samba и клиентов для такой очереди рассматривались в предыдущем разделе). Для Windows-клиентов надо выбрать универсальный PostScript-драйвер (для лазерных принтеров хорошо подходят драйверы Apple LaserWriter, а для струйных — драйвер QMS magicolor). Проблема, связанная с появлением в составе задания на печать символа <Ctrl+D>, решается так же, как и при использовании PostScript-принтеров.

Некоторые PostScript-драйверы в системе Windows включают в состав PostScript-файлов дополнительные команды, предназначенные для вывода информации на встроенные дисплеи принтеров. Такие команды часто приводят к появлению дополнительных страниц выходных данных с сообщениями типа %% [LastPage] %%. Если вы встретитесь с подобной проблемой, решить ее можно, сменив драйвер в системе Windows. Существует и другое решение. Вам надо найти файл, из которого вызывается интерпретатор Ghostscript, и добавить в строку, содержащую команду gs, последовательность символов `>/dev/null`. В результате сообщения с используемого в обычных условиях выходного устройства Ghostscript будут перенаправлены в файл `/dev/null`. В системе Caldera таким файлом является `/var/spool/lpd/`*`имя_очереди`*`/printfilter`. В Red Hat, Mandrake и TurboLinux используется файл `/usr/lib/rhs/rhs-printfilters/ps-to-printer.fpi`.

Создание очереди, не обрабатывающей PostScript-данные

Если вы хотите использовать на клиентской машине драйвер, генерирующий информацию в формате, отличном от PostScript, вам надо создать в системе Linux очередь принтера, в которой не предпринимались бы попытки модифицировать данные, переданные на печать. Некоторые фильтры Linux распознают ряд языков и передают соответствующие данные принтеру в неизменном виде, поэтому не исключено, что вам удастся использовать обычную очередь печати. Если же задания, помещенные в такую очередь, исчезают или выводятся в искаженном виде, вам придется создать очередь без обработки (raw queue).

Для создания очереди без обработки надо сформировать обычную очередь печати, а затем внести изменения в файл `/etc/printcap` (при условии, что на вашем компьютере используется система печати BSD или LPRng). В частности, вам надо удалить из описания очереди строку `if=` либо задать пустое значение if. Эта строка определяет фильтр печати Linux, и ее удаление приведет к тому, что задание будет передаваться из очереди на принтер в неизменном виде. Пример описания очереди приведен ниже.

```
lp|hp4000|raw:\
        :lp=/dev/lp0:\
        :sd=/var/spool/lpd/lp:\
        :mx#0:\
        :sh:\
        :if=:
```

В данном примере указаны три имени принтера: lp, `hp4000` и raw. Данные из этой очереди выводятся на устройство печати `/dev/lp0`, а в качестве каталога спулера используется `/var/spool/lpd/lp`. (Заметьте, что указанный здесь каталог отличается от каталога спулера Samba. Файл сначала располагается в каталоге спулера Samba, а затем перемещается в `/var/spool/lpd/lp`.) Опция `тх#0` снимает ограничения на размер файла печати, а `sh` запрещает вывод страницы заголовка. Поскольку в строке `if=` не указан фильтр печати, данные передаются в неизменном виде.

Определяя разделяемый объект принтера для очереди без обработки, необходимо убедиться, что параметр `postscript` отсутствует. Если же этот параметр указан, для него должно быть установлено значение No. Идентификатор PostScript в составе задания на печать скорее всего нарушит работу принтера, не поддерживающего данный язык.

Выбор способа поддержки принтера

Если к компьютеру, на котором выполняется сервер Samba, подключен принтер, не поддерживающий PostScript, вам необходимо принять решение о том, нужно ли установить на клиентской машине PostScript-драйвер или драйвер, ориентированный на работу с имеющимся у вас устройством печати. Следует заметить, что рассмотренные выше варианты решения не являются взаимоисключающими, как это кажется на первый взгляд. Вы можете создать две очереди, а если фильтр печати распознает задания, предназначенные для принтера, не поддерживающего PostScript, можно даже обойтись одной очередью печати. (Если вы создадите две очереди Linux, то разделяемый объект `[printers]` обнаружит и будет использовать их.) Если вы поступите таким образом, вам придется установить два драйвера на клиентской машине и предоставить пользователю право выбирать нужный драйвер.

Одно из важных различий между рассмотренными подходами состоит в том, что текст или изображение преобразуется в битовую карту на разных этапах обработки. Рассмотрим вывод на печать документа, содержащего в основном текстовые данные. Если вы используете Ghostscript в системе Linux и PostScript-драйвер на стороне клиента, клиент генерирует текстовый файл. Размеры этого файла относительно невелики, поэтому создание такого файла не создает существенной нагрузки на процессор клиентской машины. Трафик, связанный с передачей этого файла по сети, также невелик. Для обработки такого файла в системе Linux требуется большой объем ресурсов, так как содержащиеся в файле данные должны быть преобразованы в битовую карту. Если на клиентском компьютере установлен драйвер целевого принтера, основную нагрузку по обработке файла будет нести процессор этой машины, поскольку именно на стороне клиента будет осуществляться преобразование в формат битовой карты. Соответственно возрастет трафик сети, потому что размер передаваемого файла увеличится. Нагрузка на процессор компьютера, на котором выполняется сервер Samba, будет небольшой, поскольку основная обработка данных выполняется на стороне клиента. Таким образом, при использовании интерпретатора Ghostscript в работу вовлекается меньше ресурсов клиентской машины и уменьшается трафик сети. Применение драйвера целевого принтера оправдано в тех случаях, когда необходимо уменьшить нагрузку на процессор сервера. При выводе на печать графических файлов разница между различными подходами практически не ощущается, так как размеры файла почти не зависят от того, передается ли он в формате PostScript или в формате целевого принтера.

Еще одно различие между вариантами поддержки принтеров, не обрабатывающих PostScript-данные, связано с качеством выходных данных. Используя Ghostscript, вы полагаетесь на реализованные в этом продукте средства генерации изображений. Чаще всего Ghostscript хорошо справляется с этой задачей, но в некоторых случаях драйверы целевых принтеров генерируют изображения гораздо лучшего качества. В особенности это бывает заметно при работе с последними моделями струйных принтеров. Некоторые принтеры вовсе не поддерживаются Ghostscript; для такого устройства возможна лишь одна конфигурация, предполагающая использование драйвера этого принтера на стороне клиента и очереди без обработки в системе Linux. Бывают ситуации, в которых интерпретатор Ghostscript предпочтительнее драйвера целевого принтера. Это случается при работе с приложениями, непосредственно ориентированными на работу с PostScript-принтерами (например, пакетами, предназначенными для организации настольных издательских систем), или при выводе на печать файлов EPS (Encapsulated PostScript). Необходимо

заметить, что интерпретатор Ghostscript позволяет обеспечить PostScript-совместимость, затратив для этого минимальные усилия. Преимуществом использования Ghostscript также является тот факт, что данный интерпретатор обеспечивает реальную возможность стандартизировать вывод на печать, т. е. исключает необходимость переключаться между различными принтерами.

С проблемой качества изображения непосредственно связана проблема обеспечения необходимого уровня гибкости при работе с драйверами. Используя Ghostscript, вы устанавливаете разрешение и другие параметры принтера с помощью опций интерпретатора. В системе Linux эти опции обычно задаются при создании очереди. Для того чтобы выводить данные с различным разрешением, необходимо сформировать несколько очередей и переключаться между ними. Если на клиентской машине установлен драйвер целевого принтера, пользователь может выбирать разрешение и управлять другими характеристиками печати с помощью опций этого драйвера. Так, например, организуя работу со струйными принтерами, гораздо удобнее устанавливать драйверы этих принтеров на стороне клиента.

Таким образом, на вопрос о том, следует ли использовать Ghostscript или драйвер целевого принтера, невозможно дать однозначный ответ. Выбирая конфигурацию системы, следует учесть специфику решаемых задач и характеристики вашей сети. При желании вы можете также поэкспериментировать с обеими конфигурациями и на практике определить, какой подход дает лучшие результаты в вашей сети и с вашими принтерами. Скорее всего, вы убедитесь в том, что для одних принтеров целесообразно использовать Ghostscript, а для других лучше установить драйверы этих принтеров на клиентском компьютере.

Сценарии Samba

Одна из самых привлекательных особенностей Samba — возможность выполнения сценариев. Вы можете задавать команды, которые будут выполнены при наступлении определенных событий. Благодаря поддержке сценариев Samba можно использовать для решения задач, непосредственно не относящихся к совместному использованию файлов и принтеров. В начале данного раздела мы обсудим сценарии `preexec` и `postexec` и псевдопринтеры, а в заключение рассмотрим примеры использования этих средств.

Сценарии preexec и postexec

Samba поддерживает параметры `preexec` и `postexec`, которые позволяют выполнять некоторые команды при регистрации пользователя и завершении его работы с разделяемым объектом. В качестве значения параметра `preexec` задаются команды, которые должны быть выполнены при регистрации пользователя, соответственно команда, указанная как значение `postexec`, выполняется при завершении работы пользователя с объектом. Например, если вы хотите, чтобы при обращении к разделяемому объекту сервер Samba передавал почтовое сообщение по адресу `billy@harding.threeroomco.com`, вы должны включить в определение этого объекта следующее выражение:

```
preexec = mail -s "Share being used" \
    billy@harding.threeroomco.com
```

Если пользователь зарегистрируется для работы с объектом, Samba пошлет от его имени сообщение по адресу billy@harding.threeroomco.com. В поле Subject сообщения будет включена строка "Share being used", а по адресу отправителя получатель сможет выяснить, кто из пользователей работал с объектом.

Аналогично действует параметр postexec, но команда, заданная в качестве его значения, выполняется после окончания работы с объектом. Зная особенности работы Windows-клиентов с разделяемыми объектами SMB/CIFS, можно сделать вывод, что команда не будет выполнена сразу же после того, как пользователь закроет окно, открытое с помощью Network Neighborhood или My Network Places, но через некоторое время это обязательно произойдет.

Разновидностями параметров preexec и postexec являются параметры root preexec и root postexec. Отличаются они лишь тем, что команды, заданные в качестве значений root preexec и root postexec, выполняются от имени пользователя root. Таким образом, можно задавать команды, для выполнения которых требуются специальные привилегии. Используя эти параметры, следует соблюдать осторожность. Если вы допустите ошибку, у вас могут возникнуть проблемы, связанные с безопасностью системы.

При выполнении сценариев сервер Samba может обрабатывать переменные, перечисленные в табл. 7.1. Эти переменные позволяют настроить сценарии preexec и postexec для работы с конкретными пользователями, клиентами, операционными системами, установленными на клиентских компьютерах, и т. д. (Некоторые из переменных, представленных в табл. 7.1, специально предназначены для использования в разделяемых объектах принтеров.)

Параметры preexec и postexec в основном предназначены для того, чтобы задавать команды, подготавливающие разделяемые объекты к использованию. Так, например, если есть опасность, что пользователь, работающий в системе Windows, по ошибке удалит конфигурационный файл Linux, сценарий preexec можно использовать для создания резервной копии этого файла. Кроме того, параметры preexec и postexec применяются для решения самых разнообразных задач; некоторые из них описаны ниже.

- С помощью сценариев preexec и postexec можно создавать и удалять символьные ссылки между совместно используемыми каталогами и рабочим каталогом пользователя. (По умолчанию Samba следует символьным ссылкам, но поведение системы можно изменить, установив параметр follow symlinks = No.)

- С помощью параметра preexec можно монтировать заменяемые носители в устройствах, соответственно сценарии postexec могут быть использованы для их размонтирования. Такая возможность очень полезна при работе с гибкими дисками, устройствами чтения компакт-дисков и другим оборудованием.

- Вы можете записывать в файлы протоколов различные данные, в том числе информацию, которую Samba в обычных условиях не регистрирует.

- При необходимости вы можете сформировать разделяемый объект и создать сценарии, которые будут преобразовывать графические файлы, находящиеся в соответствующем каталоге, в другие форматы и передавать преобразованные файлы другим объектам.

Таблица 7.1. Переменные, доступные в системе Samba

Переменная	Назначение
%a	Операционная система на клиентском компьютере. Возможные значения: OS2 (OS/2), Samba, UNKNOWN, WfWg (DOS или Windows for Workgroups), Win2K, Win95 (Windows 95 или 98) и WinNT
%d	Идентификатор процесса сервера
%g	Основная группа, к которой относится пользователь, указанный в переменной %u
%G	Основная группа, к которой относится пользователь, указанный в переменной %U
%h	Доменное имя сервера (в домене TCP/IP)
%H	Рабочий каталог пользователя, информация о котором содержится в переменной %и
%I	IP-адрес клиента
%j	Номер задания на печать
%L	NetBIOS-имя сервера
%m	NetBIOS-имя клиента
%M	Доменное имя клиента (в домене TCP/IP)
%N	Сервер NIS
%p	Путь к каталогу, связанному с разделяемым объектом, используемый при автомонтировании
%P	Путь к каталогу, связанному с разделяемым объектом
%R	Уровень протокола SMB/CIFS. Возможные значения: CORE, COREPLUS, LANMAN1, LANMAN2 и NT1
%s	Имя файла, переданного разделяемому объекту принтера
%S	Имя разделяемого объекта
%T	Текущая дата и время
%и	Эффективное имя пользователя UNIX
%U	Имя пользователя, зарегистрированного в системе UNIX (может не совпадать с именем, хранящимся в переменной %u)
%v	Номер версии Samba

- Разделяемые объекты с ограниченным доступом можно использовать при выполнении задач администрирования. Возможна конфигурация, при которой этот объект содержит копии файлов из каталога /etc, а сценарий, заданный с помощью параметра `postexec`, просматривает копии файлов и в случае обнаружения изменений копирует их в каталог /etc и перезапускает компьютер.

- Чтобы уменьшить риск потери информации, необходимо периодически создавать резервные копии данных. Разделяемый объект можно использовать для создания процедуры копирования, запускаемой по инициативе пользователя. Для этого надо создать Windows-сценарий, который открывал бы разделяемый объект Samba, копировал все файлы с компьютера в этот объект и закрывал его. Для решения данной задачи следует также построить сценарий разделяемого объекта, осуществляющий копирование данных на резервный носитель. При наличии такого объекта

и сценариев пользователю остается установить носитель на устройство, запустить сценарий Windows и ожидать завершения копирования данных.

Некоторые из описанных примеров реализуются чаще других. Ряд задач (например, задачи администрирования) создают потенциальную угрозу безопасности системы, поэтому соответствующие сценарии находят лишь ограниченное применение. Поэтому приведенные выше примеры — это не описания задач, которые должны быть решены с помощью сценариев `preexec` и `postexec`, а лишь демонстрация возможностей этих сценариев.

Несмотря на то что вы можете задавать необходимые команды непосредственно в составе параметров `preexec` и `postexec`, гораздо удобнее оформлять команды в виде сценария оболочки и задавать сценарий в качестве значения соответствующего параметра. Это позволит вам выполнять в ответ на действия пользователей сколь угодно сложные операции.

В некоторых случаях возникает необходимость ограничить число пользователей, которые могут одновременно обращаться к разделяемому объекту. Это можно сделать с помощью параметра max connections. Чтобы полностью исключить одновременные действия пользователей, надо задать параметр max `connections = 1`. Однако такая конфигурация иногда создает нежелательные побочные эффекты, так как при обращении к разделяемым объектам с помощью Network Neighborhood и My Network Places соединения с разделяемыми объектами закрываются с некоторым опозданием.

Использование псевдопринтеров

Еще одну возможность использования сценариев предоставляет параметр `print` command, предназначенный для включения в описание разделяемого объекта печати. Первоначально этот параметр создавался для осуществления операций, связанных с передачей задачи на печать, но в качестве его значения можно задавать любые команды. Параметр `print` command позволяет выполнить специальную обработку PostScript-файлов и реализовать эффекты, имеющие лишь отдаленное отношение к выводу данных на печать. Этот параметр можно применять для обработки любых данных, содержащихся в файле, сгенерированном в системе Windows. Ниже приведены примеры задач, решаемых с помощью параметра `print` command.

- Передача факсов с помощью программного обеспечения Linux и PostScript-драйвера Windows. При этом даже можно воспользоваться Windows-программами, например, продуктом Respond (`http://www.boerde.de/~horstf/`) для создания интерфейса.

- Конвертирование PostScript-файлов в другие типы данных, например, представление их в виде PDF-файлов или преобразование в графический формат. Пример подобного использования параметра `print` command рассматривается ниже в этой главе.

- Непосредственный вывод на печать данных в формате, отличном от PostScript. Некоторые из таких данных корректно обрабатываются с помощью фильтров печати Linux, для них необходимость в создании специальной конфигурации не возникает, однако в ряде случаев применение `print` command вполне оправдано. Например,

вы можете использовать данный параметр для обработки файлов, созданных с помощью текстового процессора, извлечения определенных полей из базы данных и выполнения других подобных действий.

- Объединяя данные в файл архива и передавая этот файл на компьютер под управлением Linux, вы можете решать задачи, подобные тем, которые решаются с помощью сценариев `preexec` и `postexec`. Вы можете создавать резервные копии файлов, преобразовывать форматы и выполнять другие действия. Пример решения подобной задачи приведен ниже.

Подобно сценариям `preexec` и `postexec`, параметр `print command` позволяет выполнять действия, которые могут создавать угрозу безопасности системы. При решении задач, предполагающих подобные действия, будьте внимательны, особенно если вы используете параметр `force user` для предоставления специальных привилегий.

В составе параметра `print command` можно использовать переменные, приведенные в табл. 7.1 (некоторые из них, например %s, специально предназначены для такого применения, и их появление в составе сценариев `preexec` и `postexec` не оправдано). Переменная %H в особенности полезна при доставке данных пользователю, инициировавшему задачу; в частности, вы можете использовать эту переменную для указания пути к каталогу, в который должны быть помещены файлы после окончания обработки.

Параметр `print command` имеет существенное преимущество перед `preexec` и `postexec`. При взаимодействии с разделяемым объектом соединение может остаться открытым, в результате сценарий `postexec` долгое время не получит управления. Если же клиент инициирует задачу печати, команды, указанные в качестве значения `print command`, сразу же выполняются. Однако следует учесть, что при выполнении некоторых операций две последовательно переданные на печать задачи могут повредить друг другу файлы с данными.

Пример использования средств Linux для записи компакт-дисков

С помощью сценариев можно сконфигурировать сервер Samba как платформу для создания компакт-дисков. Предположим, что вы администрируете сеть небольшого офиса, к которой подключены десятки клиентских компьютеров. Предположим также, что в офисе имеется лишь одно устройство, позволяющее записывать компакт-диски. Время от времени у разных пользователей возникает необходимость записать информацию на компакт-диск. Вы можете подключить устройство записи к компьютеру, на котором выполняется сервер Samba, и предоставить всем желающим возможность использовать программное обеспечение Linux для создания компакт-дисков. Однако при этом возникнут проблемы. Во-первых, вам придется обучить пользователей работать с программами записи. Во-вторых, некоторые пользователи не будут удалять после окончания записи свои файлы, что приведет к нерациональному использованию дискового пространства. Сценарии Samba позволяют автоматизировать процесс записи и разрешить возникающие проблемы.

Использование сценариев preexec и postexec для записи компакт-дисков

Предположим, что вы решили выделить один из разделяемых объектов Samba для записи компакт-дисков. Этот объект не должен использоваться ни для каких других целей. Для создания компакт-диска с помощью сценариев preexec и postexec вам необходимо обеспечить выполнение следующих задач.

1. Удаление из разделяемого объекта всех файлов.
2. Получение файлов, предназначенных для записи на компакт-диск.
3. Создание образа диска с помощью mkisofs или другой подобной утилиты.
4. Запись образа на компакт-диск с помощью cdrecord или другой утилиты.
5. Удаление образа диска и файлов, из которых он был создан.

Описание разделяемого объекта, предназначенного для решения данных задач, выглядит следующим образом:

```
[cd-create]
    path = /home/samba/cd-create
    create mask = 0666
    directory mask = 0777
    read only = No
    max connections = 1
    preexec = /bin/rm -r %P/*
    postexec = /usr/local/bin/create-cd %H %P %U
```

Параметр preexec решает первую задачу. Вторая задача решается с помощью обычных операций Samba. Для решения задач 3-5 предназначен сценарий /usr/local/bin/create-cd, указанный в качестве значения второго параметра. Код этого сценария приведен в листинге 7.1.

Листинг 7.1. Сценарий, предназначенный для записи компакт-диска

```
#!/bin/sh
# $1 - Рабочий каталог пользователя, выполняющего запись на диск
# $2 - Каталог с исходными файлами
# $3 - Имя пользователя, выполняющего запись на диск
mkisofs -J -r -o $1/image.iso $2
cdrecord speed=2 dev=4,0 $1/image.iso
mail -s "CD-R creation finished" $3
rm $1/image.iso
rm -r $2/*
```

Для создания описанного выше разделяемого объекта выполните следующие действия.

- Создайте сценарий create-cd и сохраните его в каталоге /usr/local/bin. Для файла, содержащего сценарий, надо установить права, позволяющие запускать

его на выполнение (это можно сделать с помощью команды chmod a+x /usr/local/bin/create-cd). Опции утилит mkisofs и cdrecord необходимо выбрать в соответствии с характеристиками вашего устройства записи.

- Создайте разделяемый объект Samba с именем [cd-create]. При желании вы можете задать каталог, отличный от того, который был указан выше, но следите за тем, чтобы права доступа, установленные для него, позволяли всем пользователям читать и записывать данные.

- Установите признак SUID для исполняемой программы cdrecord. Для этого можно использовать команду chmod a+s /usr/bin/cdrecord. В некоторых дистрибутивных пакетах для организации доступа к данной утилите создана специальная группа. Вы можете включить в эту группу пользователей, которым необходимо записывать компакт-диски, либо использовать параметр force group. Можно поступить и по-другому: заменить в определении объекта [cd-create] параметр postexec на root postexec. Необходимо лишь обеспечить, чтобы сценарий create-cd выполнялся с привилегиями, достаточными для успешного запуска утилиты cdrecord.

После выполнения описанных выше действий вы можете использовать созданный разделяемый объект. Работая в системе Windows, можно смонтировать этот объект с помощью Network Neighborhood или My Network Places. Для этого надо щелкнуть правой кнопкой мыши на имени объекта и выбрать в контекстном меню пункт Map Network Drive. Затем следует связать разделяемый объект с именем устройства. Монтирования каталога пользователь должен скопировать файлы, предназначенные для записи на компакт-диск, на сервер Samba. Он может перемещать файлы в пределах каталога, копировать, удалять их и выполнять другие подобные действия. Когда пользователь будет готов начать запись, ему следует вставить чистый диск в устройство и размонтировать разделяемый объект, щелкнув правой кнопкой мыши на соответствующем ему пункте в окне My Computer и выбрав в контекстном меню пункт Disconnect.

К сожалению, при активизации этого пункта меню Windows может не разорвать соединение с объектом. В этом случае придется завершить сеанс работы или (при использовании Windows 9x/Me) перезагрузить компьютер. Через некоторое время (от нескольких секунд до нескольких минут) начнется запись на компакт-диск, по завершении которой пользователю, инициировавшему данную задачу, будет передано почтовое сообщение. Получив сообщение, пользователь может извлечь диск из устройства и проверить качество записи на своей машине.

Определение разделяемого объекта и код сценария, приведенные в данном примере, далеки от совершенства. В сценарии не приняты меры, запрещающие одновременное обращение к разделяемому объекту двух пользователей. Поэтому, если пользователь предпримет попытку начать запись до того, как другой пользователь извлечет свой диск из устройства, неминуемо возникнет проблема. Кроме того, сценарий не оповещает пользователя об ошибках. Например, если образ диска слишком велик и не может быть записан на имеющийся носитель, пользователь узнает об этом лишь тогда, когда попытается прочесть записанные данные. Более совершенный сценарий должен сообщать о возникающих проблемах или устранять их самостоятельно. Наконец, следует заметить, что различные версии Samba по-разному интерпретируют переменную %P, поэтому описание разделяемого объекта необходимо изменять в зависимости от конкретных условий работы.

Использование псевдопринтера для записи компакт-дисков

Механизм псевдопринтеров позволяет записывать компакт-диски способом, более удобным для пользователей Windows 9x/Me, однако применение данного средства не так очевидно, как действия, основанные на использовании разделяемого объекта файлов. Данный подход заключается в следующем. Windows-клиент передает серверу Samba zip-архив, который содержит файлы, предназначенные для записи на компакт-диск. Разделяемый объект вызывает сценарий, который распаковывает архив, и записывает извлеченные из архива файлы на компакт-диск. Данный сценарий представляет собой разновидность сценария create-cd. Описание разделяемого объекта выглядит следующим образом:

```
[cd-print]
    path = /var/spool/samba
    printable = Yes
    print command = /usr/local/bin/print-cd %H %s %U %P; rm %s
```

Как и в предыдущем примере, вам необходимо уточнить особенности обработки переменной %P в вашей версии Samba. Возможно, удобнее будет заменить эту переменную значением /var/spool/samba. Основную часть работы по записи компакт-диска выполняет сценарий, код которого представлен в листинге 7.2.

Листинг 7.2. Сценарий для записи компакт-диска с помощью параметра print command

```
#!/bin/sh
# $1 - Рабочий каталог пользователя, выполняющего запись на диск
# $2 - Имя zip-файла
# $3 - Имя пользователя, выполняющего запись на диск
# $4 - Путь к zip-файлу
mkdir -p $1/cdr/samba
cd $1/cdr/samba
unzip $4/$2
mkisofs -J -r -o $1/image.iso ./
cdrecord speed=2 dev=4,0 $1/image.iso
mail -s "CD-R creation finished" $3
rm $1/image.iso
rm -r $1/cdr/samba
```

Сценарий и разделяемый объект, используемые в данном примере, надо сконфигурировать так же, как это было сделано для объекта [cd-create] и сценария create-cd. Файл, содержащий сценарий, должен быть определен как исполняемый, опции утилит mkisofs и cdrecord необходимо привести в соответствие с конфигурацией вашей системы, а для утилиты cdrecord надо установить признак SUID, чтобы она выполнялась с правами root. Для записи компакт-диска необходимо передать zip-файл разделяемому объекту, используя для этого команду COPY системы DOS или Windows.

```
C:\> COPY FILE.ZIP\\SERVER\CD-PRINT
```

В результате выполнения данной команды содержимое файла FILE.ZIP будет записано на компакт диск. Очевидно, что вместо SERVER при вызове команды должно быть

указано имя конкретного сервера. Эту команду следует поместить в файл . BAT; имя zip-файла будет передаваться ей с помощью переменной.

```
COPY %1 \\SERVER\CD-PRINT
```

При вызове файла . BAT надо указать файл архива. Так, если файл, содержащий команду копирования, имеет имя `MAKECD.BAT`, то для его вызова используется команда `MAKECD FILE.ZIP`. Если вы создадите на рабочем столе ярлык, представляющий файл . BAT, то для записи компакт-диска достаточно будет перетащить файл архива на пиктограмму файла . BAT. В состав файла . BAT можно также включить вызов утилиты архивирования файлов. В этом случае, чтобы записать диск, пользователь должен будет собрать все необходимые ему файлы в одном каталоге и перетащить этот каталог на пиктограмму файла . BAT.

Как и при использовании разделяемого объекта файлов, решение, реализованное в данном примере, имеет ряд недостатков. При выполнении сценария не проверяется размер образа диска и не принимаются меры, препятствующие одновременному обращению двух пользователей к разделяемому объекту. Но сценарий, свободный от этих недостатков, был бы гораздо сложнее.

Пример создания PDF-файлов

В качестве примера использования очереди печати можно привести задачу преобразования входных PostScript-данных в PDF-файлы. Для ее решения надо создать очередь печати, подобную той, которая используется для обработки данных, сгенерированных с помощью PostScript-драйвера. Описание разделяемого объекта имеет следующий вид:

```
[pdf-create]
    comment = Create a PDF file
    path = /var/spool/samba
    printable = Yes
    print command = gs -dNOPAUSE -q -dBATCH -sDEVICE=pdfwrite \
    -sOutputFile=%H/%s.pdf %s; rm %s
```

Символ \, завершающий предпоследнюю строку, имеет специальное значение: он сообщает Samba о том, что в следующей строке находится продолжение текущей команды. Этот символ позволяет избежать появления в составе конфигурационного файла длинных строк и делает содержимое файла более удобным для чтения.

С помощью параметра `print command` вызывается исполняемый файл Ghostscript (gs). Опции -dNOPAUSE, -q и -dBATCH обеспечивают непрерывный вывод данных с минимальным набором специальных сообщений, не требующий вмешательства пользователя. Опция -sDEVICE=pdfwrite указывает на то, что в результате выполнения программы должны генерироваться PDF-файлы, а опция `-sOutputFile=%H/%s.pdf` формирует имена файлов, отличающиеся от имен заданий на печать только суффиксом `.pdf`. Сформированные PDF-файлы сохраняются в рабочем каталоге пользователя. Определение данного разделяемого объекта можно модифицировать так, чтобы PDF-файлы помещались в другой каталог или передавались пользователю в составе почтовых сообщений.

Резюме

Samba — сложный инструмент, позволяющий организовать совместное использование файлов и принтеров. Этот продукт позволяет добиться большой степени гибкости при решении конкретных задач. Если Samba поставляется в составе дистрибутивного пакета Linux, то сформированный по умолчанию конфигурационный файл обеспечивает работу данного продукта в сети. Изменять конфигурацию Samba приходится в тех случаях, когда необходимо реализовать более сложные функции (например, когда сервер Samba должен действовать как контроллер домена). В составе конфигурационного файла также приходится определять разделяемые объекты. Как правило, изменения, вносимые в состав конфигурационного файла, не сложные и легко могут быть выполнены администратором средней квалификации. Одной из привлекательных особенностей Samba является способность выполнять заданные команды при наступлении определенных событий. Благодаря такой возможности Samba можно использовать для решения задач, непосредственно не связанных с разделением файлов и принтеров.

Глава 8

Совместное использование файлов с помощью NFS

Протоколы Server Message Block (SMB)/Common Internet Filesystem (CIFS), рассмотренные в предыдущей главе, очень удобны для организации совместного доступа к файлам и принтерам клиентов, работающих под управлением DOS, Windows, OS/2 и многих других систем. Однако эти протоколы не поддерживают некоторые характеристики файловых систем UNIX и Linux, например, не позволяют задавать владельца файла и права доступа. Поэтому для разделения файлов в системах UNIX и Linux целесообразно использовать другой протокол, а именно NFS (Network Filesystem — сетевая файловая система). В отличие от SMB/CIFS, NFS не поддерживает принтеры. Вопросы совместного использования принтеров будут рассмотрены в главе 9.

Использование серверов NFS

Как правило, серверы NFS применяются для разделения файлов в системах UNIX и Linux. Необходимость в совместном доступе к файлам может возникнуть по разным причинам. Возможно, вы захотите хранить на сервере программы большого объема для того, чтобы их можно было запускать на клиентских машинах с дисками малого размера. Часто сервер NFS используют как централизованное хранилище файлов; изменения, внесенные в файл, сразу становятся доступными всем пользователям. Если в сети применяется централизованная система регистрации, например Kerberos, целесообразно размещать на сервере NFS рабочие каталоги пользователей. Такой подход обеспечивает высокую степень гибкости. В этом случае пользователь перестает быть "привязанным" к своей рабочей станции и может регистрироваться с любого компьютера и работать со своими данными. Существует множество других ситуаций, в которых оправдано применение серверов NFS. Например, вы можете разместить рабочие каталоги пользователей на локальных компьютерах, а сервер NFS использовать для обмена файлами или организовать чтение данных из статической базы, расположенной на сервере.

Несмотря на то что система NFS ориентирована на использование в сетях, состоящих в основном из компьютеров под управлением UNIX, клиенты и серверы NFS разработаны и для других систем, например для Windows, OS/2 и MacOS. Выбор инструмента

разделения файлов зависит от конкретной ситуации. В большинстве случаев наилучшие результаты достигаются при использовании в системе Linux протокола, специально разработанного для других систем. Примером подобного решения является организация сервера SMB/CIFS на компьютере под управлением Linux. Кроме того, для настройки сервера Samba в системе Linux потребуется намного меньше времени и усилий, чем для инсталляции и конфигурирования средств поддержки NFS на клиентских компьютерах. Если же в сети в основном используются машины под управлением UNIX и Linux и лишь несколько компьютеров Windows или MacOS, предпочтительнее использовать для разделения файлов систему NFS. (Система MacOS X базируется на UNIX, поэтому средства поддержки NFS хорошо работают в этой среде, однако для их настройки с помощью графического интерфейса MacOS придется затратить много усилий.)

ВНИМАНИЕ | Как вы узнаете из последующих разделов, в процессе работы NFS не поверяет пароли и не реализует другие подобные способы контроля доступа. Вместо этого NFS использует принцип доверия, согласно которому сервер полагается на средства аутентификации пользователей, применяемые на клиентских машинах. На сервере NFS вы определяете узлы, пользующиеся доверием, и задаете их IP-адреса. Данный механизм защиты не сложно обойти, используя фальшивый IP-адрес или изменяя конфигурацию локальных компьютеров, поэтому, планируя систему NFS, надо уделять особое внимание безопасности данных. В частности, нельзя допускать передачу секретной информации средствами NFS. Если возникает необходимость обмена важными данными по локальной сети, для этого лучше использовать Samba или другие механизмы передачи, например, программу scp, которая входит в состав пакета SSH (Secure Shell — защищенная оболочка).

Серверы NFS для системы Linux

В 1998-2002 г. средства поддержки NFS в системе Linux претерпели ряд важных изменений; некоторые из таких изменений рассматриваются в данном разделе. Если вы используете старые дистрибутивные пакеты или устаревшую документацию, представленные здесь сведения позволят вам составить впечатление о реальном положении дел. Чаще всего сервер NFS, поставляемый в составе Linux, можно использовать для обмена данными, но в некоторых случаях, чтобы реализовать NFS-взаимодействие с другими компьютерами, вам придется установить более новое (а возможно, и более старое) программное обеспечение. Информацию о последних разработках в области NFS-обмена в системе Linux можно получить, обратившись по адресу `http://nfs.sourceforge.net`.

Пользовательский режим и режим ядра

Сервер NFS в основном предназначен для обмена данными между файлами на диске и сетевым интерфейсом. В обычных условиях сервер NFS выполняется в системе Linux в пользовательском режиме. Это означает, что сервер не имеет специальных привилегий и не использует средства ядра. Другими словами, информация читается с диска с помощью функций ядра, затем прочитанные данные передаются программе, работающей в пользовательском режиме, а после этого они поступают на сетевой интерфейс. (Данные, принятые посредством сетевого интерфейса и записываемые на диск, проходят этот путь

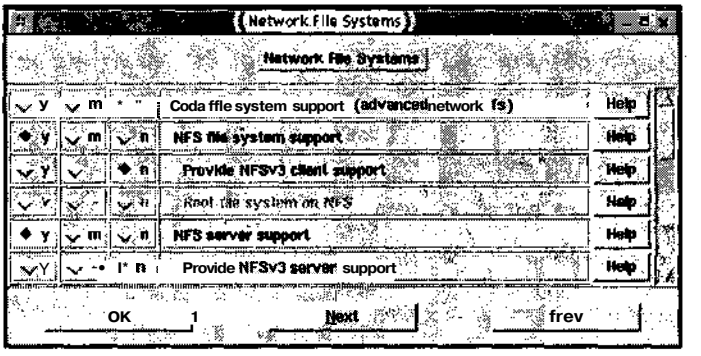


Рис. 8.1. В ядре Linux реализованы средства поддержки как клиента, так и сервера NFS

в обратном направлении.) Необходимость обмена информацией между ядром и программой, выполняющейся в пользовательском режиме, снижает производительность системы. Чтобы устранить этот недостаток, надо изменить коды сервера NFS и конфигурацию ядра так, чтобы передачей данных занимались только функции ядра. Для этого необходимо установить опцию NFS Server Support в подменю Network File Systems меню File Systems (рис. 8.1). Сделав это, вы передадите ядру часть обязанностей сервера NFS. Кроме того, надо использовать код сервера NFS, непосредственно ориентированный на взаимодействие с ядром. Обычно программа, реализующая такой сервер, называется `knfsd`, в то время как стандартный сервер NFS носит имя `nfsd`.

В инструментах настройки ядра Linux также присутствует опция NFS File System Support. Эта опция включает в ядро средства поддержки клиента NFS, которые совместно с утилитой mount позволяют монтировать каталоги, экспортируемые удаленным сервером NFS, в локальной файловой системе. Средства поддержки клиента и сервера NFS в составе ядра не связаны друг с другом, и вы можете независимо включать или отключать любую из этих опций.

NFSv2 и NFSv3

Подобно другим протоколам и программам, средства NFS периодически пересматриваются и реализуются их новые версии. В 2002 г. широко использовалась последняя на тот момент версия 3 системы NFS, или NFSv3. (На самом деле в это время уже существовала версия NFSv4, но она находилась в стадии разработки. Дополнительную информацию о состоянии дел с NFSv4 можно найти, обратившись по адресу `http://www.nfsv4.org`.) Несмотря на наличие NFSv3 (и даже NFSv4), большинство существующих клиентов и серверов NFS поддерживают лишь NFSv2. То же самое можно сказать о ядре Linux 2.2.x. Возможность работы с NFSv3 была реализована в ядре лишь начиная с версии 2.2.18. (Для более ранних версий ядра существуют дополнительные модули, обеспечивающие поддержку NFSv3.) В NFSv3 были предусмотрены дополнительные возможности, например, улучшена блокировка файлов, повышена производительность операций записи за счет применения асинхронного режима (асинхронный режим был реализован и в программах поддержки NFSv2 в системе Linux, но соответ-

ствующие средства не соответствовали стандарту). Кроме того, NFSv3 позволяет работать с NQNFS (Not Quite NFS) и использовать соединения TCP (в NFSv2 был предусмотрен только UDP-обмен). Следует заметить, что в 2002 г. соединения TCP поддерживались в Linux лишь частично. Средства NFSv2 подходят для небольших сетей, в которых необходимость в обмене данными возникает лишь эпизодически, а NFSv3 (реализованные в полном объеме) можно использовать для обеспечения работы мощных серверов. Экспериментальные версии NFSv3 для Linux были реализованы плохо — они не поддерживали операции в асинхронном режиме. Для серверных программ ситуация несколько улучшилась с появлением ядра 2.4.x. Клиентские программы в ядре 2.4.17 работают по-прежнему медленно.

Если вы хотите обеспечить работу с сервером NFSv3, в котором операции NFS ускоряются за счет ядра, вы должны при установке конфигурации ядра выбрать опцию Provide NFSv3 Server Support (которая является подопцией обсуждавшейся ранее опции NFS Server Support). Аналогично, для использования средств NFSv3 клиентом надо выбрать опцию Provide NFSv3 Client Support. Протокол NFS обеспечивает совместимость с ранними версиями, поэтому если в вашей системе поддерживаются средства NFSv3, а на других компьютерах установлены лишь средства NFSv2, то узлы сети могут обмениваться данными по протоколу NFSv2.

Если вы хотите обмениваться данными по протоколу NFSv3, то, помимо установки опций ядра, вам надо использовать утилиты, поддерживающие эту версию. Для обеспечения работы клиента вам понадобятся `nfs-utils` 0.1.6 либо более поздняя версия и версия утилиты mount не ниже 2.10т. Эти инструменты содержатся в большинстве дистрибутивных пакетов; для их поиска можно воспользоваться средствами `rpm` или `dp`kg. Так, например, чтобы найти нужную версию mount, надо задать следующую команду:

```
$ rpm -q mount
mount-2.11b-5mdk
```

В данном случае при выполнении команды было обнаружено, что в системе инсталлирована версия 2.11b утилиты mount, которая подходит для обеспечения работы клиента NFSv3.

Отображение портов

Большинство серверов TCP/IP принимают обращения от клиентов через порт с определенным номером. Так, например, сервер, реализующий протокол SMTP (Simple Mail Transfer Protocol — простой протокол передачи почты), использует при работе порт 25, а Web-сервер, поддерживающий протокол HTTP (Hyptertext Transfer Protocol — протокол передачи гипертекстовой информации), — порт 80. Обмен с сервером также может быть организован через нестандартный порт, но обычно конфигурацию сервера выбирают так, чтобы для обращения к ним не требовалась специальная настройка клиентов. NFS представляет собой класс протоколов, которые действуют несколько по-иному. При работе NFS применяется процедура *отображения портов*. Специальная программа связывается с фиксированным портом (номер порта 111) и перенаправляет обращения клиентов на требуемые порты. (NFS чаще всего использует порт UDP 2049, но NFSv3 предполагает также работу через порт TCP 2049.) Весь процесс обмена данными базируется на применении протокола RPC (Remote Procedure Call — удаленный вызов процедур).

Процедуру отображения портов реализует программа portmap, которая обычно запускается при выполнении сценария загрузки сетевых средств. Кроме того, для нее может быть создан отдельный сценарий. Несмотря на то что portmap не работает совместно с суперсервером inetd, в последних версиях этой программы предусмотрена возможность взаимодействия с TCP Wrappers. Ограничив доступ к программе отображения портов теми компьютерами, которым действительно необходимо взаимодействовать с сервером, вы существенно повысите безопасность системы. Чтобы запретить доступ всем узлам без исключения, надо включить в файл /etc/hosts.deny следующую запись:

```
portmap : ALL
```

Затем можно разрешить обращение к portmap отдельных компьютеров или сетей, включая их адреса в файл /etc/hosts.allow.

```
portmap : 192.168.1.
```

В главе 4 обсуждались вопросы настройки TCP Wrappers, в частности, способы описания клиентов. При работе с программой отображения портов не следует указывать доменные имена клиентов, так как процедура преобразования имен может привести к повторному обращению к portmap. Это, в свою очередь, приведет к необходимости нового преобразования адресов. Такая бесконечная последовательность вызовов не даст никакого результата, а лишь создаст дополнительную нагрузку на процессор. Поэтому для указания клиентов надо использовать их IP-адреса.

Для обеспечения NFS-обмена недостаточно вызвать программу отображения портов. Вам также надо определить разделяемые каталоги (этот вопрос будет рассматриваться в следующем разделе) и запустить сам сервер NFS. Для запуска сервера NFS используется сценарий SysV (обычно он называется nfs). В некоторых версиях Linux приходится также запускать дополнительные сценарии SysV. При инсталляции NFS эти сценарии устанавливаются автоматически. Изменив конфигурацию сервера, необходимо перезапустить его; это можно сделать, используя опцию restart сценария. Соответствующая команда выглядит следующим образом: /etc/rc.d/init.d/nfs restart.

Разделение файлов с помощью NFS

Для того чтобы обеспечить совместное использование файлов, надо сообщить серверу NFS о том, какие каталоги должны экспортироваться и какие клиенты имеют право доступа к конкретным каталогам. Кроме того, необходимо указать опции, управляющие доступом и определяющие другие характеристики сервера. Для монтирования каталогов, экспортированных сервером NFS, на стороне клиента используется программа mount, но вместо локального файла устройства при ее вызове указывается сервер NFS и задается имя монтируемого каталога.

Определение экспортируемых каталогов

Для управления сервером NFS используется файл /etc/exports. В этом файле содержится набор записей, каждая из которых определяет экспортируемый каталог. Запись занимает одну строку и имеет следующий формат:

```
экспортируемый_каталог клиент1(опции) [клиент2(опции) [...]]
```

Имя экспортируемого каталога может иметь вид /home или /usr/X11R6. Вы можете указать любой каталог, однако некоторые каталоги экспортировать нецелесообразно. Так, например, предоставив доступ к /etc или /proc, вы создадите угрозу безопасности компьютера, так как удаленные пользователи получат доступ к информации, определяющей конфигурацию системы. Некоторым может показаться, что, экспортируя каталог /dev, вы предоставите удаленным пользователям доступ к устройствам сервера, но это не так. Файлы, содержащиеся в этом каталоге, всегда определяют устройства локального компьютера, поэтому, смонтировав /dev, вы получите лишь копии файлов, с помощью которых можно в лучшем случае взаимодействовать с устройствами на клиентской машине. Если конфигурация компьютера, на котором расположен сервер NFS, отличается от конфигурации клиентской машины, то после монтирования /dev на клиентской машине станут доступны файлы, которые описывают несуществующие устройства. Использование таких файлов может представлять опасность для клиентской системы. (Эту проблему можно разрешить, используя опцию nodev, которая будет описана ниже.)

При описании экспортируемых каталогов указываются отдельные клиенты или группы клиентов. Ниже перечислены способы, позволяющие задавать клиентов, которым разрешен доступ к экспортируемому каталогу.

- **Отсутствующий идентификатор клиента.** Если вы зададите только список опций, помещенный в скобки, к экспортируемому каталогу сможет обращаться любой клиент. Такая конфигурация недопустима с точки зрения безопасности системы и может применяться лишь в исключительных случаях.

- **Имя одного компьютера.** Если вы укажете имя конкретного компьютера, например larch или larch.threeroomco.com, клиентские программы, расположенные на этом компьютере, получат доступ к разделяемому каталогу. Если имя домена не указано, предполагается, что клиент принадлежит тому же домену, что и сервер.

- **Группа клиентов, определенная с помощью символов групповой операции. При** описании клиента могут быть использованы знак вопроса (?), заменяющий один символ, и звездочка (*), заменяющая группу символов в имени компьютера. Например, идентификатор *.threeroomco.com определяет все машины в домене threeroomco.com. Символы (?) и (*) не заменяют точку (.), поэтому с их помощью нельзя определить компьютеры, принадлежащие поддоменам. Например, выражение *.threeroomco.com не описывает компьютер mulberry.bush.threeroomco.com.

- **Группа NIS.** Если в вашей сети присутствует сервер NIS (Network Information Service — сетевая информационная служба), вы можете задавать группы NIS, указывая в начале имени группы символ @.

- **Группа клиентов, заданная с помощью IP-адреса сети.** Ограниченную группу клиентов можно описывать, указывая адрес и маску подсети, например 172.19.0.0/255.255.0.0. Допускается также определение маски подсети как число битов, соответствующих адресу посети, например 172.19.0.0/16. (Задавая IP-адрес одного компьютера, маску подсети можно не указывать.)

С точки зрения безопасности системы предпочтительнее использовать для идентификации компьютеров их IP-адреса, так как если злоумышленник получит доступ к серверу

DNS или NIS, доменные имена и имена групп могут быть переопределены. IP-адрес также можно подделать, особенно если попытка незаконного доступа осуществляется из локальной сети, но использование IP-адреса исключает по крайней мере один способ атаки. С другой стороны, указание компьютеров с помощью IP-адресов может быть неудобным, в особенности если адреса часто меняются. Так, например, происходит, если адреса компьютерам выделяет сервер DHCP, работа которого рассматривалась в главе 5 данной книги.

Как вы уже знаете, доступ к программе `portmap` можно ограничить с помощью TCP Wrappers. При этом указание идентификаторов клиентов в описании экспортируемых каталогов может показаться *излишней* мерой. Это не совсем верно. Ограничивая доступ к серверу, надо использовать для этого все доступные способы. Дело в том, что средства, блокирующие обращения, могут быть сконфигурированы неверно, а при наличии недостатков в системе защиты их можно обойти. В этом случае избыточные механизмы ограничения доступа будут очень полезны. Рекомендуется также запретить обращение ряда узлов к серверу, применяя для этого фильтрацию пакетов. Данный способ будет рассматриваться в главе 25.

В состав многих дистрибутивных пакетов Linux входят средства брандмауэра; они легко конфигурируются в процессе инсталляции. Некоторые из них, например брандмауэр для Red Hat, уже настроены с учетом блокирования доступа к серверу NFS, поэтому в некоторых случаях разрешить доступ бывает достаточно трудно. Если у вас возникнут проблемы, связанные с взаимодействием клиентов с сервером NFS, ознакомьтесь с материалом, изложенным в главе 25, он поможет вам изменить настройку брандмауэра.

Для каждого клиента или группы клиентов можно задать отдельный набор опций. Эти опции помещаются в скобки и указываются после идентификатора клиента. Опции отделяются одна от другой запятыми. Некоторые из них управляют доступом; их использование рассматривается в следующем разделе. Другие опции задают характеристики сервера и влияют на производительность. Примеры таких опций приведены ниже.

- sync и async. Данные опции задают соответственно синхронный и асинхронный режимы выполнения операций. При записи в асинхронном режиме сервер может сообщить клиенту о том, что операция завершена, в то время как запись на диск еще продолжается. Это ускоряет процесс обмена данными, но создает угрозу их целостности; в случае выхода сервера из строя информация будет утеряна. Официально считается, что NFSv2 не поддерживает асинхронные операции, но, несмотря на это, сервер NFS в системе Linux позволял выполнять действия в асинхронном режиме. NFSv3 поддерживает асинхронный режим, а для снижения риска к клиенту предъявляются требования буферизации данных. По умолчанию в серверах NFSv3 предполагается опция `async`, но в бета-версиях программ NFS для Linux данная опция игнорируется.

- wdelay и no_wdelay. Если сервер NFS, работающий в системе Linux, предполагает, что последующие запросы могут изменить данные, предназначенные для записи на диск, он может отложить на некоторое время процедуру записи. Во многих случаях такой подход позволяет увеличить производительность сервера. Изменить

принцип записи можно, указывая опции wdelay и no_wdelay. Опция wdelay предполагается по умолчанию.

Средства контроля доступа

Многие из опций, которые указываются для каждого клиента в файле /etc/exports, предназначены для управления доступом. Как было сказано ранее, NFS использует механизм доверия, поэтому сервер не может проверить имя пользователя и пароль, как это происходит в системе Samba. Если клиент объявлен как заслуживающий доверия, то решение о предоставлении доступа принимается на основании сведений о принадлежности файла владельцу и правах. Некоторые из опций управления доступом, встречающиеся в файле /etc/exports, перечислены ниже.

- secure и insecure. По умолчанию сервер NFS считает, что запросы должны поступать с защищенных портов, номера которых не превышают 1023. В системах UNIX и Linux доступ к таким портам имеет только пользователь root (право работы через порты с номерами 1024 и выше предоставлено всем пользователям). Разрешая обращения клиентов, которые используют номера портов, превышающие 1023 (т. е. задавая опцию insecure), вы предоставляете пользователям, не обладающим привилегиями, дополнительный шанс осуществить несанкционированный доступ к серверу. В некоторых случаях, например при тестировании клиентских программ, использование опции insecure может быть оправдано.

- ro и rw. Опция ro разрешает только читать содержимое экспортируемого каталога, а опция rw предоставляет также возможность записывать данные в этот каталог. В сервере knfsd, использующем функции ядра, по умолчанию принимается опция ro, а в серверах, выпущенных ранее, по умолчанию предполагалось, что задана опция rw. Чтобы предотвратить возникновение ошибок, рекомендуется задавать требуемую опцию в явном виде.

- hide и nohide. Предположим, что на сервере NFS каталог /usr размещен в одном разделе, а каталог /usr/local — в другом. Если вы экспортируете каталог /usr, должен ли экспортироваться также и каталог /usr/local? Ответ на данный вопрос зависит от используемого сервера. В ядре 2.2.x для этого была предусмотрена специальная опция. В последних версиях сервера NFS вы можете управлять его поведением, задавая опции hide и nohide. Опция hide скрывает смонтированные разделы, а опция nohide выполняет противоположное действие. Некоторые клиенты допускают ошибки в работе со смонтированными разделами, поэтому в ряде случаев приходится задавать опцию hide. При этом клиент должен самостоятельно монтировать оба каталога.

- noaccess. Данная опция запрещает доступ к каталогу, даже если он является подкаталогом экспортируемого каталога. Предположим, например, что вы хотите экспортировать поддерево /home, за исключением каталога /home/abrown. Для этого надо создать в файле /etc/exports обычную запись для каталога /home и отдельную запись для каталога /home/abrown, указав в ней опцию noaccess. В результате пользователи не смогут обращаться к каталогу /home/abrown.

- subtree_check и no_subtree_check. В некоторых случаях приходится экспортировать не весь раздел, а лишь его часть. При этом сервер NFS должен вы-

полнять дополнительную проверку обращений клиентов, чтобы убедиться в том, что они предпринимают попытку доступа лишь к файлам, находящимся в соответствующих подкаталогах. Такой *контроль поддерева* (subtree checks) несколько замедляет взаимодействие с клиентами, но если отказаться от него, могут возникнуть проблемы с безопасностью системы. Отменить контроль поддерева можно с помощью опции no_subtree_check. Опция subtree_check, включающая такой контроль, предполагается по умолчанию. Контроль поддерева можно не выполнять в том случае, если экспортируемый каталог совпадает с разделом диска.

- root_squash и no_root_squash. По умолчанию сервер NFS отвергает обращения, которые исходят от пользователя root, работающего на клиентском компьютере. Эти обращения интерпретируются как попытки доступа локального анонимного пользователя. Такая мера повышает уровень безопасности системы, предполагая, что привилегии root на удаленном компьютере могли быть получены незаконно. Если же вам необходимо выполнять администрирование сервера с удаленного узла, то, для того, чтобы иметь возможность работать с привилегиями локального пользователя root, надо задать опцию no_root_squash. Подобная мера может потребоваться, например, при создании резервных копий.

- all_squash и no_all_squash. В обычных условиях обращения от пользователей принимаются, но иногда приходится запрещать доступ к экспортируемым каталогам, содержащим важные данные. Сделать это можно с помощью опции all_squash. Опция no_all_squash отменяет действие all_squash.

- anonuid и anongid. Анонимным пользователем, обращения которого отвергаются, обычно считается пользователь nobody. Вы можете переопределить такую установку, указав идентификатор пользователя (UID) и идентификатор группы (GID). Сделать это позволяют соответственно опции anonuid и anongid. В этом случае пользователю root, работающему на удаленном клиенте, будет предоставлен доступ с привилегиями указанного пользователя. Эти опции также приходится указывать при работе с клиентами PC/NFS, которые поддерживают лишь одного локального пользователя. Такая опция должна сопровождаться знаком равенства и идентификатором пользователя или группы, например anonuid=504.

Пример файла /etc/exports показан в листинге 8.1. В этом файле описаны два экспортируемых каталога: /usr/X11R6 и /home. Кроме того, в нем содержится третья запись, запрещающая с помощью опции noaccess обращения к каталогу /home/abrown. (Поскольку последняя запись лишь ограничивает доступ, в ней не указан конкретный узел; обращаться в данному каталогу не может ни один клиент.) Каталоги /usr/X11R6 и /home доступны для компьютера gingko и всех узлов сети 192.168.4.0/24, однако при экспортировании этих каталогов заданы различные опции. Каталог /usr/X11R6 доступен только для чтения, а в каталог /home клиенты имеют также право записывать данные. На компьютере gingko для доступа к /usr/X11R6 задан идентификатор анонимного пользователя, равный 514, а при обмене с каталогом /home не выполняется контроль поддерева.

Листинг 8.1. Пример файла /etc/exports

```
/usr/X11R6 gingko(ro,anonuid=504) 192.168.4.0/24(ro)
/home gingko(rw,no_subtree_check) 192.168.4.0/255.255.255.0(rw)
/home/abrown (noaccess)
```

Монтирование экспортируемых каталогов

На стороне клиента экспортируемые каталоги выглядят как разделы диска. Для их монтирования используется команда mount, но при ее вызове указываются сервер NFS и монтируемый каталог. Эти данные задаются в формате *сервер:путь_к_монтируемому_каталогу*. Так, например, следующая команда монтирует экспортируемый каталог /home в точке файловой системы /mnt/userfiles:

```
# mount larch:/home /mnt/userfiles
```

Если вы хотите, чтобы экспортируемый каталог был постоянно доступен, вам надо создать запись в файле /etc/fstab. Как и при использовании команды mount, вместо имени устройства вы указываете имя сервера и путь к экспортируемому каталогу. Тип файловой системы задается как nfs (при желании вы можете задать соответствующую опцию и при вызове mount, но это не обязательно, поскольку Linux автоматически распознает тип файловой системы). Приведенная ниже запись в файле /etc/fstab выполняет те же действия, что и рассмотренный ранее вызов утилиты mount.

```
larch:/home /mnt/userfiles nfs defaults 0 0
```

В результате пользователь, обращаясь к каталогу /mnt/userfiles, на самом деле увидит содержимое каталога /home на узле larch. С содержимым смонтированного каталога NFS можно выполнять большинство операций, допустимых для локального раздела Linux. Например, вы можете читать, редактировать и удалять файлы, а также выполнять прочие действия. Существуют также операции, которые недопустимы для экспортируемых каталогов, например, вы не можете объявлять раздел NFS как файл подкачки. В большинстве случаев эффективность работы с экспортируемыми каталогами NFS ниже, чем с разделами локального диска, так как обмен по сети осуществляется значительно медленнее, чем обмен с современными жесткими дисками. Однако в отдельных случаях, например при использовании гигабитовой Ethernet-сети, производительность NFS-обмена может даже превышать производительность работы с локальными устройствами, особенно если на клиентской машине используются устаревшие диски. На производительность системы NFS существенное влияние оказывают также быстродействие диска сервера и число клиентов.

Экспортируя каталоги и содержащиеся в них файлы, сервер NFS экспортирует также права доступа к ним. Информацию о пользователях и правах можно применять для контроля обращений к файлам и каталогам. Эти средства можно использовать даже для управления доступом со многих компьютеров, например, в случае, если один сервер NFS обслуживает несколько клиентов. Однако при этом может возникнуть проблема, которая состоит в следующем. Для идентификации пользователей в системе NFS применяются UID и GID. Если такие идентификаторы не совпадают на клиентах и на сервере, это может угрожать безопасности системы. Способы разрешения данной проблемы будут рассмотрены ниже в этой главе.

В последующих разделах будут описаны некоторые опции программы mount, которые влияют на поведение клиентов и серверов NFS и могут быть использованы для увеличения производительность и решения других задач. Ряд опций утилиты mount перечислен ниже.

- hard. Если сервер выходит из строя или не отвечает на запросы, программа, которая пытается обратиться к этому серверу, ожидает ответа неопределенно долгое время. Такое поведение системы реализовано по умолчанию.
- **soft**. Если сервер NFS часто выходит из строя и становится недоступным, целесообразно использовать данную опцию. Она позволяет ядру возвращать программе сообщение об ошибке в том случае, если сервер не отвечает в течение установленного времени (это время задается с помощью опции **timeo=*время***).
- nodev. Данная опция предотвращает попытки клиента использовать файлы символьных и блочных устройств, находящиеся в составе экспортируемых каталогов NFS. Такая мера увеличивает безопасность системы, так как снижает риск использовать файл устройства, специально включенный в каталог NFS с целью получения несанкционированного доступа к клиентской машине.
- nosuid. Эта опция не позволяет клиенту обрабатывать бит SUID в файлах, находящихся в экспортируемом каталоге. Эта опция также призвана повысить защиту системы. Она не дает возможности использовать бит SUID для **незаконного** получения специальных полномочий.
- noexec. Эта опция предотвращает обработку клиентом признака исполняемых файлов в экспортируемых каталогах NFS. Другими словами, пользователь не может запускать на выполнение файлы, содержащиеся в каталогах NFS. В некоторых случаях эта опция неуместна, например, тогда, когда NFS используется для хранения файлов с программами. Если же в каталоге находятся лишь данные, эта опция повысит безопасность системы.

Перечисленные опции можно задавать при вызове команды mount, указывая их после -o, например:

```
# mount -o noexec,nodev larch:/home /mnt/userfiles
```

Если вы создаете запись в файле **/etc/fstab**, данные опции указываются в специально предназначенном поле (в этом поле в приведенном выше примере находилось ключевое слово **defaults**).

Повышение производительности системы

Выше были описаны два способа повышения производительности системы: поддержка NFS ядром (совместно с программой **knf**sd) и использование асинхронного режима. (Необходимо заметить, что асинхронный режим записи повышает риск потери данных вследствие сбоя сервера.) Ниже перечислены другие способы, позволяющие увеличить эффективность работы NFS.

- **Оптимизация размера передаваемых блоков.** Опции **rsize** и wsize программы mount определяют размер блоков данных, передаваемых между клиентом и сервером. Размер блока по умолчанию зависит от используемых программ, но чаще всего

принимается значение 4096. Команда, в которой явно задается размер блока, выглядит приблизительно так: `larch:/home /mnt/userfiles -o rsize=8192`. При создании записи в файле `/etc/fstab` эти опции помещаются в специальное поле (в приведенном ранее примере в этом поле указано значение `defaults`).

- **Оптимизация за счет исключения информации об обращении к файлу.** Опция noatime утилиты mount сообщает Linux о том, что информация о времени обращения к файлу не должна обновляться. В обычных условиях Linux записывает сведения о времени создания и изменения файла, а также о времени последнего обращения к нему. Отказавшись от данных о времени обращения, можно повысить производительность системы.

- **Выбор количества экземпляров сервера** NFS. Как правило, в сценарии запуска сервера NFS предусматривается одновременное выполнение восьми экземпляров этой программы. Если нагрузка на систему не велика, с ней может справиться и меньшее количество серверов. Если же система постоянно обрабатывает запросы, восьми серверов может оказаться недостаточно. Недостаток серверов при большой нагрузке на систему приводит к тому, что для установления соединения с клиентом потребуется неоправданно большое время. Увеличить производительность можно, выбрав наиболее подходящее число экземпляров сервера. Обычно это значение задается в начале сценария и имеет вид RPCNFSDCOUNT=8.

- **Устранение причин снижения производительности, не связанных с работой средств поддержки** NFS. Производительность NFS может снижаться по ряду причин, многие из которых непосредственно не связаны с работой сети. Например, если ваша сетевая карта работает медленно, эффективность NFS-обмена будет низкой. Качество работы сервера NFS существенно зависит от используемого жесткого диска. Важно, чтобы данное устройство было достаточно быстродействующим и не создавало излишней нагрузки на центральный процессор. (Для сервера хорошо подойдет EIDE-контроллер с поддержкой адаптера DMA или SCSI; производительность устройств SCSI выше, чем производительность дисков EIDE.)

Если производительность сервера NFS недостаточна, необходимо прежде всего выяснить, что является источником проблем: программное обеспечение сервера NFS, клиент-программы или конфигурация сети. Возможно также, что производительность снижается по другим причинам, например, из-за недостаточной эффективности работы жестких дисков. Выяснить это можно, выполняя тестирование с использованием различных протоколов и с участием разных клиентов. (Для проверки производительности жестких дисков можно вызвать команду `hdparm`, указав опцию -t.)

Отображение пользовательских имен

На компьютере под управлением Linux, работающем независимо от других машин, за отображение пользовательских имен в числовые идентификаторы (UID) отвечает файл `/etc/passwd`. Аналогично, информация о соответствии имен групп и их идентификаторов (GID) хранится в файле `/etc/group`. В системе NFS существуют как минимум два независимых файла /etc/passwd: один — на сервере и по одному — на каждом клиенте. При этом может возникнуть проблема, связанная с тем, что одному и тому же

пользователю на сервере и на клиентской машине будут соответствовать различные UID. Эта проблема может быть решена различными способами.

При рассмотрении данного вопроса предполагается, что пользователь имеет учетную запись и на клиентском компьютере, и на сервере. Если сервер используется только в качестве файлового сервера и содержащаяся на нем информация доступна только для чтения, учетная запись пользователя на нем может отсутствовать. Такая же конфигурация может быть реализована на некоторых серверах, допускающих чтение и запись. При настройке сервера, который допускает чтение и запись и обслуживает несколько пользователей, вопрос синхронизации числовых идентификаторов становится очень важным. Если на сервере NFS хранятся файлы, принадлежащие некоторому **пользователю,** целесообразно создать на сервере учетную запись данного **пользователя,** даже если он никогда не будет регистрироваться на этой машине. Если же пользователь не является владельцем ни одного из файлов, хранящихся на сервере, отображать пользовательское имя не требуется.

Согласование идентификаторов пользователей на клиентском компьютере и на сервере

Самым простым решением описанной выше проблемы отображения пользовательских имен является синхронизация UID и GID на клиентской машине и на сервере. Например, если некоторому пользователю на сервере соответствует идентификатор 504, необходимо принять меры к тому, чтобы и на клиентском компьютере его UID был также равен 504. Аналогичным образом должны быть синхронизированы GID. Если сеть насчитывает большое число компьютеров, синхронизация UID и GID займет очень много времени. Однако в небольшой сети, при условии, что число пользователей в ней невелико, действия по синхронизации могут быть выполнены относительно просто. Привести в соответствие существующие пользовательские идентификаторы можно с помощью утилиты `usermod`. Например, для того, чтобы изменить UID пользователя abrown с 507 на 504, надо вызвать следующую команду:

```
# usermod -u 504 abrown
```

При выполнении этой команды будет изменена запись в файле `/etc/passwd` и скорректированы данные о владельце файлов, расположенных в рабочем каталоге данного пользователя. (Информацию о принадлежности файлов, которые находятся за пределами рабочего каталога, следует ввести вручную.) Для завершения этой команды потребуется некоторое время. Если вы прервете ее выполнение, вам придется самостоятельно задавать нового владельца некоторых файлов, содержащихся в рабочем каталоге пользователя.

Команда `groupmod` аналогичным образом изменяет информацию о группе. В отличие от команды usermod, новый идентификатор группы задается с помощью опции -д. Например, чтобы задать GID группы `project3` равным 127, надо выполнить следующую команду:

```
# groupmod -g 127 project3
```

ВНИМАНИЕ Не пытайтесь изменить UID или GID в то время, когда пользователь, идентификатор которого должен быть скорректирован или члены группы зарегистрированы в системе. Это приведет к тому, что пользователь не сможет сохранить результаты своей работы, прочитать файл, запустить программу и выполнить другие подобные действия. Если вы внесли такие изменения непреднамеренно, постарайтесь отменить их либо предложите пользователю завершить сеанс работы и зарегистрироваться повторно. Если при этом пользователю необходимо сохранить данные, их можно записать в один из общедоступных каталогов, например в `/tmp`.

Говоря о данном способе синхронизации идентификаторов, важно заметить, что имена пользователя на клиентской машине и на сервере не обязательно должны совпадать. Например, один и тот же пользователь может иметь имя `abrown` на сервере и имя `alyson` — на клиентском компьютере. Когда этот пользователь, работая в клиентской системе, обращается к серверу, считается, что файлы на сервере принадлежат пользователю `alyson`. Если тот же пользователь зарегистрируется на сервере NFS, система сообщит, что владельцем файлов является abrown. Первоначально такая особенность затрудняет администрирование системы, но в некоторых ситуациях она может оказаться полезной.

Обеспечить синхронизацию UID и GID можно, используя отдельный сервер для аутентификации пользователей как на клиентских компьютерах, так и на сервере NFS. В этом случае пользователь получит один и тот же UID, независимо от компьютера, на котором он зарегистрируется, а конкретной группе будет соответствовать единственный GID. В качестве такого средства аутентификации можно использовать систему Kerberos, которая была рассмотрена в главе 6. Кроме того, реализация NFS для Linux включает поддержку аутентификации NIS; для этой цели используется опция map_nis. Если вы включите эту опцию в файл `/etc/exports` для некоторого клиента, сервер NFS предоставит серверу NIS выполнить отображение пользовательского имени.

Средства синхронизации идентификаторов пользователей, выполняемые на стороне сервера

Предположим, что вы занимаетесь администрированием сети, состоящей из двух компьютеров. На каждом узле этой сети существуют учетные записи для пользователей, перечисленных в табл. 8.1. В данном примере компьютер `gingko` выполняет функции сервера, а компьютер `larch` выступает в роли клиента. Только у одного из пользователей (`james`) идентификаторы на обоих компьютерах совпадают. Чтобы пользователь `james` мог обращаться к своим собственным файлам, никакие специальные меры не требуются. Работая на компьютере `larch`, `alyson` обнаружит, что его файлы, хранящиеся на gingko, принадлежат пользователю, идентифицировать которого невозможно (UID, равный 500, на компьютере `larch` отсутствует). Что касается остальных двух пользователей, `Jennie` и `samuel`, система сообщит, что каждый из них является владельцем файлов, принадлежащих на самом деле другому.

Один из способов решения проблемы синхронизации пользовательских идентификаторов состоит в следующем. На сервере NFS создается файл соответствия идентификаторов, содержащий информацию, подобную приведенной в табл. 8.1. О наличии этого файла сервер оповещается посредством опции `map_static`; в качестве значения опции

Таблица 8.1. Идентификаторы пользователей на двух компьютерах

Пользователь	UID на gingko	UID на larch
alyson	500	504
james	501	501
Jennie	502	503
samuel	503	502

задается имя файла соответствия идентификаторов. В файл /etc/exports включается запись, которая может выглядеть следующим образом:

```
/home  larch(rw,map_static=/etc/nfs/larch-map)
```

Эта запись сообщает системе о том, что, предоставляя каталог /home пользователю larch, надо использовать файл соответствия идентификаторов с именем /etc/nfs/larch-map. Поскольку опция map_static входит в состав списка опций для конкретного клиента, вы можете назначать разным клиентам различные файлы соответствия. Пример содержимого файла larch-map показан в листинге 8.2. Строки, в начале которых находится символ #, содержат комментарии. Строки, начинающиеся с uid, представляют информацию о соответствии пользовательских идентификаторов, а строки, в начале которых расположено ключевое слово gid, содержат сведения о соответствии идентификаторов групп. Первое из числовых значений (или диапазон значений) в строке представляет идентификатор на клиентской машине. Второе числовое значение соответствует идентификатору, в который должен отображаться UID или GID, полученный на удаленном компьютере. Например, из листинга 8.2 видно, что UID 504 на клиентском компьютере отображается в UID 500 на сервере. Если вместо идентификатора на сервере указан символ -, обращение данного пользователя или члена группы к серверу NFS запрещен. Такое обращение интерпретируется как попытка доступа анонимного пользователя.

Листинг 8.2. Пример содержимого файла соответствия идентификаторов

```
# Отображение идентификаторов для клиента larch
#       удаленный     локальный
uid     0-99          -             # доступ запрещен
uid     504           500
uid     501           501
uid     503           502
uid     502           503
gid     0-99          -             # доступ запрещен
gid     100-102       100
```

В файле соответствия необходимо задать идентификаторы всех пользователей. Например, в листинге 8.2 указан UID 501, который отображается в тот же идентификатор на сервере. Отсутствующий UID приведет к некорректному отображению, что, в свою очередь, станет источником проблем. В листинге 8.2 явным образом указано, что попытки обращения с системных UID (с номерами меньше 100) должны отвергаться. Аналогичное правило задано для идентификаторов групп 0-99. GID 100-102 отображаются в GUID

100. Несмотря на то что вы можете отобразить диапазон клиентских идентификаторов в единственный идентификатор на сервере, обратное действие не имеет смысла. При попытке пользователя с определенным UID на стороне клиента создать файл сервер не сможет выбрать локальный идентификатор.

Как и в случае, когда синхронизация идентификаторов на клиентской машине и сервере производится **вручную,** имена пользователей на клиентском компьютере и на сервере могут различаться. В файле соответствия содержится исключительно информация об идентификаторах пользователей и групп; сведения об именах отсутствуют. Несмотря на то что подобная ситуация не мешает нормальной работе, желательно согласовать пользовательские имена на клиентских компьютерах и на сервере.

Средства синхронизации идентификаторов пользователей, выполняемые на стороне клиента

Решить проблему синхронизации пользовательских идентификаторов можно, задавая на стороне сервера опцию map_daemon. Эта опция позволяет использовать на стороне клиента специальный демон, который называется `ugidd` или `rpc.ugidd`. Однако при работе с таким демоном могут возникать проблемы. Во-первых, программа `ugidd` поставляется не со всеми системами. Из дистрибутивных пакетов, которые обсуждались в данной книге, она входит только в состав Debian. Во-вторых, демон `ugidd` приходится устанавливать на всех клиентах, а это занимает много времени. В-третьих, чтобы программа не могла быть использована для несанкционированного доступа, необходимо запретить обращения к ней (это можно сделать, задавав требуемую конфигурацию в файле `/etc/hosts.allow`). И, наконец, что особенно важно, данная программа слишком сложна и в некоторых случаях она вовсе не работает либо отображает всех пользователей в пользователя `nobody`.

Резюме

NFS — чрезвычайно полезный инструмент, позволяющий обеспечить разделение файлов в системах UNIX и Linux. В отличие от Samba, NFS обеспечивает поддержку данных о владельцах файлов и правах доступа. Настройка NFS осуществляется несколько проще, чем конфигурирование средств Samba. С другой стороны, NFS использует принцип доверия, поэтому при работе с данной системой приходится предпринимать меры для синхронизации идентификаторов пользователей на клиентских машинах и серверах или хранить сведения о соответствии идентификаторов в специальном конфигурационном файле.

Глава 9

Совместное использование принтеров

Система печати, используемая в Linux, первоначально была разработана для BSD UNIX. Эта система, которую также называют по имени ее основного компонента LPD (Line Printer Daemon — демон принтера), намного проще, чем системы печати Windows и MacOS, и в то же время обеспечивает гораздо более высокую гибкость. Система LPD позволяет передавать задания на печать по сети; она включает и сервер печати, и клиент-программу. В системе LPD, в отличие от других систем, не предусмотрена поддержка драйверов принтеров. Для согласования с конкретными типами устройств применяются дополнительные пакеты (например, Ghostscript, информацию о котором можно получить по адресу `http://www.cs.wisc.edu/~ghost/`) и фильтры печати.

В данной главе рассматриваются система LPD, а также новые протоколы печати, которые в последнее время становятся все более популярными. Здесь не затрагиваются вопросы настройки компьютера для работы с конкретной моделью принтера; подобную информацию вы сможете найти в документации на вашу систему или в книгах, представляющих собой введение в систему Linux. В начале главы приводятся общие сведения о системе LPD, в частности, обсуждается функционирование сервера LPD и выбор программного обеспечения печати для работы в Linux. Затем рассматриваются вопросы настройки каждой из трех широко распространенных систем печати: BSD LPD, LPRng и CUPS.

Использование сервера LPD

Сетевая система печати работает подобно средствам разделения файлов. Клиент-программа передает на сервер печати файл, предназначенный для вывода на принтер (это можно сравнить *с* передачей файла на файловый сервер). Основное отличие между данными процессами заключается в том, что на файловом сервере файл сохраняется на диске для дальнейшего использования, а на сервере печати файл передается на принтер и после обработки удаляется. Сходство между этими двумя задачами приводит к тому, что в некоторых случаях они решаются посредством одного протокола. В качестве примера можно привести протоколы SMB/CIFS, реализованные в рамках одного продукта (Samba).

В системе UNIX для организации совместного использования файлов и принтеров традиционно применяются две различные системы: NFS и LPD.

В составе стандартных инструментов LPD интегрированы средства локальной и сетевой печати; таким образом, обеспечить прием данных, предназначенных для вывода на принтер, или передачу заданий на печать можно, затратив сравнительно небольшие усилия. В небольших сетях система сетевой печати позволяет сократить объем ресурсов, требуемых для организации работы. Так, например, вместо десяти низкоуровневых лазерных принтеров (стоимостью $300 каждый) можно приобрести за $1500 мощный принтер и использовать его как сетевое устройство печати. Сэкономленные $1500 можно потратить на цветной струйный принтер или другие подобные устройства. Одним из компонентов, позволяющих реализовать подобную конфигурацию, является LPD.

LPD — не единственный сетевой протокол печати. Как было сказано выше, выполнение подобных функций обеспечивают протоколы SMB/CIFS. Работу средств печати можно реализовать с помощью AppleTalk (в системе Linux для этого используется Netatalk). Таким образом, возникает вопрос: в каких случаях оправдано применение LPD, а когда следует отдать предпочтение другим инструментам? Для того чтобы ответить на него, следует рассмотреть следующие дополнительные вопросы.

- Какой протокол наиболее подходит для клиента? Linux поддерживает самые разнообразные протоколы печати, поэтому компьютер под управлением Linux может выполнять функции сервера печати для различных клиентов. Поскольку обычно число клиентов в сети значительно превышает число серверов, целесообразно выбрать тот протокол печати, который лучше всего поддерживается большинством клиентов. Если клиентские компьютеры работают под управлением UNIX или Linux, таким протоколом является LPD. При использовании клиентов DOS, Windows или OS/2 лучше выбрать SMB/CIFS. Для клиентов Macintosh наилучшим выбором будет AppleTalk, хотя MacOS X также хорошо поддерживает LPD.

- Какие из протоколов поддерживают необходимые вам возможности? Процедура сетевой печати реализуется проще, чем разделение файлов. Например, при печати не приходится поддерживать признаки прав доступа. Тем не менее для каждого протокола характерны свои особенности, например, по-разному выполняется процедура аутентификации. Если протокол, который наиболее подходит для большинства клиентов, не обеспечивает поддержку необходимых вам средств, надо рассмотреть вопрос об использовании другого протокола.

Вопрос поддержки специальных средств заслуживает более подробного рассмотрения. Подобно NFS, LPD использует принцип доверия, т. е. сервер полагается на результаты, полученные при аутентификации пользователя на стороне клиента. Поэтому решение о предоставлении доступа принимается на основании IP-адреса клиентского компьютера. Этот метод удобен в тех случаях, когда на клиентских компьютерах могут работать различные пользователи, но он обеспечивает гораздо более слабую защиту, чем способ, предполагающий проверку пользовательского имени и пароля. Если вам необходимо, чтобы вопрос о предоставлении доступа к серверу печати решался на основании проверки пароля, используйте протоколы, которые обеспечивают такую возможность, например протоколы SMB/CIFS. Однако при этом необходимо принять во внимание, что если клиенты будут работать под управлением UNIX или Linux, вам, возможно, придется хранить пароли в отдельном конфигурационном файле, доступном как клиентам, так и серверам,

а это вряд ли обеспечит более высокую степень защиты, чем принцип доверия. Новый протокол IPP (Internet Printing Protocol — межсетевой протокол печати), используемый CUPS, предусматривает проверку пользовательского имени и пароля, но при желании вы можете не использовать эту возможность.

Несмотря на то что на клиентских компьютерах, выполняющихся под управлением Windows, поддерживаются средства LPD, если такие клиенты присутствуют в сети, для разделения принтеров чаще всего используют протоколы SMB/CIFS. Если же на основной части клиентских машин установлены UNIX, Linux MacOS и другие системы, не поддерживающие SMB/CIFS, следует рассмотреть вопрос о применении других протоколов.

При организации сети вам придется также решать вопрос о том, в какой операционной среде целесообразно реализовать сервер печати. Большинство серверов печати, доступных для Linux, могут также выполняться и в различных версиях UNIX. Необходимо выбрать платформу, наиболее пригодную для вашей сети. Если вам необходимо организовать совместное использование принтеров клиентскими программами, работающими в Linux, UNIX, Windows, MacOS и других операционных системах, на роль универсальной платформы печати лучше всего подойдет Linux.

Перед администратором часто возникает вопрос: следует ли использовать в качестве сервера печати специальное сетевое устройство? Такие устройства обычно комплектуются Ethernet-интерфейсом и предоставляют параллельные, последовательные и USB-порты для подключения принтеров. Протокол поддержки сервера печати обычно реализован в них на аппаратном уровне. Такое устройство удобно применять в том случае, если вы не хотите использовать в качестве сервера печати обычный компьютер. Подобное решение может быть принято из соображений безопасности или тогда, когда компьютер, рассматриваемый в качестве претендента на роль сервера печати, должен время от времени выключаться. В качестве выделенных серверов печати могут выступать некоторые сетевые принтеры.

СОВЕТ

Выделенный сервер печати можно построить на базе старого компьютера; для этой цели подойдет даже компьютер с процессором 386. На нем надо установить систему Linux и отключить все программы-серверы и другие инструменты, непосредственно не участвующие в поддержке функций печати. Если быстродействие процессора достаточно велико, на данном компьютере можно установить Ghostscript, что позволит обрабатывать файлы PostScript. Таким способом можно эмулировать сетевой принтер PostScript. Если на компьютере есть несколько параллельных или USB-портов, к нему можно подключить несколько принтеров. Мощный компьютер может не только выступать в роли сервера печати, но и выполнять при этом задачи, непосредственно не связанные с выводом на принтер.

Серверы печати для Linux

Различные варианты UNIX и Linux позволяют использовать большое количество различных пакетов, предназначенных для организации печати, большинство из которых поддерживает протокол LPD. В 2001 г. наиболее популярными пакетами для Linux были следующие.

- Сервер BSD LPD. Этот пакет в течение длительного времени был стандартным для Linux. При выполнении многих Linux-программ предполагается, что в системе

Таблица 9.1. Стандартные программы печати в составе дистрибутивных пакетов Linux

Дистрибутивный пакет	Стандартная система печати	Альтернативная система печати
Caldera OpenLinux Server 3.1	CUPS	Отсутствует
Debian GNU/Linux 2.2	BSD LPD	LPRng,CUPS
Linux Mandrake 8.1	LPRng	CUPS
Red Hat Linux 7.2	LPRng	Отсутствует
Slackware Linux 8.0	BSD LPD	Отсутствует
SuSE Linux 7.3	LPRng	CUPS
TurboLinux 7.0	LPRng	Отсутствует

установлены средства BSD LPD. По этой причине LPRng и CUPS эмулируют BSD LPD, хотя делают это несколько по-разному. В BSD LPD используются предельно простые средства контроля доступа; это стало одной из причин перехода к другим системам печати.

- **Пакет LPRng.** Данная система печати, информацию о которой можно найти в документе `http://www.astart.com/lprng/LPRng.html`, создана для замены BSD LPD. Она отличается от BSD LPD форматом некоторых конфигурационных файлов. Базовая модель печати, согласно которой приложения должны иметь сведения об используемом принтере, осталась неизменной. В системе Linux большинство приложений предполагает, что вывод производится на принтер PostScript.

- **Common UNIX Printing** System (CUPS). Информация о CUPS находится на сервере `http://www.cups.org`. Данная система отличается от BSD LPD гораздо больше, чем LPRng, в частности, в ней используется другой набор конфигурационных файлов. Для приложений, специально написанных для взаимодействия с CUPS, эта система предоставляет информацию об используемых принтерах. (Для того чтобы эта информация стала доступной, средства CUPS должны выполняться и на клиентской машине, и на сервере.) Помимо протокола LPD, CUPS также поддерживает протокол печати IPP.

В UNIX-подобных операционных системах используются также другие системы печати. Одна из них применяется в некоторых версиях UNIX, построенных на базе SysV. Эта система печати может взаимодействовать с BSD LPD, но команды, используемые в ней, несколько отличаются от BSD LPD. Так, например, для передачи задания на печать вместо `lpr` следует задавать команду `lp`.

В табл. 9.1 перечислены системы печати, поставляемые в составе некоторых популярных дистрибутивных пакетов Linux. В случае, если система печати отсутствует в дистрибутивном пакете, вы можете установить ее отдельно, но для настройки программного обеспечения придется затратить дополнительные усилия. Одна из задач, которые вам придется решить, — обеспечить автоматический запуск сервера (этот вопрос рассматривался в главе 4).

Различия между "стандартными" и "альтернативными" системами печати, приведенными в табл. 9.1, весьма условны. Например, при инсталляции Mandrake вы можете выбирать, какая из систем печати должна быть установлена: LPRng или CUPS, а в Debian по умолчанию средства печати не устанавливаются вовсе.

При составлении стандартной документации на Linux, как правило, предполагается, что в системе установлены средства печати BSD LPD. Большая часть приведенных в документации сведений справедлива также для системы LPRng, различаются лишь детали, связанные с ограничением доступа к сетевому серверу печати. Что касается CUPS, то конфигурационные файлы этой системы существенно отличаются от BSD LPD и LPRng, поэтому документы, которые касаются конфигурации системы печати, не применимы к CUPS.

Настройка сервера BSD LPD

Среди средств настройки сервера BSD LPD наиболее важны два файла: /etc/hosts.lpd и /etc/printcap. В первом из них указываются клиенты, которые могут обращаться к серверу для выполнения сетевых операций. Во втором определяются принтеры, доступные как для локальных, так и для удаленных пользователей. Поскольку в файле /etc/printcap определяются и локальные, и удаленные принтеры, удаленный пользователь может передать задание на печать очереди, которая соответствует удаленной системе. Если это произойдет, задание будет принято по сети, а затем снова передано. В обычных условиях это означает напрасную затрату сетевых ресурсов, но иногда такое поведение может быть оправдано. В качестве примера можно привести ситуацию, при которой сервер печати использует Ghostscript для преобразования PostScript-файла в формат, совместимый с форматом целевого принтера.

Редактирование файла /etc/hosts.lpd

По умолчанию система BSD LPD не принимает задания на печать с удаленных компьютеров, т. е. реализующие ее программы не могут выполнять роль сетевого сервера печати. Для того чтобы изменить конфигурацию системы, необходимо отредактировать файл /etc/hosts.lpd. В этом файле указан список компьютеров, которым разрешен доступ к локальной очереди печати. Для идентификации компьютеров могут использоваться доменное имя, IP-адрес или имя группы NIS. В последнем случае перед именем группы указывается символ @, который, в свою очередь, может предваряться символом +. Символ + означает, что сервер должен принимать любое задание на печать, что небезопасно для системы. Если перед идентификатором узла указан символ -, это означает, что доступ для этого узла запрещен. Пример файла /etc/hosts.lpd приведен в листинге 9.1. При указании компьютера gingko предполагается, что он принадлежит тому же домену, что и сервер. Выражение +@group1 предоставляет доступ всем компьютерам в NIS-группе group1. Для компьютера oak.threeroomco.com доступ запрещен, даже если он принадлежит группе groupl.

Листинг 9.1. Пример файла /etc/hosts.lpd

```
gingko
birch.threeroomco.com
192.168.1.7
+@groupl
-oak.threeroomco.com
```

В файле /etc/hosts.lpd, как и в большинстве других конфигурационных файлов, символ # является признаком комментариев, однако не следует располагать комментарии за определением клиента; лучше поместить их в предыдущей или последующей строке.

ВНИМАНИЕ Аналогично /etc/hosts.lpd можно использовать файл /etc/hosts.equiv. Файл /etc/hosts.equiv применяется не только для решения задач, связанных с печатью; он также предоставляет доступ для клиентов, которые используют rlogin и другие протоколы. Из соображений безопасности этот файл использовать не рекомендуется; лучше настроить каждый сервер индивидуально. Если этот файл присутствует в системе, желательно удалить его и установить требуемую конфигурацию с помощью других файлов.

Указание сервера на клиенте BSD LPD

В файле /etc/printcap содержатся определения принтеров для системы BSD LPD (printcap сокращенно означает printer capabilities — возможности принтеров). В этом файле содержатся записи для каждой очереди печати в системе, независимо от того, является ли очередь локальной (обслуживающей принтеры, подключенные через параллельный, последовательный или USB-порт) или сетевой (обслуживающей другие LPD-принтеры и даже принтеры, доступ к которым осуществляется посредством SMB/CIFS, AppleTalk или других протоколов). Определение каждого принтера располагается в одной строке, а опции разделяются символом :. На практике определение принтера занимает несколько строк; все строки, кроме последней, заканчиваются символом \, который означает, что следующая строка является продолжением предыдущей. Размещенные таким образом данные более удобны для чтения.

Большинство деталей настройки принтеров с помощью файла /etc/printcap не рассматриваются в данной книге. Как было сказано в начале данной главы, необходимую информацию по установке принтеров, включая вопросы использования фильтров печати и Ghostscript, вы можете получить, прочитав документацию на операционную систему, или узнать из книг, представляющих собой введение в систему Linux. Однако некоторые из опций, непосредственно использующиеся при конфигурации сетевого клиента печати, требуют отдельного рассмотрения. Эти опции перечислены ниже.

- lp. Указывает на файл устройства, к которому подключен принтер. Например, выражение lp=/dev/lp0 сообщает системе о том, что для вывода на принтер должно быть использовано устройство /dev/lp0 (первый параллельный порт). Если вы настраиваете сетевой принтер, вам следует удалить эту опцию или не указывать после знака равенства никакого значения (например, lp=).

- rm. Определяет имя сервера печати LPD. Например, если для данной очереди сервером печати является oak, вам надо включить в определение очереди опцию

`rm=oak`. Заметьте, что для организации печати с помощью удаленной очереди этого недостаточно; данная опция лишь идентифицирует компьютер, на котором расположена очередь. Для определения удаленного компьютера можно использовать имя узла (с указанием или без указания доменного имени) или IP-адрес.

- `rp`. Действие опции `rp` начинается там, где заканчивается действие опции `rm`. Опция `rp` задает имя удаленной очереди печати. Например, если очередь на сервере печати называется `inkjet`, в файл `/etc/printcap` на клиентском компьютере надо включить выражение `rp=inkjet`. Заметьте, что имя удаленной очереди не обязательно должно совпадать с именем локальной очереди. Допустима, например, ситуация, когда принтер `inkjet` сервера будет называться на стороне клиента 1p1 или canon. Рекомендуется во избежание недоразумений синхронизировать имена; в особенности это важно в больших сетях.

Таким образом, имея локальную очередь печати, вы можете преобразовать ее в сетевую очередь, заменив опцию 1p опциями rm и `rp`. В этом случае вместо вывода данных на локальное устройство компьютер будет передавать задания на узел, заданный с помощью опции rm, в очередь, которая указана как значение опции `rp`. На сервере в описании очереди, вероятнее всего, будет присутствовать опция 1p, но вместо нее могут быть включены опции rm и `rp`. Следует заметить, что такая конфигурация нежелательна, в этом случае лучше непосредственно указать в клиентской системе реальный сервер печати. (Исключением являются ситуации, когда средства обработки данных, например Ghostscript, выносятся с сервера печати на другой компьютер либо когда конфигурация сети не обеспечивает непосредственное взаимодействие клиента и сервера.)

Если сервер печати не поддерживает LPD-соединения, необходимо использовать более сложную конфигурацию. Например, сервер может быть настроен для обработки заданий на печать, передаваемых средствами SMB/CIFS или AppleTalk. В таком случае вам надо создать сценарий, который обрабатывал бы задания на печать, и вызывать его с помощью опции `if`. Примеры подобных решений приведены в документации на Samba и Netatalk.

Настройка сервера LPRng

С точки зрения пользователя система печати LPRng работает так же, как и BSD LPD. Это вполне закономерно, так как средства LPRng были разработаны для замены BSD LPD. LPRng использует файл `/etc/printcap`, в котором содержится такая же информация, как и в одноименном файле BSD LPD. Однако средства контроля доступа к серверу печати в LPRng реализованы совершенно по-другому. Вместо списка клиентов, которым разрешен доступ, в LPRng используется гораздо более сложная система печати, для управления которой служит файл `/etc/lpd.perms`.

Редактирование файла /etc/lpd.perms

Файл `/etc/lpd.perms` управляет доступом к системе печати в целом. Файлы `lpd.perms` могут находиться также в каталогах спулинга для отдельных очередей (`/var/spool/lpd/`*`имя_очереди`*). Если такие файлы присутствуют, они осуществляют контроль за конкретными очередями, в то время как в файле `/etc/lpd.perms` указываются глобальные опции.

Независимо от расположения файла lpd.perms, в нем содержатся пять типов записей. Комментарии начинаются с символа #. В отличие от файла /etc/hosts.lpd, вы можете включать комментарии в строку после основной команды. Остальные четыре типа записей приведены ниже.

```
DEFAULT ACCEPT
DEFAULT REJECT
ACCEPT [ ключ = значение[,значение]* ]*
REJECT [ ключ = значение[,значение]* ]*
```

Первые два типа записей задают политику системы по умолчанию, т. е. общие правила по предоставлению или запрету доступа. В большинстве пакетов LPRng, поставляемых в составе дистрибутивных версий Linux, в файле /etc/lpd.perms содержится строка DEFAULT ACCEPT. Для сравнения, пакеты BSD LPD по умолчанию разрешают доступ только для узла localhost, или 127.0.0.1 (т. е. для компьютера, на котором выполняется данное программное обеспечение). Таким образом, при настройке системы печати LPRng желательно уточнить политику доступа, используя для этого опции ACCEPT и REJECT.

Опции ACCEPT и REJECT задают типы обращений, которые сервер должен соответственно принимать или отвергать. За именем опции следует один из ключей, указанных в столбце Key табл. 9.2, и значения, заданные в остальных столбцах этой таблицы. Столбец Connect определяет способность устанавливать соединения. Столбцы Job Spool и Job Print указывают на возможность передавать задание спулеру и выводить их на печать. В столбцах lpq, lprm и lpc указано, могут ли выполняться задачи, которыми в обычных условиях управляют утилиты с этими именами. В большинстве случаев наличие одной из этих возможностей означает наличие их всех (по крайней мере, тех, которые имеют смысл в конкретном контексте). Некоторые значения несовместимы с определенными ключами (они отмечены в табл. 9.2 символом –). Для того чтобы изменить значение на обратное, надо указать перед ним NOT. IP-адрес может определять всю сеть, после него надо указать символ / и задать маску подсети.

Опции, предназначенные для контроля доступа, являются достаточно сложными, поэтому имеет смысл пояснить их на конкретных примерах. Рассмотрим следующие строки, входящие в состав стандартного файла /etc/lpd.perms:

```
ACCEPT SERVICE=M SAMEHOST SAMEUSER
ACCEPT SERVICE=M SERVER REMOTEUSER=root
REJECT SERVICE=M
```

В этих трех строках указано, кто может использовать утилиту lprm для удаления заданий. В каждой строке содержится опция SERVICE=M, которая означает, что строка соответствует функциям lprm. Это соответствие можно проследить по строке SERVICE табл. 9.2. В первой из указанных трех строк содержатся также опции SAMEHOST и SAMEUSER, которые указывают на то, что команда принимается только в том случае, если она передана с того же компьютера, что и задание на печать, и от того же пользователя, который является владельцем этого задания. В состав второй строки включены опции SERVER и REMOTEUSER=root. Они означают, что пользователь, зарегистрированный на сервере как root, имеет право удалять задания. Последняя строка запрещает обработку прочих запросов, поступающих от lprm. (LPRng просматривает файл lpd.perms до тех пор, пока не будет найдена опция, которая соответствовала бы поступившей команде. В данном случае две записи ACCEPT SERVICE=M предшествуют

Таблица 9.2. Ключи и их значения, используемые при формировании файла *lpdperms*

Key	Connect	Job Spool	Job Print	**lpq**	**lprm**	lpc
SERVICE	X	R	P	Q	M	C или S
USER	—	Имя пользователя	Имя пользователя	Имя пользователя	Имя пользователя	Имя пользователя
HOST	Удаленный узел	Имя узла	Имя узла	Имя узла	Имя узла	Имя узла
GROUP	—	Имя пользователя	Имя пользователя	Имя пользователя	Имя пользователя	Имя пользователя
IP	IP-адрес удаленного узла	IP-адрес узла	IP-адрес узла	Р-адрес удаленного узла	IP-адрес узла	IP-адрес узла
PORT	Номер порта	Номер порта	—	Номер порта	Номер порта	Номер порта
REMOTEUSER	—	Имя пользователя	Имя пользователя	Имя пользователя	Имя пользователя	Имя пользователя
REMOTEHOST	Удаленный узел	Удаленный узел	Узел	Удаленный узел	Удаленный узел	Удаленный узел
REMOTEGROUP	—	Имя пользователя	Имя пользователя	Имя пользователя	Имя пользователя	Имя пользователя
REMOTEIP	IP-адрес удаленного узла	IP-адрес удаленного узла	IP-адрес узла	IP-адрес удаленного узла	IP-адрес удаленного узла	IP-адрес удаленного узла
CONTROLLINE	—	Шаблон	Шаблон	Шаблон	Шаблон	Шаблон
PRINTER	—	Имя принтера	Имя принтера	Имя принтера	Имя принтера	Имя принтера
FORWARD	—	—	—	—	—	—
SAMEHOST	—	—	—	—	—	—
SAMEUSER	—	—	—	—	—	—
SERVER	—	—	—	—	—	

записи REJECT SERVICE=M, поэтому опции ACCEPT имеют преимущество перед опцией REJECT.)

Как было сказано ранее, во многих случаях система LPRng по умолчанию настраивается так, чтобы она принимала обращения с любого узла. Подобная конфигурация недопустима с точки зрения безопасности, так как любой внешний пользователь может запустить задание на печать, в результате чего будут расходоваться бумага и ресурсы принтера. Кроме того, при обнаружении недостатков в защите LPRng неограниченный доступ к этой системе предоставит дополнительную возможность для взлома принтера. Исходя из этих соображений возможности пользователей по обращению к серверу печати необходимо ограничить. Сделать это можно с помощью брандмауэра (вопросы настройки брандмауэра будут рассмотрены в главе 25). Кроме того, я настоятельно рекомендую вам принять дополнительные меры по защите самой системы LPRng. Предположим, что вы настраиваете сервер печати, который должен обрабатывать задания, переданные из сети 172.22.0.0/16, и с компьютера, на котором установлен сервер; обращения с других узлов должны отвергаться. Сделать это можно с помощью следующих записей:

```
ACCEPT SERVICE=X SERVER
REJECT SERVICE=X NOT REMOTEIP=172.22.0.0/16
```

Данные строки ограничивают способность устанавливать соединения с сервером, а следовательно, возможность передавать задания на печать и выполнять прочие действия. Первая строка разрешает устанавливать соединения при обращении с сервера (эти обращения поступают с интерфейса, который имеет адрес 127.0.0.1, поэтому использование REMOTEIP=127.0.0.1 вместо SERVER привело бы к аналогичному результату). При отсутствии этой строки следующая запись блокировала бы обращения с локального компьютера, что в данном случае нежелательно. Вторая строка отвергает все попытки установить соединения, кроме тех, которые предпринимаются компьютерами, принадлежащими сети 172.22.0.0/16. Если бы у вас возникла необходимость принимать обращения из нескольких сетей, вам бы пришлось включить перед опцией REJECT еще одну опцию ACCEPT и указать для нее адрес дополнительной сети, компьютеры которой имели бы право устанавливать соединения с сервером.

Указание LPRng-сервера на стороне клиента

Файл /etc/printcap в системе LPRng используется аналогично одноименному файлу в системе BSD LPD. В частности, опции, lp, rm и rp, которые обсуждались выше в данной главе, применимы как в BSD LPD, так и в LPRng. Большинство других опций также может присутствовать в обеих системах, но некоторые из них интерпретируются по-разному. Обсуждение этих различий выходит за рамки данной книги.

Системы BSD LPD и LPRng используют протокол LPD, поэтому вы можете сконфигурировать клиент LPRng для печати на сервере BSD LPD и наоборот. Подобное взаимодействие можно также организовать с системой CUPS. Кроме того, CUPS поддерживает расширенный протокол, который может использоваться только в рамках этой системы.

Настройка сервера CUPS

Система печати CUPS, предназначенная для использования в Unix и Linux, обеспечивает чрезвычайно высокую степень гибкости. Вместо того чтобы вносить изменения

в пакет BSD LPD (что по сути надо было сделать при создании LPRng), разработчики CUPS создали полностью новый набор базовых средств, поддерживающих вывод на принтер в различных системах, в том числе в Linux. Часть этих базовых средств предназначена для обеспечения совместимости, поэтому пользователи могут задавать привычные им команды для вывода данных на печать. Клиенты CUPS могут работать с серверами LPD, а клиенты LPD, в свою очередь, могут передавать задания на сервер CUPS. Кроме того, в системе CUPS реализована поддержка нового протокола IPP, который базируется на протоколе HTTP, используемом Web-серверами и броузерами. В системе CUPS можно передавать на сервер информацию о типе файла, что упрощает выбор принтера; для описания возможностей принтеров могут использоваться файлы PPD (PostScript Printer Description); средства просмотра принтеров дают возможность клиенту искать нужные принтеры в сети; при этом не требуется настройка клиентской системы для работы с конкретным устройством. При условии повсеместного применения CUPS перечисленные свойства системы существенно упростят конфигурацию как локальных, так и сетевых средств печати.

Работу с CUPS усложняет одна особенность этой системы: конфигурационные файлы существенно отличаются от файлов, используемых в системах BSD LPD и LPRng. Даже если вы хорошо знакомы с данными системами, это не поможет вам при работе с CUPS; средства ее настройки придется изучать специально. Если вы предпочитаете работать с инструментами, предоставляющими графический интерфейс, можете воспользоваться KUPS (`http://cups.sourceforge.net/kups/`) и ESP Print Pro (`http://www.easysw.com/printpro/`). CUPS также можно настраивать, пользуясь Web-броузером, для этого надо запустить броузер на локальном компьютере и обратиться по адресу `http://localhost:631`.

Подробное описание конфигурации системы печати CUPS выходит за рамки данной книги. Предполагается, что вы уже умеете выполнять действия, необходимые для реализации минимальных возможностей локальной очереди. В данном разделе рассматриваются только опции, необходимые для настройки сетевых средств печати. Дополнительная информация о работе системы CUPS находится по адресу `http://www.cups.org/sam.html`.

Редактирование файла /etc/cups/cupsd.conf

Работой сервера CUPS управляет файл `/etc/cups/cupsd.conf`. Поскольку система CUPS позаимствовала многие средства сервера HTTP, структура ее конфигурационного файла напоминает соответствующий файл Apache (он будет рассмотрен в главе 20). При работе CUPS также применяются другие конфигурационные файлы, в частности `/etc/cups/printers.conf` и `/etc/cups/classes.conf`, которые описывают соответственно принтеры и группы принтеров. Для редактирования обоих файлов используется инструмент `lpadmin`, а данные в файле `cupsd.conf` рекомендуется подготавливать вручную.

Файл `cupsd.conf` содержит набор директив, с помощью которых задаются характеристики сервера, например, определяется имя сервера и расположение файлов протоколов. Ниже описаны наиболее важные директивы, определяющие работу сетевого сервера печати.

- Allow. За этой директивой следуют ключевое слово from, идентификатор All или None, имя узла (в котором может содержаться звездочка, задающая групповую операцию), частичный или полный IP-адрес или IP-адрес с указанием маски сети. Независимо от формы записи, значение данной директивы определяют компьютеры, которые имеют право доступа к серверу. Чтобы разрешить доступ для нескольких компьютеров или групп компьютеров, вы можете использовать несколько директив Allow. Данная директива должна находиться в составе директивы Location.

- AuthClass. Директива`AuthClass` может принимать значения Anonymous (значение по умолчанию), User, System и Group. Anonymous означает, что аутентификация клиентов не должна выполняться; в этом случае система печати работает подобно системе BSD LPD. Остальные три значения требуют от клиента указания пользовательского имени и пароля. Значение System требует, чтобы пользователь был членом группы sys, заданной с помощью директивы SystemGroup. Если указано значение Group, пользователь должен принадлежать группе, имя которой определено посредством директивы AuthGroupName.

- `BrowseAddress`. Средства просмотра принтеров CUPS лучше всего работают в том случае, если информация о принтерах, доступных в сети, собрана на центральном сервере. Этот сервер можно задать с помощью директивы BrowseAddress. В качестве ее значения задается доменное имя или IP-адрес, а также номер порта, например 192.168.23.34:631. (Номер порта 631 чаще всего используется для выполнения различных операций с системой CUPS.) По умолчанию принимается значение 255.255.255.255:631, т. е. широковещательный адрес локальной сети и порт 631.

- BrowseAllow. Для того чтобы клиент мог выполнять просмотр принтеров, сервер должен принимать от него специальные пакеты. Директива BrowseAllow (за ней следуют ключевое слово `from` и частичное или полное доменное имя либо адрес узла) задает компьютеры, с которых сервер должен принимать данные пакеты. По умолчанию прием пакетов разрешен со всех компьютеров.

- `BrowseDeny`. Данная директива выполняет действие, противоположное директиве BrowseAllow. С ее помощью формируется "черный список" клиентов или сетей.

- BrowseOrder. Если вы используете и BrowseAllow, и BrowseDeny, директива BrowseOrder позволяет определить порядок применения указанных директив. Она может быть задана в виде BrowseOrder Allow, Deny или BrowseOrder Deny, Allow.

- `BrowseInterval`. Данная директива задает время в секундах между запросами на просмотр. Значение 0 запрещает передачу запросов. Значение данной опции должно быть меньше, чем значение опции BrowseTimeout, в противном случае принтеры будут периодически исчезать из локального списка просмотра.

- BrowsePoll. Данная директива позволяет задавать имя или IP-адрес сервера печати для опроса принтеров. Чтобы опрашивать несколько серверов, вы можете указать данную директиву несколько раз.

- BrowsePort. По умолчанию для просмотра принтеров используется порт 631, но с помощью данной директивы вы можете переопределить это значение.

- BrowseTimeout. По истечении интервала времени, указанного посредством данной директивы, CUPS удаляет информацию о сетевых принтерах. Это значение должно быть больше, чем значение директивы `BrowseInterval`, в противном случае принтеры будут периодически исчезать из списка просмотра CUPS-клиента.

- Browsing. Задавая значение этой директивы, равное On или `Off`, вы можете соответственно разрешать или запрещать просмотр сети. По умолчанию предполагается значение On.

- Deny. Данная директива выполняет действия, противоположные действиям директивы Allow. С ее помощью задаются компьютеры, которым запрещен доступ к серверу. Директива Deny должна присутствовать в составе директивы Location.

- `HostNameLookups`. Данная директива может принимать значения Off, On и Double. Значение `Off` запрещает проверку имен клиентов, On включает проверку имени каждого клиента, если же задано значение Double, проверяется имя клиента, а по полученному имени определяется его IP-адрес. Значение Double обеспечивает защиту от некоторых типов атак, поскольку при этом отвергаются обращения клиентов, для которых некорректно сформированы записи на сервере DNS. По умолчанию предполагается значение `Off`; при этом обеспечивается максимальная производительность и надежность (остальные значения могут приводить к возникновению проблем в случае выхода из строя сервера DNS).

- Listen. Задавая одну или несколько директив Listen, вы можете сообщить CUPS о том, что в процессе взаимодействия должны использоваться лишь некоторые из сетевых интерфейсов. В качестве значения данной директивы задается IP-адрес сетевого интерфейса и номер порта (обычно 631). Например, выражение `Listen 192.168.23.8:631` означает, что компьютер должен использовать интерфейс с адресом 192.168.23.8. С помощью директивы Listen можно учесть все необходимые интерфейсы; в большинстве случаев необходимо указывать также интерфейс с адресом 127.0.0.1.

- Location. Данная директива отличается от остальных; она объединяет ряд других директив и указывает область их применения. Например, в состав Location вы можете включить директивы Allow и Deny, разрешая или запрещая для клиентов обращение к конкретным типам документов (и, следовательно, выполнение определенных типов операций). Началом данной директивы является ключевое слово Location, помещенное в угловые скобки, а окончанием — выражение </Location>. В составе <Location> могут присутствовать опции /admin (действия по администрированию), `/classes` (определение классов принтеров), `/jobs` (определение заданий на печать) и `/printers` (принтеры).

- MaxClients. Директива MaxClients позволяет ограничить число клиентов, которые могут устанавливать соединение с сервером. По умолчанию принимается значение, равное 100.

- Order. Данная директива выполняет действия, аналогичные действиям директивы BrowseOrder, но применяется к директивам Allow и Deny. Выражение Order

Allow, Deny означает, что директива Allow должна применяться перед директивой Deny, а выражение Order Deny, Allow задает обратную последовательность применения этих директив.

- Port. В обычных условиях CUPS ожидает обращение через порт 631, но при необходимости вы можете с помощью данной директивы изменить порт, используемый по умолчанию. Указывая данную директиву многократно, можно задать несколько портов. Заметьте, что директива Port не влияет на номер порта, используемый CUPS для взаимодействия с клиентами и серверами BSD LPD и другими подобными программами.

Файл /etc/cups/cupsd.conf, поставляемый с большинством пакетов CUPS, оставляет сервер совершенно открытым для обращений с внешних узлов. Подготавливая средства CUPS к реальной работе, необходимо ограничить доступ к серверу. Например, приведенная ниже директива блокирует доступ со всех узлов, за исключением компьютера, на котором расположен сервер, а также компьютеров, принадлежащих сети 172.22.0.0/16.

```
<Location /printers>
BrowseAllow from 127.0.0.1
BrowseAllow from 172.22.0.0/16
Allow from 127.0.0.1
Allow from 172.22.0.0/16
</Location>
```

Поскольку в директиве Location указана опция /printers, она не блокирует полностью доступ к серверу. Например, выполнение административных задач (опция /admin) и доступ к информации о заданиях на печать (опция /jobs) разрешены и для других систем. Настраивая CUPS, необходимо ограничить все виды доступа и даже продублировать ограничения, сконфигурировав соответствующим образом средства фильтрации пакетов (они будут рассматриваться в главе 25).

Получение заданий от клиентов BSD LPD и LPRng

Рассмотренные выше директивы, предназначенные для включения в файл /etc/cups/cupsd.conf, в основном имеют отношение к клиентам, поддерживающим IPP. Этот протокол не использует ни BSD LPD, ни LPRng; данные системы применяют в работе протокол LPD. (В настоящее время ведутся работы по включению средств поддержки IPP в состав LPRng, но они еще не завершены.) Как было сказано выше в данной главе, серверы печати CUPS могут получать задания на печать от клиентов, использующих протокол LPD. Для этого в состав CUPS включена вспомогательная программа cups-lpd.

Программа cups-lpd не может выполнять функции независимого сервера, ее необходимо сконфигурировать для работы с суперсервером inetd или xinetd (суперсерверы и взаимодействие с ними рассматривались в главе 4). Программа cups-lpd обычно располагается в каталоге /usr/lib/cups/daemon, а соответствующая ей запись в конфигурационном файле /etc/inetd.conf выглядит следующим образом:

```
printer stream tcp nowait lp /usr/lib/cups/daemon/cups-lpd \
    cups-lpd
```

Учитывая различия между inetd и xinetd, описанные в главе 4, вы можете легко сконфигурировать cups-lpd для работы с xinetd. В некоторых случаях пакет CUPS поставляется уже сконфигурированным для взаимодействия с клиентами BSD LPD, в этом случае нет необходимости предпринимать специальные меры по его настройке.

ВНИМАНИЕ В CUPS не предусмотрены никакие средства для контроля доступа клиентов, использующих протокол LPD. При передаче заданий от таких клиентов используется адрес сервера, и директивы в файле /etc/cups/cupsd.conf не оказывают никакого влияния на ход взаимодействия. Для того чтобы ограничить доступ к серверу CUPS с внешних узлов, надо настроить соответствующим образом брандмауэр или использовать другие механизмы ограничения доступа.

Определение сервера CUPS на стороне клиента

Для добавления принтеров к системе CUPS используется утилита lpadmin, вызываемая из командной строки или доступная посредством специального графического интерфейса. Кроме того, эта задача может решаться с помощью Web-броузера; для этого надо запустить Web-броузер на компьютере, на котором расположен сервер, и обратиться с его помощью по URL http://localhost:631. (Вы можете также запустить броузер и на другом узле, для которого разрешено выполнение задач администрирования, в этом случае вместо localhost необходимо указать имя компьютера, на котором выполняется сервер CUPS.) Используя любой из описанных здесь способов, вы можете добавлять или удалять принтеры или выполнять другие действия по администрированию сервера. Ниже приведен пример вызова утилиты Ipadmin.

```
# Ipadmin -p Имя_Принтера -E -v lpd://имя.сервера/имя_очереди\
  -m ppdfile.ppd
```

В данном примере *Имя_Принтера* — это имя очереди печати, используемой локально, *имя.сервера* — это имя узла, на котором установлен сервер печати, а *имя_очереди* — имя очереди на этом сервере. Поскольку в качестве протокола указано имя lpd, доступ к очереди печати осуществляется посредством протокола LPD. Для использования другого протокола надо вместо lpd задать имя ipp. (Аналогичным способом вы можете задать локальную очередь, но в качестве значения опции -v необходимо указать parallel:/dev/lp0 или задать идентификатор другого локального устройства.) Опция -т определяет PPD-файл для принтера, при этом CUPS может передавать приложению информацию о возможностях принтера. В состав большинства пакетов включается набор файлов PPD; они располагаются в каталоге /usr/share/cups/model. Файлами PPD снабжаются также многие принтеры PostScript. Для получения файла PPD вы можете воспользоваться списком драйверов по адресу http://www.linuxprinting.org/driver_list.cgi. Щелкните на имени драйвера Ghostscript, затем выберите модель вашего принтера в области CUPS-O-Matic и щелкните на Generate CUPS PPD. Через некоторое время вы получите файл PPD, описывающий возможности вашего принтера. Как сказано в комментариях, этот автоматически сгенерированный файл не свободен от недостатков, более того, не исключено, что он вовсе не будет работать. Поэтому, если это возможно, лучше использовать файлы PPD, поставляемые производителями принтеров.

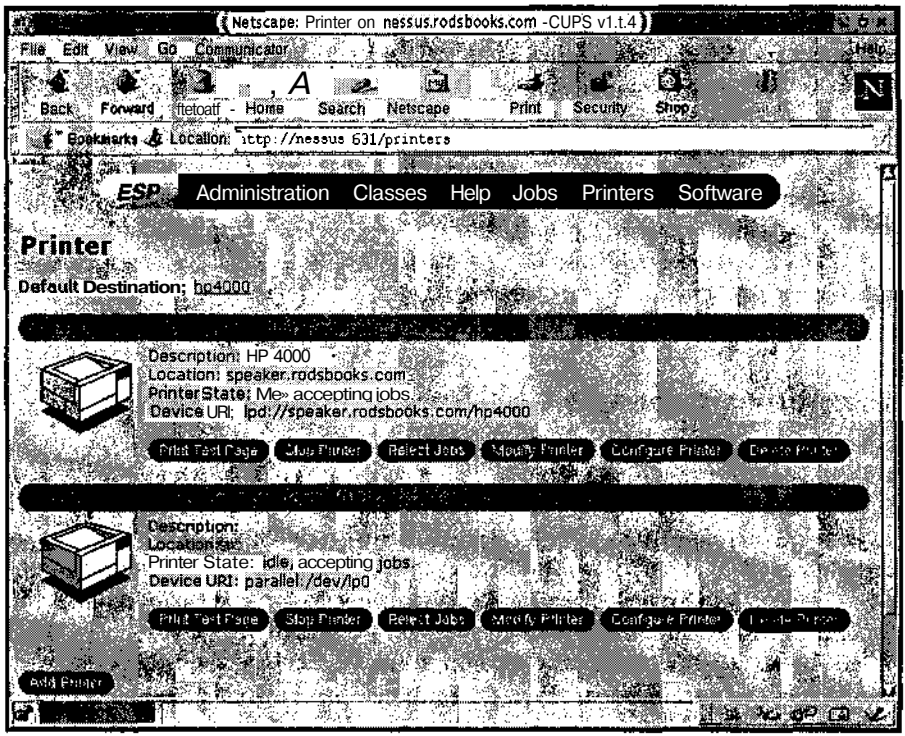


Рис. 9.1. Web-интерфейс CUPS упрощает настройку как локальных, так и сетевых принтеров

СОВЕТ ✔ Если вы сконфигурировали клиент и сервер для просмотра принтеров, вам нет необходимости специально указывать принтеры IPP. Клиент автоматически получит список доступных принтеров. Описанная операция нужна лишь при использовании очередей печати LPD.

Если вы хотите модифицировать существующую очередь печати, для этого также можно применить инструмент `lpadmin`. Задайте исходное имя и необходимые опции. Например, чтобы преобразовать локальную очередь в сетевую, надо указать опцию -v и задать ее новое расположение.

Если вы предпочитаете инструменты с графическим пользовательским интерфейсом, можете воспользоваться средствами Web. В стандартном пакете CUPS предусмотрена возможность выполнения операций администрирования по протоколу HTTP. Соответствующая Web-страница показана на рис. 9.1. Для того чтобы приступить к администрированию, надо ввести URL компьютера, на котором установлен сервер, и указать порт 631. Затем CUPS запросит у вас имя и пароль администратора. После этого вы сможете выбрать конкретный пункт, например Do Administration Tasks или Manage Printers. Результат выбора Manage Printers показан на рис. 9.1. На этой странице отображается информация о двух принтерах. Первый из них, `hp4000`, используется по умолчанию и представляет собой принтер LPD. Второй, lexmark, подключен к параллельному порту. Щелкнув на кнопке Modify Printer, вы можете изменить базовые установки, например

имя сервера, а щелкнув на кнопке Configure Printer, — задать установки для конкретного принтера, например размер страницы.

Резюме

Традиционно в Linux использовалась система печати BSD LPD, но в последние годы она перестала соответствовать требованиям, предъявляемым к подобным системам. Существуют альтернативные системы, которые в последние годы все чаще включаются в состав дистрибутивных пакетов Linux. К таким системам относятся LPRng и CUPS. В них улучшены средства защиты, а в системе CUPS реализованы инструменты администрирования, предоставляющие Web-интерфейс.

Настраивая компьютер для выполнения функций сервера печати, обязательно надо учитывать требования безопасности. Лучше всего использовать средства фильтрации пакетов, сконфигурировав их так, чтобы обращаться к портам 515 и 631 могли только те компьютеры, для которых разрешено взаимодействие с сервером печати (порты 515 и 631 используются соответственно при работе с помощью протоколов LPD и IPP). Кроме того, в качестве дополнительных мер защиты следует применять средства, реализованные в конкретных системах печати. LPRng предоставляет возможность управлять взаимодействием посредством протокола LPD, а CUPS обеспечивает контроль при использовании средств IPP.

Глава 10

Служба времени

Настраивая компьютер, администратору приходится устанавливать показания системных часов. Как правило, встроенные аппаратные таймеры работают не точно, кроме того, погрешность их различается в разных системах. Если сеть проработает несколько недель, то вы увидите, что системные часы, точно установленные вначале, показывают совершенно разное время (если погрешность велика, то расхождение можно заметить уже через несколько часов). В процессе перехода на летнее и зимнее время также возникают проблемы; при этом вам надо решить, следует ли вручную переводить часы или предоставить системе сделать это самостоятельно. Ситуация осложняется еще больше, если на одном компьютере установлено несколько операционных систем. Другими словами, для поддержки точного времени в сети приходится затрачивать время и усилия, которые могли бы быть направлены на решение более важных задач. Для того чтобы избавить администратора от необходимости постоянно следить за показаниями системных часов и корректировать их, разработаны специальные программы, которые называются временными серверами. Эти программы предоставляют клиентам сведения о точном времени, поэтому, для того, чтобы на всех компьютерах были установлены одинаковые показания системных часов, достаточно инсталлировать центральный временной сервер и сконфигурировать клиенты для работы с ним. Временной сервер можно также синхронизировать с одним из внешних серверов, которые получают сведения от эталонных источников. В результате такой синхронизации достигается точная установка времени во всей сети.

Использование временного сервера

Временной сервер поддерживается в сети для того, чтобы клиенты могли получать информацию о точном времени. На основании этой информации клиенты могут устанавливать свои системные часы так, чтобы они показывали время, близкое к эталонному (выражение "близкое к эталонному" может означать, что время на системных часах будет отличаться от эталонного всего на несколько миллисекунд). Это позволит избежать проблем, связанных с неодинаковой установкой системного времени. Так, например, если показания системных часов на разных компьютерах различаются, может возникнуть ситуация, при которой клиент запишет файл на сервер, а обратившись к этому же файлу через минуту, обнаружит, что время его создания опережает текущее время на две мину-

ты. Разница во времени может существенно затруднить интерпретацию записей в файлах протоколов. Работа некоторых инструментов, например Kerberos, непосредственно основана на предположении о том, что показания часов клиента и сервера совпадают. Если в сети используются подобные средства, наличие в ней временного сервера обязательно.

Работа программы, реализующей временной сервер, может показаться несколько необычной. Дело в том, что эта программа выполняет функции как сервера, так и клиента. В частности, она не только предоставляет сведения клиентам, но и корректирует показания часов того компьютера, на котором она выполняется. Если в вашей сети есть несколько машин, работающих под управлением UNIX или Linux, вы можете настроить одну из них так, чтобы она получала сведения о точном времени от внешнего сервера, а остальные сконфигурировать для получения данных от первого компьютера. Такая конфигурация снижает нагрузку на внешние серверы. Существуют серверы, специально выделенные для поддержки информации о времени и предоставления ее клиентам. На некоторых из таких серверов для отсчета времени применяются атомные часы, а на других показания часов синхронизируются по радиосигналу, который передает официальная служба времени. Если вы не хотите устанавливать полнофункциональный временной сервер на клиентских компьютерах, можете использовать простую программу, которая выполняет лишь функции клиента. Чтобы показания часов оставались корректными, вам необходимо обеспечить периодический запуск этой программы.

Настройка сервера NTP

Из протоколов, обеспечивающих работу временных серверов, наиболее популярен NTP (Network Time Protocol — сетевой протокол времени), который описан в документе RFC 1305 (`http://www.ietf.org/rfc/rfc1305.txt`).Рассмотрению более старых версий этого протокола посвящены документы RFC 958, RFC 1059 и RFC 1119. На момент написания данной книги, т. е. в 2002 году, последней версией NTP была версия 4, но версия 3 продолжала широко использоваться. Основной Web-узел NTP располагается по адресу `http://www.eecis.udel.edu/~ntp/`. NTP поддерживает иерархическую структуру временных серверов, в которой сервер, непосредственно получающий данные об эталонном времени, предоставляет их серверам, взаимодействующим с ним; те, в свою очередь, обслуживают другие серверы и т. д. до тех пор, пока информация о времени не доставляется клиентам. Серверы и клиенты NTP разработаны для различных операционных систем, в частности для Linux. Чтобы настроить сервер NTP для работы в системе Linux, необходимо отредактировать лишь один конфигурационный файл. Контроль за действиями сервера осуществляется с помощью специальных инструментов. На компьютерах, находящихся на самом нижнем уровне иерархии NTP, могут быть запущены клиентские программы, которые также просты в использовании.

Существует упрощенный вариант NTP, который называется SNTP (Simple NTP — простой NTP). Клиенты SNTP осуществляют синхронизацию времени, взаимодействуя с серверами NTP.

Функционирование временных серверов

Работа временных серверов начинается с получения сведений о времени от официальных источников. Эти сведения получаются путем считывания показаний атомных часов,

приема специальных радиосигналов и т. д. Служба GPS (Global Positioning System — глобальная система позиционирования) принимает временные сигналы со спутников, поэтому может быть использована в качестве официального источника данных о времени. (Информацию об устройствах отсчета времени можно получить, обратившись по адресу `http://www.eecis.udel.edu/~ntp/hardware.html`.)

Атомные часы, устройства приема радиосигналов и прочее оборудование принято называть *эталонными временными серверами,* или *серверами уровня 0.* Эти серверы не поддерживают сетевые соединения (они взаимодействуют с компьютерами через последовательные порты, и для обмена данными с ними требуются дополнительные устройства). Компьютеры, которые синхронизируют свои системные часы, используя такие устройства, называются *серверами уровня 1.* Может показаться, что на них поддерживается наиболее точное время в Internet, однако исследования показали, что приблизительно треть из них имеют погрешность в секунду и больше. Компьютеры, которые используют для синхронизации серверы уровня 1, называются серверами уровня 2 и т. д.

Процедура синхронизации предполагает обмен клиента с сервером несколькими пакетами данных (как правило, каждые пять минут передаются как минимум пять пакетов). Клиент передает серверу пакет, содержащий текущие показания своих системных часов. В ответ сервер передает клиенту тот же пакет. Сравнивая время в момент передачи и в момент приема, клиент может оценить задержку, связанную с пересылкой данных по сети, и учесть ее при коррекции значения таймера. Клиент может быть настроен для работы с несколькими серверами. В этом случае, сравнивая показания времени, задержку, вызванную передачей данных, и другие факторы, он может выбрать из этих серверов наиболее пригодный для себя.

Полнофункциональный сервер NTP работает постоянно и время от времени опрашивает вышестоящий сервер (первоначально обращения осуществляются каждые 64 секунды, но при некоторых вариантах конфигурации системы интервал между обращениями может увеличиться до 1024 секунд). Сервер NTP корректирует показания часов компьютера, на котором он выполняется, различными способами. Большие расхождения во времени (порядка секунды или больше) сначала игнорируются — сервер считает, что они могут быть вызваны ошибками при обмене данными. Но если такая ситуация сохраняется в течение некоторого времени, NTP компенсирует ошибку; он либо непосредственно устанавливает требуемое значение времени, либо ускоряет или замедляет ход системных часов до тех пор, пока системное время и время внешнего сервера не сравняются. (Процедура ускорения и замедления хода называется *подстройкой* системных часов.) Сервер NTP также поддерживает специальный файл (обычно это `/etc/ntp/drift`, `/var/state/ntp.drift` или другой файл с подобным именем), в котором он хранит данные об ошибке или о "дрейфе" системного таймера. Информация в этом файле помогает серверу компенсировать ошибку системных часов в том случае, если компьютер на длительное время остается выключенным или если вышестоящий сервер NTP не доступен.

СОВЕТ ✔ Если ошибка превышает 1000 секунд, сервер NTP прекращает свою работу, так как предполагает, что подобная ситуация требует непосредственного вмешательства системного администратора. Поэтому перед запуском сервера NTP необходимо установить хотя бы приблизительное значение системного времени. Для установки системных часов перед запуском сервера NTP можно также использовать программу `ntpdate`, работа которой будет рассмотрена далее в этой главе. В некоторых случаях вызов `ntpdate` предусматривают в сценарии запуска сервера NTP.

В обычных условиях сервер NTP, работающий в небольшой сети, использует для синхронизации данные, предоставляемые тремя внешними серверами. (Число три выбрано произвольно, при желании вы можете увеличить или уменьшить количество внешних серверов, с которым будет взаимодействовать сервер NTP в вашей сети. Использование трех серверов обеспечивает избыточность данных, требуемую для надежной работы.) Для небольшой сети роль внешних серверов, как правило, выполняют серверы уровня *2*. Количество серверов уровня 1 относительно невелико, и они обычно используются для синхронизации серверов NTP, которые обслуживают сотни клиентов. Клиентские компьютеры в небольшой сети практически постоянно обращаются к серверу уровня 3 для получения сведений о текущем времени. Вместо серверов на компьютерах могут быть установлены клиенты NTP, в этом случае опрос сервера может производиться не так часто. Если ваша сеть насчитывает несколько десятков компьютеров и для их работы необходимо обеспечивать точные показания системных часов, в ней имеет смысл установить два сервера NTP уровня 3. Это позволит избежать проблем, если сервер выйдет из строя или станет работать ненадежно. Если по каким-либо причинам необходимо, чтобы системные часы клиентских компьютеров работали особенно точно, вам следует рассмотреть возможность приобретения специального устройства и организации в вашей сети сервера уровня 1. Необходимое для этого устройство может стоить несколько сотен долларов.

В системе NTP используется универсальное время (Coordinated Universal Time — UTC), которое практически совпадает с гринвичским временем (Greenwich Mean Time — GMT) без учета переходов на летнее и зимнее время. UTC отличается от GMT лишь некоторыми деталями. В частности, UTC не определяется исходя из скорости вращения Земли, а отсчитывается на основании показаний высокоточных и высоконадежных атомных часов. При необходимости значение UTC можно привести в соответствие со скоростью вращения Земли, прибавляя или вычитая около секунды в день. Локальное время отличается от UTC смещением часового пояса. Кроме того, при определении локального времени также должны учитываться правила перехода на летнее и зимнее время.

Большинство операционных систем, установленных на компьютерах x86, требуют, чтобы системные часы показывали локальное время. Linux может работать с системным таймером, отсчитывающим либо локальное время, либо UTC, а также поддерживает отдельные программные часы, установленные в соответствии с UTC. Если на компьютере установлена только система Linux, желательно применять UTC, так как при этом нет необходимости переводить таймер на летнее и зимнее время. Если же на компьютере кроме Linux инсталлирована Windows (или другая операционная система, использующая локальное время), вам придется установить таймер по локальному времени. Кроме того, при этом возникает проблема при переходе на летнее и зимнее время. Устранить ее можно, запуская средства поддержки временного протокола при загрузке системы. Заметьте,

что преимущества применения NTP в Linux не распространяются на другие системы, так как при коррекции системных часов Linux не изменяет значение аппаратного таймера. Для приведения аппаратного таймера в соответствие с системными часами можно использовать команду hwclock -systohc -localtime; в этом случае на аппаратном таймере устанавливается локальное время. Если на вашем компьютере показания времени хранятся в формате UTC, то при вызове данной команды надо заменить -localtime на -utc.

Временные серверы для Linux

Сервер NTP для работы в Linux реализуется с помощью программы ntp или ее разновидностей: xntp, xntp3 и xntpd. Символ x в начале имени означает "экспериментальный" (experimental), что не совсем верно, так как эти программы успешно используются в течение нескольких лет. В именах программ, содержащихся в пакете NTP 4, символ x отсутствует. В составе большинства версий Linux поставляется версия 4 пакета NTP, но нередко встречается также версия 3.

Большинство пакетов NTP содержат сервер NTP и несколько вспомогательных программ. Компоненты пакета описаны ниже.

- ntpd. Основная программа, реализующая сервер NTP. (В некоторых поставках она называется xntpd.) Как было сказано ранее, несмотря на то, что эта программа считается сервером, она объединяет в себе функции клиента и сервера. Для вышестоящих серверов она является клиентом, а для нижележащих программ — сервером. (Вышестоящим считается сервер с меньшим значением уровня.)

- ntpdate. Данная программа намного проще, чем программа ntpd; она реализует лишь функции клиента. Если поддержка точного времени на компьютере не слишком важна, вы можете установить вместо сервера программу ntpdate и обеспечить ее периодические вызовы. Работа ntpdate будет рассмотрена далее в этой главе.

- ntptrace. В некоторых случаях возникает необходимость проследить источник данных о времени. Данная программа отслеживает путь от локального компьютера к серверу NTP, используемому для синхронизации, и далее вверх по дереву NTP. Такая информация может быть полезной для диагностики системы.

- ntpq. Данная программа осуществляет NTP-мониторинг. Она будет рассмотрена далее в этой главе.

- xntpdc. Эта программа также предназначена для мониторинга и управления системой NTP. Она позволяет выполнять более сложные операции, чем ntpq.

Помимо NTP, в Linux для согласования времени могут быть использованы и другие программы. Одной из таких программ является rdate, которая по своим возможностям напоминает ntpdate; она используется для однократной установки системных часов. Программа rdate входит в состав некоторых дистрибутивных пакетов, но в ряде пакетов она отсутствует. Эта программа уступает по точности ntpdate. Если ntpdate может обеспечивать точность порядка нескольких миллисекунд, то rdate имеет точность около секунды.

Структура конфигурационного файла ntp.conf

Для настройки средств NTP используется файл `ntp.conf`, который обычно размещается в каталоге /etc. Как и во многих других конфигурационных файлах, строки, содержащие комментарии, начинаются с символа #, а в остальных строках задаются различные опции NTP. Наиболее важными из этих опций являются следующие.

- server *адрес* [key *ключ*] [version *номер*] [prefer]. Данная опция задает имя сервера, который используется для синхронизации показаний времени с помощью протокола NTP. В качестве адреса может быть задан IP-адрес или имя узла. При необходимости в файл `ntp.conf` можно включить несколько опций server, в результате ваш сервер NTP установит соединение с каждым из указанных серверов и выберет для синхронизации наилучший из них. В составе данной опции может задаваться дополнительная информация. Значение, следующее после key, определяет ключ аутентификации, оно указывается, если доступ к серверу ограничен. Номер версии сообщает о том, какая версия протокола должна быть использована при взаимодействии. Ключевое слово prefer указывает, что данный сервер предпочтительнее других.

- fudge *адрес* stratum номер. Данная опция в основном используется для того, чтобы указать, что сервер 127.127.1.0 (локальные системные часы) должен интерпретироваться как сервер уровня 7 — сервер NTP с самым низким приоритетом. Это позволяет серверу продолжать работу даже в том случае, если другие серверы недоступны.

- driftfile *имя_файла*. Указанный в качестве значения данной опции файл включает информацию, которая используется при возобновлении работы после длительного отключения компьютера. Содержимое данного файла позволяет серверу NTP компенсировать погрешность внутреннего таймера и увеличить точность при работе в автономном режиме.

- broadcast *адрес* [key *ключ*] [version номер] [ttl номер]. Если вы укажете данную опцию, сервер будет периодически передавать в широковещательном режиме данные о текущем времени. Информация будет передаваться по сети, адрес которой является значением данной опции (это может быть также адрес группового вещания 224.0.1.1). Использование широковещательного адреса позволяет уменьшить трафик в больших сетях, в которых многие серверы NTP работают в качестве клиентов.

- broadcastclient [yes | no]. Данная опция указывает серверу NTP на то, что он должен принимать широковещательные сообщения от других локальных серверов NTP.

В файле `ntp.conf` могут быть указаны и другие опции, с помощью которых задаются специальные функции. Информацию о них можно получить в документации, представленной в формате HTML, которая поставляется в составе пакета и обычно находится в каталоге `/usr/share/doc/xntp-`*версия*.

Файл `ntp.conf`, поставляемый в составе дистрибутивного пакета, практически обеспечивает работу сервера. Вам надо лишь добавить одну или несколько опций server,

указывающих на серверы NTP. К выбору сервера надо подходить очень внимательно. Если сервер, используемый для синхронизации, расположен далеко или работает ненадежно или синхронизирован с помощью некорректного источника, показания системных часов на компьютеров вашей сети будут неточными. Как было сказано ранее, для небольшой сети в качестве источника синхронизирующих данных целесообразно выбирать сервер уровня 2. Этот вопрос интенсивно обсуждается в сети; материалы дискуссий вы можете найти по адресу `http://www.eecis.udel.edu/~mills/ntp/servers.htm`. В конце этого документа даны ссылки на Web-страницы, содержащие списки временных серверов уровней 1 и 2. Постарайтесь использовать для синхронизации сервер, расположенный ближе других. Заметьте, что топология сетей отличается от географического размещения компьютеров. Так, например, компьютер, расположенный на другом континенте, может быть "ближе" к локальной машине, чем компьютер, находящийся в часе езды от нее.

Для того чтобы сравнить время передачи данных при обмене с различными серверами NTP, можно воспользоваться утилитой `ping`. Для синхронизации желательно использовать тот сервер, от которого ответы на ping-пакеты приходят быстрее.

Если в списке указано, что, прежде чем использовать некоторый сервер, надо оповестить об этом оператора, не забудьте послать соответствующее сообщение. Возможно, вам имеет смысл рассмотреть в качестве претендентов на роль источника данных о времени менее известные серверы, расположенные ближе к вашей сети. Подобные серверы поддерживают многие крупные организации, в том числе провайдеры Internet. Если вы устанавливаете временной сервер для отдела, обсудите этот вопрос с системным администратором, обслуживающим сеть всей организации, и с провайдером.

Если вы приобретете GPS либо другое устройство, позволяющее принимать эталонные данные времени, вы можете установить в своей сети сервер уровня 1. Для работы с таким оборудованием вам понадобятся специальные драйверы. Эти драйверы устанавливают принадлежность устройства сети 127.127.0.0/16, в результате для работы с ним можно использовать обычную опцию `server`. Дополнительную информацию об использовании указанных устройств вы можете найти в документации на драйверы Linux. Сведения о производителях устройств, позволяющих получать сигналы эталонного времени, приведены в документе `http://www.eecis.udel.edu/~ntp/hardware.html`.

После редактирования ntp . conf надо перезапустить сервер NTP. Сделать это можно с помощью сценария запуска SysV (подробно вопрос использования сценариев SysV обсуждался в главе 4). Если в сценарии запуска не предусмотрен вызов `ntpdate`, перезапуск `ntpd` не приведет к резкому изменению показаний системных часов, даже если компьютер был выключен в течение нескольких минут. Вначале `ntpd` несколько раз сравнит показания системных часов с данными, предоставленными удаленным сервером, а лишь затем предпримет меры для коррекции системного времени. Вопросы контроля операций, выполняемых ntpd, будут обсуждаться в следующем разделе.

Контроль операций NTP

Помимо визуального контроля показаний часов с помощью программы `xclock`, для мониторинга операций NTP часто применяется программа ntpq. После вызова эта программа запрашивает команды, определяющие ее дальнейшую работу. Команды вводятся

в текстовом режиме. В процессе выполнения программа отображает информацию о работе сервера. Некоторые наиболее важные команды ntpq описаны ниже.

- host *имя_узла*. По умолчанию ntpq опрашивает сервер, находящийся на локальном компьютере. Задавая команду host, можно использовать данную программу для проверки любого сервера NTP в сети. Аналогичный результат можно получить, задавая имя целевого узла при вызове ntpq, например ntpq remote.threeroomco.com.
- hostnames [yes | no]. Если вы укажете опцию yes, программа ntpq, сообщая о действиях удаленных компьютеров, будет отображать имена узлов (подобная конфигурация предусмотрена по умолчанию). Опция по указывает на то, что вместо имен должны отображаться IP-адреса. Такой же эффект вызовет опция -n, заданная при вызове программы ntpq.
- ntpversion номер_версии. Данная команда позволяет указать версию протокола NTP, которая будет использоваться при передаче запросов серверу NTP.
- quit. Данная команда задается после окончания работы с программой ntpq и завершает ее выполнение.
- peers. Данная команда предоставляет одно из самых мощных средств диагностики. Она отображает список серверов, с которыми взаимодействует ваш сервер. Если вы предварительно не задали команду host, в этом списке будут содержаться сервер на локальном компьютере и все серверы, указанные в файле ntp.conf. Кроме того, при вызове этой команды будет отображена дополнительная информация, в частности, серверы, используемые для синхронизации; уровень каждого сервера; время последнего обращения к каждому серверу и интервал между обращениями; числовой код, отражающий надежность соединения между компьютерами; задержка, смещение и погрешность синхронизации. В начале каждой записи отображается символ, указывающий на то, каким образом ваш сервер использует данные, предоставляемые другими серверами. Символ + означает, что сервер рассматривался как претендент на роль источника синхронизации, но вместо него был выбран другой сервер; символ * указывает на то, что сервер является вышестоящим по отношению к вашему серверу; символ x определяет "испорченные часы" — сервер, показания которого признаны неверными. Кроме того, ntpq может отображать другие символы, определяющие различные характеристики серверов. Разновидностями команды peers являются lpeers (она может отображать информацию о большем количестве серверов) и opeers (не выводит имена серверов, с которым взаимодействует ваш сервер).
- associations. Данная команда выводит статистику соответствия для каждого сервера. Серверы указываются не с помощью имен или IP-адресов, а посредством идентификаторов соответствия, используемых в других командах. Разновидностями этой команды являются lassociations, passociations и lpassociations.
- readvar *идентификатор_соответствия имя_переменной*. Эта команда позволяет читать содержимое переменной. Она чаще всего применяется при отладке. Синонимом readvar является rv, а mreadvar представляет собой разновидность этой команды.

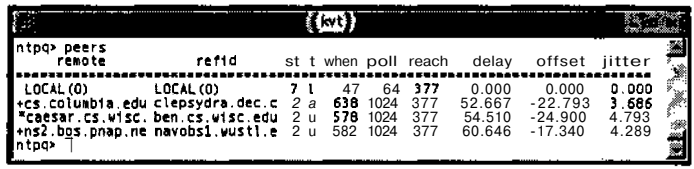


Рис. 10.1. Программа ntpq отображает информацию о состоянии NTP-сервера

- readlist *идентификатор_соответствия*. Данная команда действует подобно readvar, но выводит список всех стандартных переменных. Синонимом readlist является rl, а mreadlist представляет собой разновидность этой команды.

- pstatus *идентификатор_соответствия*. Команда pstatus запрашивает информацию о состоянии системы. Результат выполнения данной команды практически совпадает с результатом команды readlist.

- writevar *идентификатор_соответствия имя_переменной*. Данная команда позволяет изменить значение переменной. Как правило, в ее использовании не возникает необходимости.

Программа ntpq вызывается при первоначальной настройке сервера NTP и при изменении его конфигурации. Кроме того, с ее помощью периодически выполняется контроль за функционированием сервера. На рис. 10.1 показан результат работы программы ntpq; в данном примере эта программа вызвана тогда, когда сервер NTP уже проработал некоторое время. Если вы вызовете ntpq сразу же после запуска ntpd, многие поля останутся пустыми или будут содержать значения, не имеющие смысла (чаще всего нулевые). Если сервер проработает около минуты, все поля будут заполнены реальными значениями, как это показано на рис. 10.1. Символы + и * в начале записей появляются лишь спустя несколько минут, так как для выяснения того, какие из серверов более надежны, требуется определенное время. В течение нескольких минут некоторые значения могут изменяться, а затем они станут стабильными. Если слева от имени сервера отображается символ ×, этот сервер имеет смысл удалить из конфигурационного файла, поскольку, вероятнее всего, он работает некорректно.

Если вы заметите, что показания системных часов изменяются странным образом, имеет смысл вызвать программу ntpq и проверить текущее состояние сервера. Возможно, он не получает синхронизирующих данных из-за изменения IP-адреса сервера или вследствие нарушения работы сети. (Эпизодические сбои при обмене данными по сети не могут серьезно повлиять на работу временного сервера. Он лишь переключится на использование внутреннего таймера, а затем при возобновлении работы сети снова станет действовать в обычном режиме.) Если в течение нескольких минут после запуска ntpd сервер не сможет синхронизировать свои данные с одним из внешних временных серверов, вам следует проверить работу сети. Доступен ли удаленный сервер для пакетов, передаваемых с помощью программы ping? Не блокирует ли брандмауэр запросы NTP? (Возможно, вам придется перенастроить брандмауэр для прохождения пакетов UDP, адресованных на порт 123.) Имеете ли вы право обращаться к удаленному серве-

ру? (Не исключено, что на этом сервере используется брандмауэр или установлен ключ аутентификации.)

Обеспечение точного отсчета времени

В большинстве компьютеров работа внутреннего таймера основана на использовании генератора — электронного устройства, вырабатывающего периодический сигнал. Например, сигнал с частотой 100 Гц изменяет свое состояние 100 раз в секунду. Считая изменения сигнала, вырабатываемого генератором, компьютер отсчитывает время. Как было сказано ранее в этой главе, компьютерные таймеры чаще всего работают неточно. Это происходит по разным причинам. Во-первых, частота сигнала, вырабатываемого генератором, может отличаться от ожидаемой. Если, например, вместо 100 Гц частота сигнала составляет 100,1 Гц, то системные часы будут спешить более чем на минуту в день. Во-вторых, частота сигнала может изменяться в зависимости от внешних факторов, например от температуры. В этом случае погрешность системных часов будет зависеть от температуры в комнате, от того, сколько времени компьютер проработал после включения, и т. д.

Погрешность в работе системных часов может также быть вызвана причинами, не связанными с работой генератора. Как правило, при поступлении очередного сигнала (этот сигнал носит название "тик") генерируется прерывание (на компьютерах x86 прерывание с номером 0). Если в это время процессор компьютера занят обработкой более приоритетных событий, некоторые из прерываний таймера могут остаться необработанными, т. е. некоторые "тики" могут быть пропущены.

Описанные причины приводят к возникновению "дрейфа" системных часов, что затрудняет работу пользователей и приводит к сбоям в выполнении некоторых важных программ. В ряде случаев, например при управлении научными экспериментами, отсчет времени должен производиться с высокой точностью. (Обычные версии Linux плохо справляются с такими задачами. Для управления научными экспериментами обычно используется разновидность данной системы, которая называется Real-Time Linux; дополнительную информацию о ней можно получить по адресу `http://fsmlabs.com/community/`.) Если вам необходимо организовать работу высокоточных часов, установите опцию ядра Enhanced Real Time Clock в меню Character Devices.

Наличие модулей ядра часто приводит к потере "тиков", поэтому, если необходимо отсчитывать время с высокой точностью, вам следует по возможности включить все необходимые драйверы в состав ядра и минимизировать число модулей. Если компьютер будет постоянно находиться во включенном состоянии в комнате с постоянной температурой, "дрейф" часов окажется почти постоянным и может быть учтен в процессе работы. Сервер NTP предпринимает попытки компенсации "дрейфа", сравнивая показания часов с информацией» полученной от вышестоящего сервера. Следует заметить, что в результате работы самого сервера NTP показания часов могут изменяться, поэтому данный сервер следует отключать на время выполнения тех операций» для которых необходим точный отсчет времени.

Использование клиентских средств NTP

Ранее уже шла речь о том, что при формировании сети можно расположить на одном из компьютеров сервер NTP, который получал бы информацию о времени от внешнего

сервера, и настроить остальные компьютеры так, чтобы они обращались за этой информацией к серверу NTP, расположенному в локальной сети. В результате во всей сети будет поддерживаться точное время, а трафик, связанный с NTP-обменом, будет относительно невелик. На каждом из компьютеров такой сети будет выполняться программа ntpd, но на всех машинах, кроме одной, она выступит в роли клиента. В сложной сети вы можете организовать несколько уровней серверов NTP. Например, для обслуживания клиентов в каждой подсети можно установить отдельный сервер NTP, а один из них настроить так, чтобы он обращался за сведениями о текущем времени в Internet. Такой подход минимизирует трафик NTP в локальной сети.

На подавляющем большинстве компьютеров используется лишь незначительная часть возможностей, предоставляемых ntpd. Эта программа синхронизирует показания системных часов на узлах локальной сети с точностью до миллисекунд и обеспечивает отклонение от UTC меньше секунды. Кроме того, программа ntpd работает постоянно и корректирует "дрейф" часов в течение дня. Но, как правило, потребности пользователей, работающих в сети, гораздо скромнее, для них вполне допустима погрешность в несколько секунд. Заметьте также, что ntpd представляет собой сервер и при работе этой программы на компьютере возникает определенная угроза безопасности системы. Если в программе будет обнаружена ошибка, создающая "лазейку" для злоумышленников (а подобные ошибки были выявлены в ранних реализациях сервера), ваш компьютер окажется открытым для всех пользователей локальной сети, а возможно, и для всей Internet. По этой причине ntpd в некоторых случаях целесообразно заменить программой ntpdate. В качестве клиента службы времени также может применяться программа rdate, но она использует отдельный протокол и уступает ntpdate в точности.

В настоящий момент разработчики NTP занимаются модернизацией программы ntpd. Они реализуют в ней возможность однократной коррекции времени так, как это происходит при использовании ntpdate. В последующих версиях NTP программа ntpdate не будет поставляться. Уже сейчас некоторые пакеты NTP 4 распространяются без ntpdate.

Для того чтобы запустить программу ntpdate, надо ввести ее имя и указать адрес сервера, который будет использован для синхронизации времени. Вы можете задать несколько серверов, в этом случае программа автоматически выберет наиболее подходящий из них. Между именем программы и адресом сервера могут присутствовать следующие опции.

- -B. По умолчанию ntpdate поступает следующим образом. Если погрешность системных часов превышает половину секунды, программа устанавливает новое значение времени; при меньшей погрешности она выполняет подстройку часов, т. е. замедляет или ускоряет их ход. Данная опция указывает на то, что подстройка должна применяться в любом случае, даже если значение ошибки очень велико.
- -b. Данная опция указывает на то, что даже при малом значении ошибки должно устанавливаться новое показание часов.
- -o *версия*. С помощью этой опции вы можете указать программе версию NTP для использования.
- -p число_показаний_времени. В обычных условиях ntpdate устанавливает системные часы на основании четырех показаний времени, полученных с сервера.

С помощью данной опции вы можете увеличить или уменьшить это значение (но оно должно оставаться в диапазоне от 1 до 8).

- -q. При указании опции `-q` программа опрашивает сервер, не изменяя значение системного времени. В этом случае сервер не возвращает данные в формате, удобном для восприятия; результаты опроса могут быть использованы для вычисления времени задержки при взаимодействии с сервером.

- -s. Данная опция применяется при запуске программы с помощью инструмента `cron`.

- -и. В обычных условиях `ntpdate` использует при передаче пакетов стандартный порт 123. С помощью этой опции вы можете указать на необходимость применения непривилегированного порта (с номером выше 1024). В некоторых случаях это приходится делать при работе через брандмауэр.

После запуска программа `ntpdate` выводит различные статические данные, в частности, уровень сервера, используемого для синхронизации, смещение и задержку. Если при выполнении программы не возникнет ошибка и если при ее вызове не была задана опция -q, `ntpdate` скорректирует показания системных часов, установив новое значение либо выполнив их подстройку.

Для периодического запуска программы `ntpdate` часто используется инструмент `cron`. В большинстве случаев достаточно запускать `ntpdate` один раз в сутки, но если необходима более высокая точность, ее можно вызывать чаще, например, один раз в день. Периодическое выполнение `ntpdate` уменьшит NTP-трафик по сравнению с использованием ntpd.

ВНИМАНИЕ Если вы используете для синхронизации времени общедоступный сервер, не планируйте вызов `ntpdate` на полуночь. Почему-то многие администраторы считают, что полночь — наилучшее время для коррекции системных часов, в результате в это время на сервер обрушивается лавина запросов. Указывайте для вызова `ntpdate` любое другое подходящее время, например 1:23 или 3:48. Это обеспечит более равномерную нагрузку на сервер, а ваша служба времени будет работать более точно и надежно, так как при обращении к серверу не возникнут непредвиденные задержки.

Использование Samba для предоставления данных о времени

Как вы уже имели возможность убедиться, NTP — чрезвычайно полезный протокол, позволяющий поддерживать с высокой точностью показания системных часов на компьютерах под управлением Linux. Кроме NTP, существуют и другие протоколы подобного назначения. Один из них реализован в составе протоколов SMB/CIFS, используемых для разделения файлов и принтеров. (Эти протоколы и реализующий их продукт Samba рассматривались в главе 7.) Если вы планируете запустить сервер NTP на компьютере, на котором установлен сервер Samba, примите во внимание тот факт, что гораздо проще сконфигурировать Samba для предоставления данных о времени, чем инсталлировать клиенты

NTP на каждом из Windows-компьютеров. (Samba не позволяет устанавливать системное время, обращаясь к серверу SMB/CIFS, работающему под управлением Windows.)

Опция временного сервера в конфигурационном файле Samba

Как вы уже знаете, конфигурационный файл `smb.conf`, используемый для настройки сервера Samba, состоит из нескольких разделов, большинство из которых описывает разделяемые каталоги. Однако первый раздел с именем `[global]` содержит установки по умолчанию, а также опции, которые не могут быть включены в описания разделяемых объектов. Одна из этих опций, `time server`, позволяет активизировать временной сервер. Для этого надо включить в файл `smb.conf` следующее выражение:

```
time server = Yes
```

Данное значение опции указывает на то, что сервер Samba должен отвечать на запросы клиентов SMB/CIFS и предоставлять им сведения о текущем времени. Вы можете задавать данное значение опции независимо от того, используется ли на вашем компьютере `ntpdate`, `rdate` или другая программа. Однако следует заметить, что для того, чтобы сервер Samba предоставлял точные данные о времени, следует принять меры для синхронизации системных часов этого компьютера.

Протокол SMB/CIFS не обеспечивает такой точности установки времени, как NTP. Сразу после выполнения процедуры синхронизации времени разница в показаниях системных часов различных Windows-клиентов может составлять около секунды.

НастройкаWindows-клиентадляавтоматической коррекции системного времени

Для того чтобы установить текущее время на клиентской машине под управлением Windows, надо выполнить следующую команду:

```
C:\> NET TIME\\SERVER/SET /YES
```

В данном случае *SERVER* — это NetBIOS-имя сервера Samba. Как и при использовании программы `ntpdate` в системе Linux, вы можете ввести данную команду вручную. Возможно, вы предпочтете, чтобы эта команда выполнялась при загрузке системы или при регистрации пользователя в ней. Для этого надо включить ее в файл . BAT (он может называться, например, `SETTIME.BAT`) и скопировать этот файл в папку `Startup`. Если ваша сеть состоит из доменов, вы можете включить данную команду в состав сценария регистрации, используемого по умолчанию. (Из `AUTOEXEC.BAT` ее вызывать нельзя, поскольку при выполнении этого файла сетевые средства еще не запущены.)

Windows 2000 и XP обеспечивают непосредственную поддержку NTP. Команда NET TIME `/SETSNTP:`*`NTP_сервер`* позволяет выполнять синхронизацию времени с использованием сервера *`NTP_сервер`*. В состав этих систем входит даже полнофункциональный сервер NTP, но обсуждение его конфигурации выходит за рамки данной книги.

Резюме

Временной сервер дает возможность синхронизировать показания системных часов компьютеров вашей сети друг с другом, а также с внешним сервером, который, в свою очередь, получает сведения от эталонного источника времени. Использование временного сервера позволяет устранить проблемы, возникающие из-за неодинаковой настройки системных часов разных узлов сети. Одним из самых популярных протоколов, предназначенных для обеспечения работы временных серверов, является NTP. Сервер, поддерживающий NTP (обычно он реализуется с помощью программы ntpd или xntpd), работает постоянно и периодически сверяет свои данные о времени со сведениями, полученными от других серверов NTP. В небольших сетях достаточно установить один сервер NTP уровня 3, синхронизировав его с сервером уровня 2 (выше такого сервера в иерархии NTP находятся сервер уровня 1 и источник эталонных данных о времени). Для синхронизации клиентских компьютеров с сервером уровня 3 используется `ntpd` либо клиентская программа `ntpdate`. В больших сетях целесообразно установить несколько серверов NTP и даже источник эталонных данных, например устройство GPS.

Для получения данных о времени и коррекции системных часов предназначены также клиент-программа `rdate` и временной сервер SMB/CIFS, реализованный в пакете Samba. Для этой же цели служит команда NET системы Windows. Команду NET можно использовать для коррекции системных часов клиентского компьютера под управлением Windows, не инсталлируя на нем программное обеспечение NTP.

Глава 11

Получение почты: протоколы POP и IMAP

Электронная почта — одна из наиболее популярных служб Internet. Большинство пользователей глобальной сети ежедневно прибегают к ее услугам. С ее помощью можно одинаково просто послать сообщение сотруднику, работающему за соседним столом, и адресату, находящемуся на другом континенте. В системе Linux реализована поддержка различных почтовых протоколов. В данной главе рассматривается один из классов почтовых протоколов, а именно *протоколы получения почты.* При использовании этих протоколов доставка писем осуществляется по инициативе получателя. Кроме протоколов получения существуют также *протоколы передачи почты,* которые осуществляют передачу писем по инициативе отправителя. (Наиболее популярный на сегодняшний день протокол передачи почты SMTP будет рассмотрен в главе 19.) Протокол передачи почты является непременным участником процесса доставки писем. Протоколы получения используются, как правило, на последней стадии, в редких случаях — на промежуточных этапах доставки сообщений.

Несмотря на то что протокол передачи почты будет рассматриваться лишь в главе 19, необходимо заметить, что, для того, чтобы вы могли пользоваться протоколом получения почты, в вашем распоряжении должен быть действующий сервер, реализующий протокол передачи. Готовый к работе сервер SMTP поставляется практически со всеми версиями Linux, поэтому в системе Linux с самого начала реализованы по крайней мере минимальные средства передачи писем. Если у вас возникнут проблемы с использованием сервера передачи или вам потребуется установить конфигурацию этого сервера, отличную от конфигурации, заданной по умолчанию, ознакомьтесь с материалом, изложенным в главе 19.

В начале данной главы будут обсуждаться протоколы получения почты и использование двух протоколов такого типа, наиболее популярных в настоящее время. Затем речь пойдет о настройке Linux для использования этих протоколов. В заключение будет рассмотрен продукт Fetchmail, который реализует протокол получения, но может также перенаправлять письма по другому адресу, используя протокол передачи.

Использование серверов доставки почты

Предположим, что вам необходимо обеспечить почтовые услуги для небольшого офиса или другой ограниченной группы пользователей. Средства передачи почты, поставляемые в составе каждого дистрибутивного пакета Linux, позволяют организовать на компьютере почтовый сервер, который принимал бы письма, адресованные вашим пользователям. Возникает вопрос: "Как организовать доступ пользователей к письмам, хранящимся на сервере?" Это можно сделать двумя способами.

- Вы можете предоставить пользователям возможность просматривать корреспонденцию непосредственно на почтовом сервере. Для этого могут быть использованы такие Linux-программы, как pine, mutt или KMail. Эти утилиты имеют непосредственный доступ к поступающей на сервер почте, и для их работы не требуется дополнительное программное обеспечение. Пользователи должны регистрироваться либо непосредственно на почтовом сервере, либо осуществлять удаленную регистрацию с помощью протоколов Telnet, SSH (Secure Shell — защищенная оболочка) или специальных средств X Window. (Вопросы удаленной регистрации будут подробно рассмотрены в главах 13 и 14.)

- Вы можете инсталлировать на компьютере, содержащем почтовый сервер, еще один сервер, реализующий протокол получения почты. В этом случае на пользовательских компьютерах выполняются программы просмотра и подготовки писем. Такие программы представляют собой клиенты сервера получения почты. С точки зрения почтового сервера сервер получения почты является локальным компонентом системы просмотра писем, а программа на компьютере пользователя — удаленным компонентом такой системы.

Первый подход широко применялся тогда, когда сервер UNIX был единственным компьютером в лаборатории или другом подразделении организации, а пользователи работали за удаленными терминалами. В настоящее время пользователи предпочитают для работы с почтой программы с графическим интерфейсом, выполняющиеся в среде Windows или MacOS. В системе Linux существуют программы просмотра почты с графическим интерфейсом, но для работы с ними с удаленных компьютеров необходимо использовать серверы X Window. Такие серверы крайне редко применяются в системах Windows и MacOS. Таким образом, для пользователей, работающих на компьютерах под управлением Windows или MacOS, второй подход предпочтительнее первого. Для его реализации на пользовательском компьютере должна быть установлена программа просмотра почты. В ней необходимо указать имя или IP-адрес узла, на котором инсталлирован сервер получения почты. При наличии такой программы, чтобы инициировать доставку почты на свой компьютер, пользователю достаточно щелкнуть мышью на кнопке в окне. Более того, доставка почты может осуществляться и без участия пользователя, так как многие средства просмотра автоматически проверяют наличие почты.

Установка сервера получения почты имеет смысл тогда, когда вам необходимо обеспечить доступ к почтовым сообщениям тех пользователей, которые хотят работать с программами просмотра почты на своих компьютерах, не регистрируясь на почтовом сервере. Серверы получения часто используются в сетях различных организаций и на компьютерах провайдеров. При наличии соединения с достаточно высокой пропускной способностью, быстродействующего процессора и жесткого диска большого объема такой сервер может обслуживать тысячи пользователей.

Принцип действия протоколов POP и IMAP

В предыдущем разделе были описаны лишь общие принципы доставки почты. Для того чтобы понять работу протокола получения почты, надо более подробно рассмотреть функционирование почтовой системы и вопросы взаимодействия протокола получения с другими компонентами доставки почты. В данной главе обсуждаются два основных протокола получения: POP (Post Office Protocol — протокол почтового отделения) и IMAP (Internet Message Access Protocol — протокол доступа к сообщениям Internet). Приведенные ниже рассуждения в равной степени относятся к обоим протоколам, несмотря на то, что они существенно различаются между собой. На практике достаточно использовать один протокол получения. Представляя себе круг задач, на которые ориентирован каждый протокол, вы сможете обоснованно выбрать для себя наиболее подходящий из них.

Функции протоколов получения почты

Как было сказано ранее, при организации почтового сервера необходимо установить сервер передачи почты. Без него использование сервера получения не будет иметь смысла, так как письма, предназначенные для доставки пользователям, попросту не будут поступать на почтовый сервер. В большинстве случаев письмо, подготовленное на компьютере пользователя, попадает на сервер передачи почты (не исключено, что на пути к получателю оно побывает на нескольких серверах передачи) и в конечном итоге попадает на сервер получения. В ответ на очередное обращение клиента сервер получения передает ему письмо.

В отличие от многих других протоколов, при доставке электронных писем некоторые серверы используются в качестве *ретрансляторов* (relay). Вместо того чтобы доставить письмо непосредственно на компьютер адресата, почтовая система пытается передать его на компьютер, расположенный как можно ближе к адресату. Строго говоря, почтовая система в принципе не может доставить письмо на компьютер пользователя, так как в почтовом адресе отсутствует информация об этом компьютере. Предположим, например, что вы передаете сообщение по адресу `sammy@threeroomco.com`. Анализируя записи сервера DNS, можно выяснить, что это письмо должно быть передано на компьютер `mail.threeroomco.com`. Не исключено, что соответствующие средства на этой машине сконфигурированы так, что письмо будет перенаправлено на другой компьютер, например `gingko.threeroomco.com`. Если пользователь применяет протокол получения почты, он может обращаться за своими письмами с удаленной машины, например `larch.threeroomco.com`. При подготовке письма вы пользуетесь специальным почтовым клиентом (предположим, что он выполняется на компьютере `trilobite.pangaea.edu`). Этот клиент сконфигурирован для работы с определенным сервером передачи почты (например, `franklin.pangaea.edu`). Сервер, в свою очередь, может быть сконфигурирован для передачи писем через ретранслятор (пусть ретранслятор имеет адрес `osgood.pangaea.edu`). В результате в процессе доставки сообщения от отправителя к получателю участвует достаточно длинная цепочка почтовых серверов. Условно путь, который проходит письмо, представлен на рис. 11.1. Большинство серверов передачи использует протокол SMTP (Simple Mail Transfer Protocol — простой протокол передачи почты). При условии, что сетевые средства и программы-серверы функционируют нормально, почтовое сообщение быстро преодолеет путь от `trilobite.pangaea.edu` к `gingko.threeroomco.com`. На компьютере `gingko.threeroomco.com`, который является предпоследним в цепочке, письмо мо-

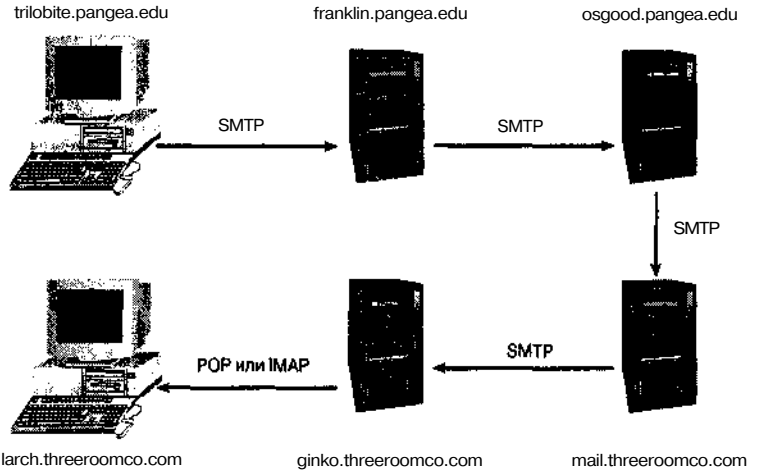


Рис. 11.1. В процессе доставки электронных писем может участвовать несколько серверов передачи, выполняющих роль ретрансляторов. Как правило, доставку сообщения на клиентскую машину осуществляет сервер получения

жет задержаться на неопределенно долгое время, так как сервер получения не сможет передать его до тех пор, пока клиент (в данном случае это программа на компьютере `larch.threeroomco.com`) не обратится к серверу. По этой причине компьютеры, на которых устанавливаются серверы получения, должны быть оснащены жесткими дисками большого объема, которые необходимы для хранения писем. Эти требования предъявляются как к серверам POP, так и к серверам ШАР.

На рис. 11.1 представлен лишь один из возможных путей доставки почты. В некоторых случаях в процессе передачи письма может участвовать лишь один компьютер (так происходит, если письмами обмениваются пользователи одной системы). Не исключено, что цепочка серверов будет более длинной, чем это показано на рис. 11.1. Как в домене отправителя, так и в домене получателя передачей сообщения могут заниматься дополнительные компьютеры. В процесс доставки письма могут быть также вовлечены компьютеры из других доменов. Например, если студент закончил учебное заведение и устроился на работу в некоторую организацию, письма, адресованные ему, будут приходить по адресу, который он использовал ранее, а затем, если система настроена корректно, они будут перенаправляться по новому адресу. Как вы узнаете, изучив материал данной главы, серверы получения могут применяться не только на последнем этапе доставки письма, но и на других стадиях.

Следует помнить, что сервер получения не используется для передачи почты. Передачей сообщений занимаются другие серверы, реализующие SMTP или другой протокол аналогичного назначения. Но на компьютере может присутствовать как сервер передачи, так и сервер получения. Поэтому пользователь может передавать и получать письма посредством одного и того же узла сети, но использовать при этом различные протоколы. В некоторых случаях сервер передачи и сервер получения располагаются на разных компьютерах. Например, на узле `franklin.pangaea.edu` может быть установлен сервер SMTP, используемый для передачи писем, а на компью-

тере ponyexpress.pangaea.edu — сервер получения (POP или ШАР). На узле ponyexpress.pangaea.edu также может присутствовать сервер SMTP, но использоваться лишь для получения почты от других серверов передачи.

Хранение писем на стороне клиента и на стороне сервера

Как было сказано ранее, одна из функций сервера, реализующего протокол получения почты, состоит в том, чтобы хранить сообщения до тех пор, пока пользователь не обратится за ними. Когда клиент-программа обращается к серверу, почтовые сообщения копируются на клиентскую машину. На первый взгляд может показаться, что при этом они должны удаляться с сервера, однако так происходит не всегда. Ниже описаны случаи, когда целесообразно оставлять копии сообщений на сервере.

- При получении почты с двух различных компьютеров. В этом случае многие предпочитают оставлять копии сообщений на сервере, независимо от того, применяют ли они протокол POP или ШАР. Следует заметить, что при использовании протокола POP накопившиеся на сервере сообщения вскоре начинают мешать работе, так как POP предоставляет очень несовершенные средства для идентификации отдельных сообщений и их обработки.

- При использовании протокола ШАР. В этом случае есть возможность организовывать "папки" сообщений на сервере. Поэтому, если вы работаете на различных компьютерах, вы можете разместить сообщения на сервере удобным вам способом и не испытывать ненужных трудностей, возникающих вследствие дублирования сообщений. IMAP также предоставляет возможность копировать на пользовательский компьютер только заголовки сообщений (содержащие, в частности, адрес отправителя и тему письма), поэтому для просмотра корреспонденции с помощью данного протокола не требуется высокая пропускная способность линии связи.

В некоторых случаях различия между POP и ШАР несущественны для пользователя, но иногда они играют решающую роль при выборе протокола. Если вы всегда работаете на одном и том же компьютере и хотите хранить почтовые сообщения на локальной машине, вам подойдет как POP, так и ШАР. Если же вам приходится работать на разных машинах или если, работая на одном компьютере, вы используете различные программы просмотра почты, протокол ШАР будет более удобен для вас. Следует заметить, что при хранении писем на сервере для их просмотра требуется больше времени, особенно если вам необходимо часто обращаться к старым письмам. Как правило, для организации сервера ШАР требуется больший объем диска и более высокая пропускная способность соединения, чем для сервера POP, поэтому, если вам не нужны специальные возможности ШАР, лучше использовать для получения почтовых сообщений протокол POP.

Пример сеанса взаимодействия по протоколу POP

На самом деле POP — это несколько взаимодействующих между собой протоколов. На сегодняшний день наиболее популярна версия POP-3, которая использует TCP-порт 110. (Более ранняя версия POP-2 использовала порт 109.) Подобно многим другим протоколам, применяемым в Internet, POP-взаимодействие основано на обмене текстовыми

сообщениями между клиентом и сервером. В POP-3 предусмотрено более десяти команд. Среди них можно отметить команды USER (указание имени пользователя), PASS (указание пароля), RETR (получение сообщения), DELE (удаление сообщения) и QUIT (завершение сеанса). Пример простого сеанса POP-взаимодействия, в ходе которого клиент получает с сервера одно письмо, представлен в листинге 11.1. В данном примере для обращения к серверу POP-3 и ввода необходимых команд вручную использовалась клиентская программа telnet. Программы просмотра почты скрывают от пользователя реальный ход обмена и предоставляют лишь конечные результаты.

Листинг 11.1. Пример сеанса взаимодействия по протоколу POP-3

```
$ telnet nessus 110
Trying 192.168.1.3. ..
Connected to nessus.rodsbooks.com.
Escape character is '^]'.
+OK POP3 nessus.rodsbooks.com v7.64 server ready
USER rodsmith
+OK User name accepted, password please
PASS password
+OK Mailbox open, 1 messages
RETR 1
+OK 531 octets
>From rodsmith Wed Aug 8 14:38:46 2001
Return-Path: <ben@pangaea.edu>
Delivered-To: rodsmith@nessus.rodsbooks.com
Received: from speaker.rodsbooks.com (speaker.rodsbooks.com
[192.168.1.1])
        by nessus.rodsbooks.com (Postfix) with SMTP id EB2A01A2BD
        for <rodsmith@nessus.rodsbooks.com>; Wed, 8 Aug 2001
14:38:26 -0400 (EDT)
Message-Id: <20010808183826.EB2A01A2BD@nessus.rodsbooks.com>
Date: Wed, 8 Aug 2001 14:38:26 -0400 (EDT)
From: ben@pangaea.edu
To: undisclosed-recipients:;
Status:

This is a test message.
.
DELE 1
+OK Message deleted
QUIT
+OK Sayonara
Connection closed by foreign host.
```

Как видно из листинга 11.1, при использовании протокола POP сообщения идентифицируются по номерам. В данном примере на сервере присутствует лишь одно сообщение;

на это указывает строка +OK mailbox open, 1 messages. Номер сообщения указывается при его передаче клиенту (команда RETR 1) и удалении (команда DELE 1). Протоколом POP не предусмотрена передача части сообщения: оно должно передаваться целиком либо не передаваться вовсе. Средства определения длины сообщения, адреса отправителя и получения другой информации в данном протоколе отсутствуют. Интересующие вас характеристики письма можно узнать лишь после того, как оно будет скопировано на клиентскую машину. В данном примере объем заголовка (в котором указывается адрес отправителя, дата и другие сведения) превышает объем тела сообщения. При передаче реальных писем тело сообщения, как правило, значительно больше его заголовка.

Анализируя заголовок письма в листинге 11.1, нетрудно заметить одну особенность, которая обеспечивает гибкость в работе почтовой системы, но в то же время затрудняет определение реального отправителя письма. В полях From: и Return-Path: указано, что отправителем письма является пользователь ben@pangaea.edu. Тем не менее эти поля заголовка нетрудно подделать. Кроме того, в заголовке каждого письма присутствует поле Received:, в котором указан сервер, использованный при получении письма, и адрес, с которого письмо попало на этот сервер. *Я* отправил это сообщение с одного компьютера, подключенного к моей сети, на другой компьютер; этот факт отражен в поле Received:. Как видно из листинга, письмо отправлено с speaker.rodsbooks.com и доставлено на nessus.rodsbooks.com. Компьютер pangaea.edu в передаче письма не участвовал.

Пример сеанса взаимодействия по протоколу IMAP

Как и POP, IMAP представляет собой протокол получения почты, однако IMAP позволяет использовать расширенные средства управления сообщениями. Применяя IMAP, пользователь, перед тем как копировать письма на свой компьютер, может ознакомиться с их заголовками. Наличие дополнительных возможностей предполагает реализацию дополнительных команд; таковых в IMAP-4 предусмотрено больше двадцати. (IMAP-4 является текущей версией данного протокола и использует при работе порт 143.) Пример сеанса взаимодействия по протоколу IMAP приведен в листинге 11.2. В ходе этого сеанса достигается такой же результат, как и при использовании протокола POP (листинг 11.1). Отличие лишь в том, что листинг 11.2 включает команду копирования сообщения в папку IMAP.

Листинг 11.2. Пример сеанса IMAP-4

```
$ telnet nessus 143
Trying 192.168.1.3...
Connected to nessus.rodsbooks.com.
Escape character is '^]'.
* OK nessus.rodsbooks.com IMAP4rev1 v12.264.phall server ready
A1 LOGIN rodsmith password
A1 OK LOGIN completed
A2 SELECT Inbox
* 1 EXISTS
```

```
* NO Trying to get mailbox lock from process 29559
* 1 RECENT
* OK [UIDVALIDITY 997295985] UID validity status
* OK [UIDNEXT 4] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)]
Permanent flags

* OK [UNSEEN 1] first unseen message in /var/spool/mail/rodsmith
A2 OK [READ-WRITE] SELECT completed
A3 FETCH 1 BODY[HEADER]
* 1 FETCH (BODY[HEADER] {494}
>From rodsmith Wed Aug 8 16:02:47 2001
Return-Path: <ben@pangaea.edu>
Delivered-To: rodsmith@nessus.rodsbooks.com
Received: from speaker.rodsbooks.com (speaker.rodsbooks.com
[192.168.1.1])
        by nessus.rodsbooks.com (Postfix) with SMTP id 2C7121A2BD
        for <rodsmith@nessus.rodsbooks.com>; Wed, 8 Aug 2001
16:02:25 -0400 (EDT)
Message-Id: <20010808200225.2C7121A2BD@nessus.rodsbooks.com>
Date: Wed, 8 Aug 2001 16:02:25 -0400 (EDT)
From: ben@pangaea.edu
To: undisclosed-recipients:;

)
* 1 FETCH (FLAGS (\Recent \Seen))
A3 OK FETCH completed
A4 FETCH 1 BODY[TEXT]
* 1 FETCH (BODY[TEXT] {25}
This is a test message.
)
A4 OK FETCH completed
A5 COPY 1 demos
A5 OK COPY completed
A6 LOGOUT
* BYE nessus.rodsbooks.com IMAP4revl server terminating connection
A6 OK LOGOUT completed
Connection closed by foreign host.
```

Листинг 11.2 демонстрирует дополнительные возможности ШАР, которые отсутствуют в протоколе POP. IMAP требует от клиента передавать ему нумерованные команды, например, вместо LOGOUT в листинге указано **A6** LOGOUT. Эта особенность скрыта от пользователя, так как обработка команд полностью производится клиентской программой. IMAP позволяет копировать заголовки отдельно от текста сообщений (команды A3 и A4 в приведенном листинге). Использование папок предполагает выбор нужной папки в ходе

сеанса взаимодействия (команда A2), но пользователь получает возможность копировать письма из одной папки в другую (команда A5). В листинге 11.2 представлена лишь часть возможностей ШАР. Существует много разновидностей приведенных команд, в частности, различные способы обработки писем обеспечиваются с помощью команды FETCH. Дополнительные сведения о протоколе IMAP можно получить в специальных документах, один из которых находится по адресу `http://www.ietf.org/rfc/rfc2060.txt`.

Несмотря на то что рассмотрение низкоуровневых команд позволяет получить представление о работе IMAP, вам, как системному администратору, вряд ли необходимо знать детали функционирования этого протокола. Однако наличие некоторых команд оказывает влияние на конфигурацию сервера. Поскольку IMAP позволяет работать с папками, эти папки надо где-то хранить. Расположение папок зависит от используемого сервера. В настоящее время наиболее популярен сервер IMAP, разработанный в Вашингтонском университете (UW IMAP; `http://www.washington.edu/imap/`). Этот сервер хранит все папки в рабочем каталоге пользователя. Исключение составляет папка INBOX, которая находится в одном из стандартных каталогов, используемых почтовой системой, а именно, в `/var/spool/mail/`*`имя_пользователя`*. Когда пользователь впервые обращается к серверу IMAP, для него существует только папка INBOX. В процессе работы пользователь может создавать новые папки, применяя для этого соответствующие команды программы просмотра почты. Получив подобную команду, сервер UW IMAP создает каталог в рабочем каталоге пользователя. Прочие серверы используют для организации папок другие каталоги. Необходимые сведения по этому вопросу вы можете получить из документации на конкретный сервер. Выполняя администрирование системы, необходимо знать, где размещаются папки, чтобы выделить необходимое для них дисковое пространство. Это особенно важно на крупных серверах, обслуживающих большое количество пользователей, либо в тех случаях, когда пользователи хранят на сервере почтовые сообщения большого объема.

Выбор протокола

Выбор протокола получения почты зависит от имеющихся в наличии ресурсов и от того, какие клиентские программы применяют пользователи. Не секрет, что на решение вопроса большое влияние оказывают вкусы и привычки самого системного администратора. При использовании протокола POP требования к ресурсам сервера и пропускной способности линии минимальны, поскольку подавляющее большинство пользователей не хранят письма на сервере, а копируют их на клиентские машины. Некоторые пользователи предпочитают работать с сервером IMAP. Чтобы организовать функционирование такого сервера, необходимы дополнительные ресурсы, в частности, для хранения писем на сервере потребуется жесткий диск большего объема. В составе почтовых систем серверы POP используются чаще, чем IMAP, поэтому, если вы установите сервер IMAP, желательно установить на том же компьютере и сервер POP. При этом дополнительная нагрузка на компьютер практически не будет ощущаться, но пользователи, которые работают с клиентскими программами, не поддерживающими IMAP, смогут получать свои письма. POP-клиент может очистить папку INBOX, но он не разрушает папки, созданные сервером IMAP. С другой стороны, использование дополнительного почтового сервера нежелательно с точки зрения безопасности системы. Если в защите сервера POP будут обнаружены недостатки, позволяющие проникать в систему, наличие такого сервера создаст дополнительную возможность незаконного доступа к данным.

ВНИМАНИЕ По умолчанию серверы POP и IMAP передают всю информацию, включая пароль, в незашифрованном виде. Поэтому пароли, используемые для получения почты, не следует применять в других целях, а в особенности для регистрации в системе. Существуют также защищенные варианты серверов POP и ШАР, в которых для передачи данных используется SSL-соединение. Несмотря на то что такие серверы сложнее настроить, это необходимо сделать, если в письмах могут содержаться секретные данные. Принять меры защиты особенно важно, когда обращение к серверам POP и IMAP может осуществляться через Internet. Если вы хотите ограничить доступ к серверам получения почты только компьютерами локальной сети, вы можете применить TCP Wrappers либо задать соответствующие правила `xinetd`. Кроме того, запретить обращение к серверам извне можно, настроив соответствующим образом брандмауэр.

Обеспечение работы по протоколу **POP**

Как правило, для обеспечения работы сервера POP не приходится затрачивать больших усилий. Чаще всего, настройка не требуется, и вам достаточно лишь запустить сервер. Тем не менее следует убедиться, что выбранная вами программа сервера совместима с сервером SMTP. Не менее важно обеспечить совместимость с форматом хранения писем. В настоящее время используются два таких формата: mbox (почтовый ящик) и maildir (почтовый каталог). Большинство серверов, например `sendmail`, Postfix, and Exim, по умолчанию используют формат mbox, но в некоторых серверах, например в `qmail`, изначально включен режим работы с maildir. Серверы Postfix, Exim и `qmail` можно настроить на использование как mbox, так и maildir. Выбранный вами сервер POP должен предоставлять возможность чтения писем в том формате, в котором их записывает сервер SMTP.

Серверы POP для Linux

Серверы POP входят в состав практически каждого дистрибутивного пакета Linux. Как правило, самым простым решением является использование сервера, поставляемого вместе с системой, но если вы перенастроите сервер SMTP для хранения писем в другом формате, вам, возможно, придется заменить сервер POP программой, в которой предусмотрена поддержка этого формата. Ниже перечислены некоторые программные продукты, обеспечивающие работу сервера POP.

- **UW** IMAP. Сервер IMAP, созданный в Вашингтонском университете (`http://www.washington.edu/imap/`), может также работать в режиме сервера POP. Этот сервер поставляется со многими версиями Linux и поддерживает формат mbox, используемый по умолчанию многими серверами SMTP.

- Cyrus IMAP. Подобно UW IMAP, Cyras IMAP (`http://asg.web.cmu.edu/cyrus/imapd/`) наряду с IMAP поддерживает протокол POP. Данный сервер использует для хранения поступающей корреспонденции формат mbox.

- nupop. Сервер nupop(`http://nupop.sourceforge.net`) предназначен для работы на крупных узлах, его имеет смысл устанавливать в том случае, если ваша почтовая система должна обслуживать большое количество пользователей. Данный

продукт ориентирован на использование формата maildir, поэтому лучше всего он взаимодействует с сервером qmail.

- **Courier.** Продукт Courier (http://www.courier-mta.org) реализует функции серверов POP, IMAP и SMTP. Серверы Courier POP и IMAP доступны в виде отдельного пакета Courier-IMAP (http://www.inter7.com/courierimap/). Эти серверы используют формат maildir.

- **QPopper.** Несмотря на свое название, данный пакет (http://www.eudora.com/qpopper/) не имеет никакого отношения к SMTP-серверу qmail. QPopper 3.0 распространялся на коммерческой основе. Версия 4.0 данного продукта доступна бесплатно в исходных кодах. QPopper использует формат mbox. QPopper 4.0 поддерживает работу через SSL-соединение.

- **qmail-pop3d.** Данная программа поставляется с сервером qmail (http://www.qmail.org) и использует формат maildir. Если вы решили использовать в качестве SMTP-сервера qmail, имеет смысл установить qmail-pop3d для поддержки протокола POP.

Здесь приведена информация лишь о незначительной части продуктов, реализующих обмен по протоколу POP. На сервере http://www.sourceforge.net можно найти большое количество серверов POP, многие из которых входят в состав пакетов, поддерживающих IMAP, SMTP и другие протоколы. В комплекте со многими версиями поставляется UW IMAP, а некоторые дистрибутивные пакеты включают также Cyrus, QPopper и другие продукты.

Инсталляция и настройка сервера POP

Как правило, серверы POP запускаются с помощью суперсервера (вопросы использования суперсерверов рассматривались в главе 4). При инсталляции некоторых серверов автоматически устанавливается конфигурация xinetd. Не исключено, что для запуска сервера вам придется вручную отредактировать файл /etc/inetd.conf. Для того чтобы сервер POP начал обрабатывать поступающие запросы, необходимо перезапустить inetd или xinetd. Если используемый вами сервер POP не входит в состав дистрибутивного пакета, вам следует прочитать документацию на данный продукт. Это позволит избежать возникновения проблем.

По умолчанию UW IMAP и большинство других серверов POP полагаются на результаты выполнения стандартной процедуры аутентификации Linux. Поэтому, чтобы сервер обслуживал пользователей вашей системы, не надо принимать никаких специальных мер. Если для пользователя создана учетная запись и если сервер SMTP получает его корреспонденцию, сервер POP будет доставлять письма на удаленный компьютер. Пользователю лишь необходимо указать в клиент-программе POP свое имя и пароль. Поскольку действия, выполняемые сервером POP, чрезвычайно просты, в большинстве программ специальный конфигурационный файл не используется.

Обеспечение работы по протоколу IMAP

Серверы IMAP инсталлируются и настраиваются практически так же, как и серверы POP. Как было сказано ранее, продукт UW IMAP поставляется с большинством дистри-

бутивных пакетов Linux, кроме того, в состав некоторых систем включаются и другие серверы. Действия по настройке обычно сводятся к изменению конфигурации и перезапуску суперсервера.

Серверы IMAP для Linux

Многие из пакетов, сведения о которых были приведены в предыдущем разделе, в частности UW IMAP, Cyrus IMAP и Courier, обеспечивают также работу сервера IMAP. В 2002 г. в стадии разработки находился ряд проектов по созданию серверов IMAP, но реально работающий код еще не был доступен. Информацию об этих проектах можно найти на сервере `http://www.sourceforge.net`. Некоторые из них направлены на решение совершенно экзотических задач, например, просмотр содержимого Web средствами IMAP.

Наиболее популярный сервер для Linux, UW IMAP, использует для организации большинства почтовых папок рабочие каталоги пользователей. Если пользователь время от времени регистрируется и работает на этом компьютере, такое решение нежелательно, так как пользователь может непреднамеренно удалить или переместить каталоги с папками. (Местоположение папок можно изменить; для этого надо модифицировать исходный код программы и перекомпилировать ее. Соответствующие действия описаны в файле CONFIG, входящем в состав документации на данный продукт.) Cyras IMAP хранит все папки в собственном формате. Исключением является лишь папка, которая содержит входящие сообщения; она представлена в формате mbox.

Инсталляция и настройка сервера IMAP

В состав большинства версий Linux входит продукт UW IMAP, соответствующий пакет обычно носит имя imap. При инсталляции этого пакета сервер IMAP будет настроен для запуска посредством суперсервера. При работе UW IMAP полагается на результаты стандартной процедуры аутентификации Linux, поэтому каждый пользователь, имеющий учетную запись на компьютере, может получать почту средствами IMAP. Для большинства серверов IMAP специальные конфигурационные файлы не предусмотрены, так как действия, выполняемые этими серверами, чрезвычайно просты.

Использование Fetchmail

Fetchmail — не совсем обычная программа. Она не является ни программой просмотра почты, ни почтовым сервером, но в то же время сочетает элементы их обоих. Fetchmail извлекает письма с сервера получения и передает их другой программе; чаще всего Fetchmail взаимодействует с сервером передачи почты и перенаправляет полученные письма локальным пользователям. Программу Fetchmail можно сконфигурировать для выполнения различных задач, более того, редактируя конфигурационный файл `.fetchmailrc`, ее можно настроить с учетом интересов конкретных пользователей. Существует инструмент с графическим интерфейсом `fetchmailconf`, специально предназначенный для настройки Fetchmail. Он предоставляет возможность сконфигурировать данную программу для выполнения различных задач.

Предыдущие разделы были посвящены организации работы сервера получения почты. Программа Fetchmail взаимодействует с уже работающим сервером получения.

Участие Fetchmail в процессе доставки почты

Протоколы получения почты были разработаны для того, чтобы обеспечить клиентским почтовым программам возможность извлекать сообщения с почтового сервера. Считается, что компьютер, на который приходят почтовые сообщения, работает постоянно и на нем выполняется сервер передачи почты. В отличие от сервера, время работы компьютера, на котором установлена клиентская программа, не определено, кроме того, его IP-адрес может изменяться от одного сеанса работы к другому. Возможны ситуации, при которых возникает необходимость извлечь почту с помощью протокола получения, а затем возобновить ее передачу посредством другого сервера. Подобные ситуации описаны ниже.

- **Компьютер под управлением Linux, подключенный к Internet по коммутируемой линии.** В большинстве случаев на компьютере Linux, даже если он подключен к Internet через PPP-соединение, присутствует почтовый сервер. Этот локальный сервер позволяет организовать обмен письмами между несколькими локальными пользователями или передавать пользователям сообщения, сгенерированные системой. Для того чтобы интегрировать эти сообщения с письмами, которые приходят на сервер провайдера, надо извлечь письма с помощью протокола POP или ШАР и включить их в очередь локального почтового сервера. В результате пользователь получает возможность читать все сообщения (как локальные, так и удаленные) с помощью одной программы; при этом он избавлен от необходимости обращаться к серверу посредством протокола POP или ШАР.

- **Локальная сеть, подключенная к Internet по коммутируемой линии.** В небольших компаниях компьютеры часто объединяют в локальную сеть, которая, помимо выполнения других задач, позволяет организовать обмен почтовыми сообщениями между пользователями. Подобная сеть может быть подключена к Internet по коммутируемой линии; при этом письма, адресованные пользователям локальной сети, доставляются на сервер провайдера. Администратору локальной сети приходится решать задачу получения писем и перенаправления их на пользовательские компьютеры в пределах локальной сети.

- **Получение почты, приходящей по нескольким адресам.** Если у вас есть несколько учетных записей, на которые приходят письма, вы можете автоматизировать процесс получения корреспонденции. Работая на компьютере под управлением Linux, подключенном к Internet по коммутируемой линии, вы можете использовать протокол получения почты для того, чтобы собрать письма из нескольких источников и перенаправить их на одну учетную запись. (Эта учетная запись может располагаться как на локальном, так и на удаленном компьютере.) Альтернативным решением данной проблемы является организация перенаправления почты на тех серверах, на которые она поступает, либо использование почтовой программы, опрашивающей различные учетные записи.

- Использование одного почтового ящика несколькими абонентами. Бывают случаи, когда на одну учетную запись приходят письма, адресованные нескольким пользователям. Если существуют правила, позволяющие рассортировать письма, вы можете сделать это, используя средства фильтрации Fetchmail (либо реализовать фильтрацию посредством отдельной программы) и разместить сообщения в отдельных очередях.

- Преобразование POP в IMAP. Предположим, что почта, адресованная пользователям вашей сети, приходит на внешний сервер, на котором установлен сервер POP, но пользователи предпочитают работать посредством протокола IMAP. С помощью компьютера под управлением Linux вы можете получать почту с сервера POP и помещать ее в локальную очередь. Установив IMAP на локальном компьютере, вы обеспечите для пользователей возможность работать с корреспонденцией, применяя средства, обеспечиваемые протоколом IMAP.

Планирование получения почты

Если адресованные вам письма приходят на PPP- или IMAP-сервер, вы должны принимать специальные меры для их получения. Вы не можете ожидать, пока программа оповестит вас о приходе очередного письма, как это происходит при работе на том компьютере, на котором присутствует сервер передачи почты. Чтобы письма оказались на вашей машине, вам надо специально вызвать клиентскую программу (вручную или с помощью инструмента, обеспечивающего периодическое выполнение некоторых действий, например `cron`). Если вы планируете периодические вызовы программы Fetchmail с помощью инструмента `cron`, вам надо решить, как часто будет опрашиваться почтовый сервер. Если приходящие письма должны быть прочитаны как можно скорее, вам следует установить как можно меньший интервал опроса, например, обращаться к серверу каждые пять минут. Однако такой подход имеет существенный недостаток. Частые обращения к серверу создают дополнительную нагрузку на сеть и занимают ресурсы как локального, так и удаленного компьютера. При этом также увеличивается вероятность того, что пароль будет перехвачен и использован в незаконных целях. Очень большой интервал (скажем, шесть часов) приведет к снижению нагрузки на сеть и компьютеры и снизит вероятность перехвата пароля, но при этом пришедшее письмо будет длительное время оставаться непрочитанным. Инструмент `cron` позволяет задавать интервалы между вызовами программы, различающиеся в зависимости от времени суток. Например, вы можете указать, что в течение рабочего дня сервер должен опрашиваться каждые полчаса, а в ночное время — не опрашиваться вовсе.

Если ваш компьютер подключается к Internet по коммутируемой линии, вам нет смысла постоянно вызывать Fetchmail. Лучше предусмотреть вызов этой программы в сценарии запуска сетевых средств. Например, вы можете включить вызов Fetchmail в сценарий `ppp-on-dialer`, который рассматривался в главе 2. Если программа Fetchmail настроена для работы в режиме демона, желательно в этом же сценарии принудительно завершить ее работу. В качестве альтернативного решения можно применить опции `interface` и `monitor`, которые будут рассматриваться в следующем разделе. В результате их использования Fetchmail будет предпринимать попытку получения писем только при наличии сетевого соединения.

Сказанное выше можно выразить следующим образом. Средства доставки почты с помощью серверов получения в основном предусматривают работу одного пользователя с применением достаточно простых средств. В особенности это относится к протоколу POP, который лучше всего подходит для тех случаев, когда пользователь работает с одной клиентской программой и получает письма с одного сервера. В более сложных ситуациях, например, когда один пользователь должен получать письма с нескольких серверов, либо когда письма, приходящие на одну учетную запись, должны быть распределены по нескольким потокам, нужны дополнительные инструменты. Одним из таких инструментов является Fetchmail.

Настраивая программу Fetchmail, важно обеспечить ее работу в качестве клиента как сервера получения, так и сервера передачи. (Fetchmail может не использовать сервер передачи, но такая конфигурация применяется крайне редко). Поскольку в составе Fetchmail объединены два клиента, эта программа требует, чтобы оба сервера были доступны. Fetchmail может работать как в пакетном режиме, так и в режиме демона. В пакетном режиме данная программа опрашивает один или несколько серверов получения и перенаправляет полученные письма с помощью сервера передачи. В режиме демона Fetchmail работает постоянно, опрашивая сервер получения через заданные интервалы. Этот режим нежелателен, поскольку, как показывает опыт, при длительной работе в режиме демона в работе Fetchmail происходят сбои. Поэтому предпочтительнее запускать этот инструмент в пакетном режиме. Исключение составляют случаи, когда работа в режиме демона продолжается недолго. Если вы хотите организовать периодический опрос сервера, можно планировать вызовы Fetchmail с помощью `cron`. Данную программу также можно запускать вручную.

Использование fetchmailconf

Настройка Fetchmail предполагает редактирование текстового конфигурационного файла. Этот файл обычно хранится в рабочем каталоге пользователя, от имени которого запускается программа. При необходимости Fetchmail могут вызывать несколько пользователей, работающих на одном компьютере. Для настройки Fetchmail можно использовать специальный инструмент с графическим интерфейсом fetchmailconf, который рассматривается в данном разделе. В следующем разделе будет описан конфигурационный файл `.fetchmailrc`, который модифицирует программа `fetchmailconf`.

В состав большинства версий Linux программа `fetchmailconf` входит как отдельный пакет. Поэтому, если вы хотите обеспечить в такой системе выполнение программы Fetchmail и ее настройку с помощью специального инструмента, вам придется установить два пакета. Как программа X Window, построенная на базе Tcl/Tk, `fetchmailconf` требует наличия дополнительных библиотек. После инсталляции Fetchmail и сопутствующих программ необходимо установить требуемую конфигурацию. Для этого выполните следующие действия.

1. Зарегистрировавшись как обычный пользователь, введите в окне xterm команду `fetchmailconf`. В результате вы увидите окно Fetchmail Launcher, в котором содержатся кнопки, предназначенные для настройки, тестирования, запуска Fetchmail и завершения работы.

Конфигурировать и запускать Fetchmail может также пользователь root, но ему не предоставляются никакие преимущества по сравнению с другими пользователями. Поэтому, для того, чтобы уменьшить риск, который неизбежен при регистрации под именем root, лучше зарегистрироваться для работы с Fetchmail как обычный пользователь. (Если в защите Fetchmail обнаружатся недостатки, то, работая с данной программой как root, вы создаете угрозу для безопасности всей системы в целом.) Зная необходимые пароли, любой пользователь может получать почту, предназначенную для других пользователей. В некоторых ситуациях подобная конфигурация может быть оправданной, но в большинстве случаев желательно настраивать Fetchmail так, чтобы пользователь имел доступ только к своим письмам.

2. Щелкните на кнопке Configure Fetchmail в окне Fetchmail Launcher. На экране отобразится окно Fetchmail Configurator, в котором вы сможете указать, предпочитаете ли вы использовать средства, ориентированные на начинающего или на опытного администратора. Конфигурация, предназначенная для начинающего, позволяет использовать лишь подмножество тех опций, которые представляются опытному администратору. При изложении материала данного раздела я в основном буду ориентироваться на расширенный набор опций, поступающих в распоряжение более квалифицированного администратора.

3. Щелкните на кнопке Expert Configuration в окне Fetchmail Configurator. Программа выведет диалоговое окно Fetchmail Expert Configurator, показанное на рис. 11.2. Если вы собираетесь запускать Fetchmail в режиме демона, введите в поле Poll Interval интервал между последовательными обращениями к серверу, выраженный в секундах (например, 1200 секунд соответствуют 20 минутам). Если вы хотите, чтобы программа Fetchmail выполнялась в пакетном режиме, оставьте в этом поле значение по умолчанию, равное 0. В поле Postmaster указывается имя пользователя, которому следует сообщать о проблемах, возникших в процессе работы Fetchmail. По умолчанию принимается имя пользователя, который запустил данную программу. Большинство опций, расположенных в средней части окна, можно оставить без изменений. Щелкнув на кнопке Help, вы получите информацию об их назначении.

4. Наиболее важный компонент окна Fetchmail Expert Configurator — панель, расположенная в нижней его части. С помощью этой панели вы можете задать имя почтового сервера, с которого собираетесь получать почту. Введите имя узла, и после нажатия клавиши <Enter> отобразится новое диалоговое окно Fetchmail Host *Имя_узла* (рис. 11.3). Заданное вами имя должно также появиться в списке, расположенном в окне Fetchmail Expert Configurator ниже поля New Server. Если вы собираетесь получать почту с нескольких серверов, вам надо ввести их имена, но задать имя следующего сервера можно лишь после установки всех конфигурационных параметров для предыдущего сервера.

5. Наиболее важными в окне Fetchmail Host *Имя_узла* являются разделы Protocol, User Entries for *Имя_узла* и Security. В разделе Run Controls задаются опции, определяющие временные соотношения при получении почты, и имя сервера (если оно отличается от введенного ранее). В области Multidrop Options указываются правила, используемые при дублировании или перенаправлении сообщения. Конкрет-

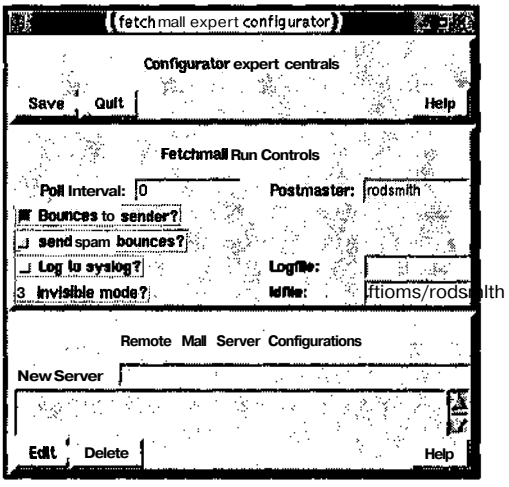


Рис. 11.2. Диалоговое окно Fetchmail Expert Configurator позволяет задавать глобальные опции и указывать почтовый сервер, к которому программа Fetchmail должна обращаться для получения почты

ное решение принимается на основании анализа указанных в этом разделе полей заголовка. Данный раздел используется в том случае, если надо обеспечить распределение писем, приходящих на одну учетную запись, однако в некоторых случаях, например при участии в списке рассылки, такое распределение может привести к возникновению проблем.

6. В разделе Protocol окна Fetchmail Host *Имя_узла* надо указать протокол получения почты. По умолчанию предполагается значение Auto; такая установка обеспечивает работу с некоторыми серверами, но если вы знаете, какой протокол поддерживается на сервере, желательно указать его явно. Для проверки протокола можно воспользоваться кнопкой Probe for Supported Protocols, но средства, активизируемые с ее помощью, не всегда работают корректно. Лучше запросить необходимую информацию у провайдера или провести сеанс вручную, используя клиентскую программу `telnet`. Примеры такого использования `telnet` были приведены в листингах 11.1 и 11.2.

7. Средства, доступные посредством раздела Security, особенно важны в тех случаях, когда взаимодействие с сервером получения производится по коммутируемой линии, причем эта линия не всегда активна. В поле Interface to Monitor введите имя интерфейса, например `ppp0`. Это заставит Fetchmail опрашивать сервер только в том случае, если с момента прошлого опроса интерфейс был использован другой программой. Информация, задаваемая в поле IP Range to Check Before Poll, используется для того, чтобы программа могла проверить, связан ли IP-адрес с указанным интерфейсом. Введите в данном поле IP-адрес и маску подсети, разделив их символом /; в результате Fetchmail будет опрашивать сервер лишь тогда, когда с устройством связан адрес, принадлежащий заданному диапазону. Например,

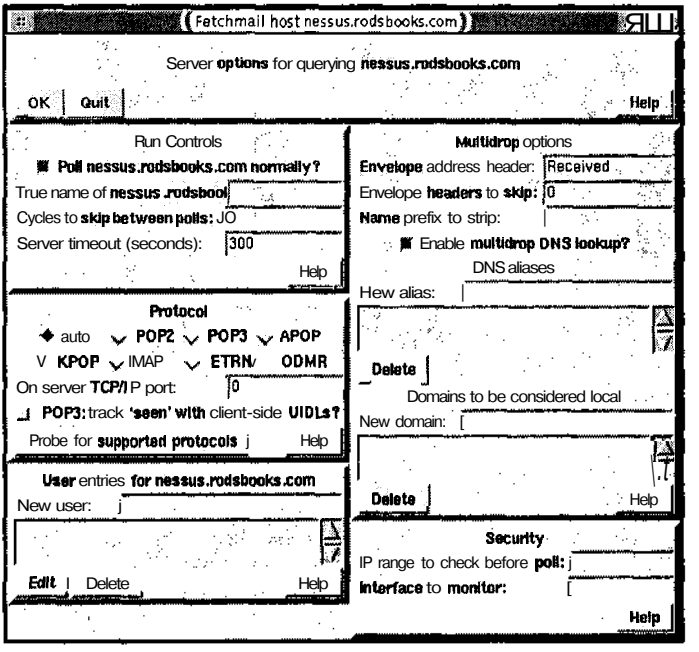


Рис. 11.3. Информация о сервере получения почты задается в диалоговом окне Fetchmail Host Имя_узла

если для интерфейса pppO указан адрес 172.20.0.0/255.255.0.0, то Fetchmail будет опрашивать сервер, только если интерфейсу `ppp0` будет соответствовать один из адресов сети 172.20.0.0/16.

8. В разделе User Entries for *Имя_узла* есть поле New User. Введите в нем имя, под которым вы зарегистрированы на почтовом сервере. После нажатия клавиши <Enter> отобразится окно Fetchmail User *Имя_пользователя* Querying *Имя_узла* (рис. 11.4). Как и при указании имени сервера, следующую учетную запись вы можете указать только после завершения работы с этим окном.

9. Наиболее важным элементом в окне Fetchmail User *Имя_пользователя* Querying *Имя_узла* является поле Password в разделе Authentication. Информация, задаваемая в этом поле, необходима для получения почты с сервера. Следует убедиться, что в поле Local Names перечислены все локальные пользователи, которые должны получать письма посредством данной учетной записи. По умолчанию предполагается, что локальное имя совпадает с именем на сервере; при необходимости вы можете задать другое имя. Раздел Forwarding Options позволяет указать узел для передачи почты. По умолчанию в качестве такого узла используется локальная система, но с помощью Fetchmail можно также организовать получение писем с одного узла и передачу их на другой узел. Опции в разделах Forwarding Options, Processing Options и Resource Limits используются редко, и назначение большинства опций понятно из их названий. При первом запуске Fetchmail целесообразно установить флажок опции Suppress Deletion of Messages After Reading в разделе Process-

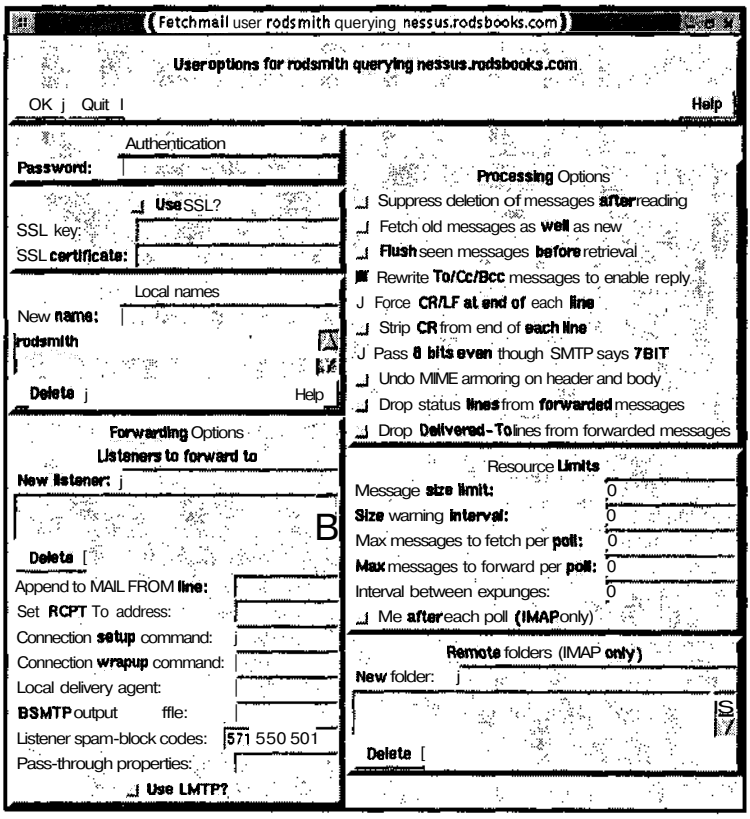


Рис. 11.4. Многие опции Fetchmail определяют работу с конкретными учетными записями на почтовом сервере

ing Options, чтобы исключить риск потери писем вследствие некорректной работы Fetchmail. После того как вы убедитесь, что программа работает корректно, опцию можно отключить. В разделе Remote Folders указываются папки IMAP, которые программа Fetchmail проверяет, помимо папки INBOX.

ВНИМАНИЕ Пароли, используемые для получения почты, Fetchmail хранит в незакодированном виде в конфигурационном файле `.fetchmailrc`. Программа Fetchmail не будет запускаться, если права доступа заданы менее строго, чем 0600 (`rw-------`), тем не менее, некоторая угроза безопасности системы все же сохраняется. Поэтому пароли, используемые для получения почты, не должны применяться в других целях, даже для регистрации на том же компьютере, с помощью той же учетной записи.

10. Щелкните на кнопке OK в окне Fetchmail User *Имя_пользователя*Querying *Имя_узла,* а затем на такой же кнопке в окне Fetchmail Host *Имя_узла*.Для того чтобы установки сохранились в файле `.fetchmailrc`, надо щелкнуть на кнопке Save в окне Fetchmail Expert Configurator.

11. Для того чтобы проверить конфигурацию программы, щелкните на кнопке Test Fetchmail в окне Fetchmail Launcher. В результате программа Fetchmail будет запущена в режиме отладки, т. е. вы увидите команды, которые данная программа передает серверу получения и серверу передачи почты, а также ответы этих серверов. Эта информация позволяет выявить и устранить проблемы, возникающие при работе Fetchmail. Убедившись в работоспособности программы, завершите ее работу щелчком на кнопке Quit.

ВНИМАНИЕ Несмотря на то что при отладке отображается дополнительная информация, в этом режиме письма не сохраняются на почтовом сервере. Поэтому, если при работе Fetchmail возникнут проблемы, письма будут утеряны (так, например, может случиться, если локальный почтовый сервер не работает или отвергает переданные ему сообщения). Поэтому на время тестирования желательно запретить удаление писем с сервера (соответствующая опция рассматривалась выше в п. 9).

Во многих случаях средства настройки, предоставляемые **fetchmailconf**, достаточны для решения задач, связанных с обработкой почты. В частности, эта программа позволяет организовать получение почты с одной или нескольких учетных записей и доставить ее локальным пользователям. Некоторые администраторы предпочитают вручную редактировать конфигурационный файл. Это позволяет быстро внести необходимые изменения или реализовать сложные конфигурации Fetchmail. Независимо от того, какой способ вы изберете для настройки данной программы, знание формата **.fetchmailrc** будет полезно для вас.

Редактирование .fetchmailrc

Если вы работаете с **fetchmailconf**, данная программа преобразует сделанные вами установки в записи, помещаемые в файл .fetchmailrc. Этот файл по умолчанию располагается в вашем рабочем каталоге. Очевидно, что структура файла отражает набор опций, предоставляемых программой **fetchmailconf**. Пример содержимого файла .fetchmailrc приведен в листинге 11.3.

Листинг 11.3. Пример файла .fetchmailrc

```
# Fetchmail file for retrieving mail from mail.abigisp.net
# and imap.asmallisp.com
set postmaster rodsmith
set bouncemail
set daemon 1800
set syslog
poll mail.abigisp.net with proto POP3
    user rodericksmith there with password abc123
    is rodsmith here fetchall forcecr
    smtphost speaker.rodsbooks.com
poll imap.asmallisp.com with proto IMAP
    user rodsmith there with password A1B2C3
    is rodsmith here
```

Как и в других конфигурационных файлах, символ # является признаком комментариев, поэтому Fetchmail не обрабатывает первые две строки. Остальную часть листинга 11.3 можно условно разделить на две категории. Выражения set устанавливают глобальные опции, большинство из которых соответствуют опциям, содержащимся в разделе Fetchmail Run Controls диалогового окна Fetchmail Expert Configurator программы fetchmailconf (рис. 11.2). Многие из этих опций можно установить с помощью параметров командной строки. Кроме них, в листинге 11.3 содержатся два выражения poll, каждое из которых определяет учетную запись на удаленном компьютере, используемую для получения почты. При работе с программой fetchmailconf эта информация задается в диалоговых окнах Fetchmail Host *Имя_узла* и Fetchmail User *Имя_пользователя* Querying *Имя_узла* (рис. 11.3 и 11.4). Выражение poll может занимать несколько строк, причем специальный символ, указывающий на то, что выражение продолжается в другой строке, не требуется. Переход на новую строку может осуществляться в произвольных точках выражения.

Fetchmail поддерживает большое количество опций и в этом разделе невозможно обсудить их все. Исчерпывающую информацию об опциях данной программы вы можете получить на страницах справочной системы, посвященных Fetchmail, а также в других документах. Значением некоторых опций является строка символов (например, посредством опции может задаваться пользовательское имя). Если в строке содержатся пробелы, строка помещается в кавычки. Некоторые из наиболее важных глобальных опций описаны ниже.

- set postmaster *имя_пользователя*. Данная опция позволяет задать имя пользователя, который будет получать письма в случае, если определить адресата не удается. На это же имя будут приходить и некоторые сообщения об ошибках в работе программы. Как правило, в данной опции задается обычное пользовательское имя, но при желании вы можете указать postmaster или root. (При настройке сервера SMTP также задается пользователь postmaster, который получает сообщения, касающиеся работы почтовой системы. В роли postmaster для Fetchmail и сервера SMTP может выступать либо один и тот же, либо разные пользователи.) Значение данной опции можно изменить с помощью параметра командной строки --postmaster *имя_пользователя*.

- set bouncemail. Данная опция указывает на то, что сообщения об ошибках должны передаваться отправителям писем. Альтернативой set bouncemail является выражение set no bouncemail, в этом случае сообщения об ошибках будет получать пользователь postmaster, указанный при настройке Fetchmail.

- set daemon *интервал*. Эта опция сообщает о том, что программа Fetchmail должна выполняться в режиме демона и опрашивать почтовый сервер через заданные интервалы времени (значение интервала указывается в секундах). Если вы хотите, чтобы программа Fetchmail выполнялась в пакетном режиме, данная опция должна отсутствовать в конфигурационном файле. Переопределить значение опции set daemon можно, задавая в командной строке параметр --daemon *интервал*. Если при вызове Fetchmail указан параметр --daemon 0, это означает, что программа должна выполнить одну операцию, связанную с опросом сервера, даже если в .fetchmailrc задан режим демона.

- set logfile *имя_файла*. Данная опция указывает на то, что протокол работы программы Fetchmail должен записываться в файл с указанным именем.

- set syslog. Если вы хотите регистрировать действия Fetchmail в системном файле протокола, вы можете сделать это посредством данной опции.

В файле .fetchmailrc могут присутствовать выражения poll различной сложности. Формат данного выражения приведен ниже.

poll *имя_сервера опции-сервера описание_пользователя*

Ключевое слово server является синонимом poll. Вы также можете заменить его на skip, в результате чего Fetchmail пропустит данную запись. Таким способом вы можете временно исключить запись из рассмотрения, не удаляя ее из файла .fetchmailrc. Опции сервера определяют особенности взаимодействия Fetchmail с сервером, а в описании пользователя приводится информация об учетных записях на сервере и на локальном компьютере. В пределах каждой из категорий порядок следования записей не имеет значения, но чередовать записи, принадлежащие разным категориям, нельзя. (Именно этим вызвано большинство проблем, возникающих при редактировании файла .fetchmailrc вручную.) Слова and, with, has, wants и options игнорируются; не принимаются также во внимание символы ":", ";" и ",". Вы можете свободно использовать их в составе опций сервера или описания пользователя для того, чтобы сделать выражение poll более удобным для восприятия.

Некоторые из наиболее важных опций сервера приведены ниже.

- proto *имя* или protocol *имя*. Эти опции, являющиеся синонимами, определяют используемый протокол получения почты. В большинстве случаев в качестве имени указывается POP3 или ШАР, но Fetchmail также поддерживает и другие значения данной опции. Переопределить значение, заданное в файле .fetchmailrc, можно указав в командной строке параметр -p.

- interface *интерфейс/IP-адрес/маска_подсети* Данная опция позволяет задать интерфейс, который должен быть активен в тот момент, когда Fetchmail опрашивает сервер. Для указания интерфейса используются имя устройства, например eth1 или ppp0, а также IP-адрес и маска подсети, которые определяют диапазон допустимых IP-адресов. Например, выражение eth1/192.168.1.0/255.255.255.0 означает, что перед тем, как Fetchmail предпримет попытку опроса сервера, с устройством eth1 компьютера должен быть связан адрес в диапазоне от 192.168.1.1 до 192.168.1.254. Аналогичные сведения можно предоставить программе с помощью опции -I, задаваемой в командной строке.

- monitor *интерфейс*. Данная опция указывает программе Fetchmail, выполняющейся в режиме демона, на то, что она должна проверять активность сетевого интерфейса. Если Fetchmail обнаружит, что после предыдущего опроса интерфейс стал неактивен, она пропустит очередной планируемый опрос. Значение опции может быть переопределено путем указания в командной строке опции -M.

Ниже перечислены опции, наиболее часто применяющиеся в описании пользователя.

- user *имя*, или username имя. Эта опция задает пользовательское имя и обычно помечает начало описания пользователя в выражении poll. Как правило, данная

опция определяет имя на удаленном узле, но если она сопровождается ключевым словом here, предполагается локальное имя. Ключевое слово there подтверждает тот факт, что имя зарегистрировано на удаленном компьютере. Опция -и в командной строке переопределяет значение данной опции.

- pass *пароль,* или password *пароль.* Эта опция определяет пароль, соответствующий учетной записи сервера получения почты. Пароль хранится в конфигурационном файле в незашифрованном виде.

- is *имя* или to имя. Эти опции связывают учетную запись на сервере с локальным пользователем. Одна из этих опций указывается после описания учетной записи на сервере получения (т. е. после выражения user имя with pass *пароль).* Если удаленная учетная запись указывается перед локальной, ключевое слово here, заданное после данной опции, идентифицирует учетную запись как локальную. Ключевое слово there задает удаленную учетную запись.

- smtphost имя_узла. В обычных условиях программа Fetchmail пытается использовать для передачи почты компьютер, на котором она выполняется, т. е. узел с адресом localhost. Данная опция указывает на то, что почтовый сервер, посредством которого должны передаваться сообщения, находится на компьютере с заданным именем. Вы можете указать в качестве значения данной опции имя вашего компьютера. В этом случае в заголовках писем, переданных с помощью Fetchmail, вместо localhost будет содержаться обычное имя узла. Значение данной опции переопределяется с помощью опции -S, задаваемой в командной строке.

- keep. По умолчанию после получения сообщений Fetchmail удаляет их с сервера. Данная опция указывает на то, что сообщения должны сохраняться. Ее можно задавать, например, при тестировании новой конфигурации. Опция -k, введенная в командной строке, представляет собой альтернативу опции keep.

- fetchall. В обычных условиях Fetchmail не копирует сообщения, которые были получены ранее. Опция fetchall указывает на то, что должны быть получены все письма с сервера. Аналогичные действия выполняет опция -a, заданная в командной строке.

- forcecr. Строки почтовых сообщений должны оканчиваться парой символов CR/LF (возвратом каретки и переводом строки). Многие почтовые программы допускают отсутствие символа возврата каретки, поэтому подобные сообщения иногда встречаются в сети. Сервер передачи qmail отвергает такие сообщения; исправить положение позволяет опция forcecr.

Если вы зададите больше одного локального имени, Fetchmail будет анализировать заголовки писем и пытаться определить, кто является получателем конкретного сообщения. Например, если вы укажете локальные учетные записи jack и j ill и если письмо поступает на имя jill, Fetchmail доставит его пользователю jill. Режим, в котором письма, поступающие на одну учетную запись, обрабатываются по-разному в зависимости от содержимого их заголовков, называют *многоточечным* (multidrop mode).

Доставку писем, обработанных с помощью Fetchmail, может выполнять программа Procmail, которая будет описана в главе 19. Procmail предоставляет возможность идентифицировать и удалять нежелательные сообщения, распределять поступающие письма по папкам и выполнять другие подобные действия.

Резюме

Серверы получения почты часто инсталлируют на том же компьютере, на котором располагается главный сервер SMTP. В результате пользователи могут просматривать свою почту с клиентских машин, подключенных к сети организации, и даже из Internet. Поддержка протокола получения почты избавляет пользователей от необходимости регистрироваться на сервере с помощью Telnet, SSH или других средств удаленного доступа и дает возможность запускать программы просмотра писем на своих компьютерах. Наиболее популярными протоколами получения почты в настоящее время являются POP-3 и IMAP-4. Протокол IMAP предоставляет пользователям более обширные возможности обработки почты по сравнению с POP, но он предъявляет более высокие требования к объему жесткого диска и пропускной способности линии связи. По этой причине администраторы предпочитают устанавливать на своих компьютерах серверы POP.

Программа Fetchmail служит своеобразным "мостом" между почтовым сервером, поддерживающим протокол получения почты, и другими почтовыми системами, в частности почтовой системой локальной сети. Как правило, Fetchmail используется для извлечения писем с почтового сервера и включения их в локальную очередь. При этом работа пользователей с почтовыми сообщениями упрощается.

Глава 12

Поддержка сервера новостей

В главе 11 обсуждалась работа серверов получения почты. Эти серверы позволяют пользователям принимать сообщения, адресованные непосредственно им. В данной главе рассматриваются средства обработки сообщений другого типа — *серверы новостей.* Если электронная почта обеспечивает взаимодействие типа "один к одному", то служба новостей (Usenet) реализует среду для обмена сообщениями "один ко многим". Если пользователь отправит сообщение на сервер новостей, любой другой пользователь сможет прочитать его. Более того, серверы новостей взаимодействуют между собой, в результате чего сообщения распространяются по всему миру. Установив сервер новостей, вы предоставите своим пользователям удобный инструмент для взаимодействия.

Традиционно серверы новостей взаимодействуют друг с другом как равноправные партнеры, обмениваясь всеми *сообщениями,* или *статьями,* переданными в группы. Эти серверы могут работать с многочисленными клиентскими программами, расположенными как на локальном узле, так и на других компьютерах. Как правило, клиенты получают с сервера относительно небольшое число хранящихся на нем материалов. Возможен также вариант сервера новостей, который обращается к другим серверам как клиентская программа и копирует лишь подмножество сообщений, необходимое для локальных пользователей. Такие серверы с усеченными функциональными возможностями часто выполняются на персональных компьютерах и позволяют пользователю читать новости в автономном режиме. В отличие от полнофункциональных серверов новостей, с которыми клиент должен поддерживать постоянное соединение в течение всего времени просмотра, данный сервер дает возможность скопировать сообщения на локальный компьютер, а затем просматривать их, отключившись от сети. Такой подход позволяет минимизировать время работы в сети, что важно при поминутной оплате за предоставление сетевых услуг.

Следует заметить, что обычная электронная почта также может обеспечить режим взаимодействия "один ко многим". Составляя письмо, вы можете указать в нем несколько получателей. Такой режим, называемый *списками рассылки,* позволяет организовать дискуссионные группы, аналогичные группам Usenet. В отличие от Usenet, процедура подписки на списки рассылки практически не стандартизована.

Использование сервера новостей

Сервер новостей предоставляет следующие возможности.

- **Поддержка групп новостей Usenet.** Глобальная сеть серверов новостей носит название Usenet. Если сервер принимает участие в обмене сообщениями, он тем самым способствует распространению материалов групп по всему миру. Серверы новостей Usenet поддерживают многие провайдеры, университеты, а также коммерческие предприятия.

- **Взаимодействие внутри организации.** Организация может установить у себя сервер новостей для того, чтобы сотрудники могли обмениваться информацией, в частности участвовать в обсуждении производственных вопросов. Так, например, с помощью такого сервера могут взаимодействовать участники одного проекта. Часто серверы новостей используются для проведения дискуссий о товарах, выпускаемых компанией, предоставляя пользователям информацию о продукции предприятия.

- **Чтение материалов новостей в автономном режиме.** Как было сказано ранее, для отдельных пользователей можно организовать копирование сообщений на локальный сервер и просмотр их в автономном режиме. Такой подход возможен только в том случае, если пользователям должно предоставляться ограниченное количество групп новостей.

Независимо от того, обеспечивает ли сервер полную поддержку Usenet или организует обмен материалами внутри предприятия, для него используется одно и то же программное обеспечение. Различие состоит лишь в наличии или отсутствии обмена с другими серверами. Полнофункциональный сервер Usenet предъявляет гораздо более высокие требования к ресурсам, чем сервер, поддерживающий ограниченное число групп новостей. Только для хранения содержимого материалов Usenet нужен жесткий диск на десятки и даже сотни мегабайт. Для того чтобы сервер новостей можно было разместить на обычном компьютере, подключенном посредством линий со средней пропускной способностью, необходимо ограничить число групп новостей, поддерживаемых этим сервером.

ВНИМАНИЕ

Как правило, серверы новостей хранят материалы групп в каталоге /var/spool/news. Если при установке Linux вы не планировали инсталляцию сервера новостей, то этот каталог, вероятнее всего, находится в корневом разделе либо в небольшом разделе в каталоге /var. На компьютере, специально предназначенном для сервера новостей, каталог /var или /var/spool/news размещается в разделе диска очень большого объема. Принимая решение об установке сервера, необходимо убедиться, что в разделе, в котором находится каталог /var/spool/news, имеется достаточно места, либо задать другой каталог в файловой системе компьютера.

СОВЕТ

Ведение журнала файловых систем существенно уменьшает время загрузки после аварийного отключения компьютера (например, при сбое в системе питания). Это особенно важно для компьютеров с большим объемом жесткого диска, на которых устанавливаются, например, серверы новостей. Для буфера новостей желательно выделить отдельный раздел; это увеличит стабильность всей системы в целом, так как ошибка в соответствующем каталоге не повлияет на работу остальных компонентов системы.

Поскольку для полнофункционального сервера Usenet необходимо выделять мощный компьютер, подключенный по линии, которая позволяет передавать мегабиты информации в секунду, такие серверы практически никогда не устанавливаются в небольших офисах. Пользователи, работающие в сетях небольших организаций либо на домашних компьютерах, обычно обслуживаются серверами, расположенными у провайдеров, либо независимыми серверами. Среди серверов новостей, работающих на коммерческой основе, можно отметить Giganews (`http://www.giganews.com`), Supernews (`http://www.supernews.com`) и NewsGuy (`http://www.newsguy.com`). Информацию о серверах, предоставляющих свои услуги бесплатно, можно получить, обратившись по адресу `http://www.newsservers.net`. На узле `http://groups.google.com` хранятся архивы многих популярных групп новостей. Для доступа к ним предоставляется Web-интерфейс. Несмотря на то что в данной главе при обсуждении серверов новостей основное внимание будет уделяться использованию этих серверов для внутреннего обмена новостями, вопросы настройки сервера для обмена данными с серверами Usenet также будут рассмотрены.

Работа с материалами групп новостей в автономном режиме обычно обеспечивается клиентскими программами новостей (программами просмотра), но для этой цели также можно использовать сервер новостей с ограниченными функциональными возможностями, например, программу Leafnode, которая будет рассматриваться ниже в этой главе. С точки зрения клиента программы, подобные Leafnode, работают так же, как и обычные серверы новостей, но их конфигурация отличается от конфигурации серверов Usenet. Если ваша сеть постоянно подключена к Internet, вам, возможно, не придется устанавливать специальный сервер; проще будет использовать сервер новостей провайдера. Собственный сервер новостей полезен, если вашим пользователям необходимо работать с материалами групп, а сервер провайдера перегружен. В этом случае вы можете сконфигурировать Leafnode так, чтобы копирование материалов выполнялось в наиболее удобное время. Использование внутреннего сервера новостей оправдано также в том случае, когда большинство пользователей просматривают материалы одних и тех же групп. Копируя эти группы на внутренний сервер, вы уменьшите сетевой трафик.

Принцип работы протокола NNTP

Современные серверы новостей используют для обмена между собой и для взаимодействия с клиентами протокол NNTP (Network News Transfer Protocol — протокол передачи сетевых новостей). Как правило, серверы NNTP используют порт 119. Следует заметить, что распространение групп новостей не всегда осуществлялось посредством протокола NNTP. Более того, на ранних этапах развития данной службы материалы передавались в сетях, отличных от TCP/IP. Несмотря на то что NNTP — не единственный протокол, применяемый для поддержки новостей, в сетях TCP/IP он используется для этой цели наиболее часто.

При работе протокола NNTP происходит обмен *сообщениями,* которые также называют *статьями.* Сообщение — это отдельный документ, автором которого является один пользователь. (Существуют средства для работы нескольких пользователей над одним документом, но на практике подобное взаимодействие осуществляется крайне редко.) Сообщения объединяются в группы. Одно сообщение может быть отправлено одновременно в несколько групп, но такое дублирование во многих случаях нежелательно. Группы

новостей, в свою очередь, объединяются в категории, организуя иерархию групп. Полное имя группы состоит из нескольких компонентов, разделяемых точками. Имена групп создаются по тому же принципу, что и имена каталогов файловой системы. В начале расположено имя, определяющее общую тему, а затем тема уточняется. Например, группы `comp.os.linux.misc` и `comp.os.linux.hardware` принадлежат категории `comp.os.linux`, и темы этих групп сходны. Материалы группы `comp.dcom.modems` существенно отличаются от `comp.os.linux.misc` и `comp.os.linux.hardware`, а группа `rec.arts.sf.dune` не имеет ничего общего с перечисленными выше группами.

Когда пользователь посылает сообщение, сервер добавляет к нему поле заголовка `Message-Id`, содержащее идентификационный код. Этот код состоит из последовательного номера, генерируемого сервером, и имени сервера. Поскольку идентификатор содержит имя сервера, он является уникальным во всей системе Usenet. Посредством идентификаторов сервер новостей определяет, какие сообщения были просмотрены и какие должны быть переданы клиенту.

Для взаимодействия серверов новостей используются два типа протокола NNTP: *протокол передачи* (push protocol) и *протокол получения* (pull protocol). Во время передачи данных один из серверов выступает в роли клиента, а другой — в роли сервера. При использовании протокола передачи клиент сообщает серверу о каждом имеющемся у него сообщении, передавая его идентификатор. Сервер ищет это сообщение в своей базе данных и определяет, нужно ли оно ему. Процесс повторяется для каждого сообщения, которые присутствуют на сервере, выполняющем в процессе взаимодействия роль клиента. При этом производятся многочисленные обращения к базе данных. Альтернативой протоколу передачи является протокол получения, при использовании которого принимающий сервер выступает в роли клиента. Сначала принимающая система получает полный список сообщений, поступивших на сервер с указанного момента, а затем запрашивает конкретные сообщения. Такой протокол работает более эффективно, но при этом необходимо принимать меры предосторожности, чтобы сервер не передал сообщение, предназначенное для внутреннего использования.

Поскольку пользователи постоянно присылают новые сообщения, сервер должен удалять устаревшие статьи, в противном случае жесткий диск быстро переполнится. (На самом деле угроза переполнения диска существует даже тогда, когда старые сообщения периодически удаляются с сервера.) Обычно сообщения удаляются с сервера через определенное время после их поступления. Время хранения сообщений на сервере зависит от многих факторов, например, от имеющегося в наличии дискового пространства, от количества поддерживаемых групп новостей, от трафика, связанного с передачей сообщений в эти группы, и от того, насколько популярна та или иная группа среди пользователей. Серверы новостей позволяют устанавливать для разных групп новостей различное время хранения материалов.

Независимо от используемого протокола и времени хранения сообщений, в передаче материалов групп может участвовать несколько компьютеров. Серверы новостей взаимодействуют друг с другом, образуя сложную структуру. Передача материалов групп от одного сервера другому называется *поставкой новостей*. Как правило, небольшие серверы, подключенные через линии с относительно невысокой пропускной способностью, получают материалы групп у крупных серверов. Например, сервер, находящийся в небольшом колледже (назовем его условно Tiny College), может получать содержимое групп новостей у сервера большого университета (Pangaea University). Это означает, что основная часть сообщений, находящихся на сервере `news.tiny.edu`, получена с серве-

ра news.pangaea.edu. Однако часть данных может передаваться и в противоположном направлении, так как пользователи news.tiny.edu посылают свои сообщения в группы и эти сообщения должны быть доставлены на сервер news.pangaea.edu. Кроме того, в Tiny College могут поддерживаться свои группы новостей; при передаче этих групп news.tiny.edu станет поставщиком для news.pangaea.edu. Pangaea University, в свою очередь, получает и т. д.

В службе новостей не соблюдается строгая иерархия. Например, не исключено, что на сервере news.pangaea.edu не поддерживаются некоторые группы, в которых испытывают необходимость пользователи news.tiny.edu. В этом случае администратор Tiny College должен принять меры для получения материалов этих групп с другого сервера. То же самое делает администратор Pangaea University для получения групп, необходимых его пользователям. Не все группы, находящиеся у поставщика новостей, должны быть переданы. С целью экономии дискового пространства получатель может отказаться от некоторых групп и даже от целых категорий групп, не требующихся пользователям.

В результате взаимодействия серверов новостей формируется система, в которой несколько крупных серверов являются поставщиками новостей для других серверов меньшего размера. Эти серверы, в свою очередь, поставляют материалы групп другим серверам и т. д. Любой из серверов новостей может обслуживать клиентские программы. Клиентские программы, или программы просмотра новостей, также используют протокол NNTP. Они могут принимать и передавать сообщения, но не могут выступать в качестве поставщиков новостей. Помимо обмена материалами групп, серверы также модифицируют заголовки сообщений, включая в них идентификаторы серверов, а часто и идентификаторы клиентов, с которых были переданы сообщения.

Необходимо помнить, что поток сообщений передается в обоих направлениях. Если бы новые сообщения не передавались от сервера, получающего материалы, серверу, выступающему в роли поставщика, то у поставщика не было бы материалов групп. Крупные серверы выполняют функции "накопителей", на которые стекаются новые сообщения от пользователей. Но следует заметить, что небольшие серверы получают от поставщиков новостей гораздо больший объем материалов, чем они генерируют сами.

Сервер INN

Среди серверов новостей, предназначенных для выполнения в системе Linux, наиболее популярным является InterNetNews, или INN (http://www.isc.org/products/INN/). Пакет INN состоит из нескольких программ, работающих совместно. Основная программа, innd, предназначена для обработки новых статей и поддержки соединений. Программа nnrpd обслуживает соединения с программами просмотра новостей. Для инициализации соединений с другими серверами применяется программа innxmit, которая, в свою очередь, использует для решения многих задач nntpsend. Для каждой из указанных программ предусмотрен отдельный конфигурационный файл, некоторые из них используют несколько файлов. Основные конфигурационные файлы находятся в каталоге /etc/news, но некоторые файлы располагаются также в /var/lib/news и других каталогах.

Сервер INN в составе операционной системы Linux обычно поставляется в пакете под названием inn. В этой главе описана версия 2.2.2 INN, но конфигурация остальных реализаций INN 2.x практически не отличается от INN 2.2.2. В некоторых системах INN

настраивается для совместной работы с Cleanfeed. Cleanfeed — это дополнительный пакет, автоматически удаляющий с сервера новостей некоторые типы спама. (Спам в составе групп новостей создает большие проблемы для администраторов. Работая с материалами групп, вы не замечаете сообщений, содержащих навязчивую рекламу, лишь потому, что существуют средства, эффективно удаляющие их.)

Получение материалов групп

Если вы устанавливаете на своем компьютере сервер новостей, который должен поддерживать хотя бы часть Usenet, вам необходимо найти поставщика новостей и сконфигурировать свой сервер для работы с ним. Настройка сервера для получения материалов групп будет обсуждаться далее в этой главе, а здесь мы кратко рассмотрим вопросы выбора поставщика. Протокол NNTP позволяет использовать для получения групп любой сервер новостей, однако не следует думать, что вы можете указать в конфигурационном файле произвольные серверы. Вам необходимо найти такой сервер, администратор которого согласился бы предоставлять вам необходимые материалы.

Поиск сервера на роль поставщика новостей лучше всего начать с провайдера. Если быстродействие вашего соединения достаточно для того, чтобы вы могли получать материалы требуемых групп, провайдер предоставит их вам или, по крайней мере, укажет сервер новостей, услугами которого вы могли бы воспользоваться. Многие провайдеры, в особенности те, которые предоставляют услуги Internet небольшим компаниям, не содержат своего сервера новостей, так как пропускная способность их линий не позволяет им этого. В этом случае поставщиком новостей может быть независимый сервер, например NewsGuy (`http://www.newsguy.com`).

Поддержка групп новостей при наличии ограниченных ресурсов

Если вы хотите поддерживать сервер новостей, имея в своем распоряжении ограниченные аппаратные ресурсы и недостаточно высокую пропускную способность линии, вам следует отказаться от так называемых *двоичных групп*. Посредством таких групп распространяются двоичные файлы, содержащие звуковые, клипы, оцифрованные фотоснимки, коды программ и т. д. Материалы подобных групп создают трафик большого объема, поскольку размеры двоичных файлов существенно превышают средние размеры текстовых сообщений, передаваемых в обычные группы. В именах большинства двоичных групп присутствует имя `binary` или `binaries`. Как правило, такие группы находятся в категории `alt`, но они также встречаются в `comp` и других категориях.

Если же вы собираетесь предоставлять пользователям материалы групп в полном объеме, но имеющиеся ресурсы не позволяют сделать это, вы можете организовать получение материалов из внешнего источника (outsourcing). Для этого надо заключить соответствующий договор с поставщиком новостей и создать на своем сервере NDS-запись, указав в ней IP-адрес сервера поставщика. У пользователей создастся впечатление, что они работают с вашим сервером новостей, но на самом деле основную работу по их обслуживанию будет выполнять внешний сервер. Так часто поступают провайдеры, предоставляющие услуги небольшим компаниям. Недостаток подобного подхода состоит в том, что администратор внешнего сервера может не согласиться поддерживать группы новостей для локального использования абонентами вашей сети.

Чтобы организовать полнофункциональный сервер новостей, необходимо затратить значительные средства как на приобретение оборудования и организацию быстродейству-

ющего соединения, так и на оплату услуг поставщика новостей. Например, на момент написания данной книги стоимость получения материалов групп на сервере NewsGuy составляла 1200 долларов в месяц, а для взаимодействия с этим сервером требовался компьютер не ниже, чем Pentium 400, с объемом оперативной памяти не менье 500 Мбайт, оснащенный жестким диском объемом не меньше 64 Гбайт. Для получения данных требовалась пропускная способность соединения не ниже 3 Мбод. Планируя пропускную способность соединения, необходимо также учитывать, что на сервер будут обращаться клиенты. В зависимости от набора поддерживаемых вами групп новостей, времени хранения сообщений и других характеристик сервера, требования к ресурсам могут увеличиваться или уменьшаться. Поскольку количество групп и объем сообщений в них ежегодно возрастает, не исключено, что приведенные выше требования вскоре придется пересмотреть.

Настройка INN

Настройка INN предполагает установку большого количества опций в различных конфигурационных файлах. В пакете, предназначенном для инсталляции, указаны такие значения опций, которые практически обеспечивают функционирование сервера. Вам остается лишь привести некоторые установки в соответствии с требованиями системы. Например, вам понадобится объявить поддерживаемые группы новостей и настроить средства для контроля доступа. (Если вы хотите установить полнофункциональный Usenet-сервер, администратор поставщика новостей, скорее всего, сообщит вам установки, необходимые для доступа к его серверу.) Необходимо также определить политику удаления сообщений по истечении срока их действия и сообщить INN, как следует обрабатывать специальные управляющие сообщения (например, команды на удаление статей, создание новых групп и т. д.).

Общие установки

Основным конфигурационным файлом является `/etc/news/inn.conf`. В этом файле содержатся выражения следующего вида:

`имя_опции: значение`

Такой же синтаксис используется и в других конфигурационных файлах. Значения большинства опций в файле `inn.conf`, заданные по умолчанию, не нуждаются в редактировании. Наиболее важные из опций, которые вам придется установить в соответствии с требованиями вашей системы, описаны ниже.

- `organization`. Данная опция позволяет указать имя вашей организации. Строка, заданная в качестве значения опции `organization`, включается в заголовки всех сообщений, переданных через ваш сервер.

- `server`. Имя компьютера, на котором выполняется сервер INN. Эта опция очень важна, поскольку имя, указанное в ней, используется при установлении соединения, которое необходимо для доставки сообщений. Настраивая сервер, можно оставить значение по умолчанию `localhost`, но лучше заменить его реальным именем вашего компьютера.

- `pathhost`. Получая сообщение, сервер INN включает значение данной опции в поле заголовка Path. Это поле позволяет выяснить путь сообщения и устранить си-

туации, при которых сообщение вторично попадает на тот же сервер. Задавая значение данной опции, желательно указать полное доменное имя сервера, например news.threeroomco.com.

- moderatormailer. Некоторые группы новостей *модерируются*, т. е. перед тем, как отправленные сообщения становятся доступными всем пользователям, их проверяет один из участников административной группы — *модератор*. Если вам надо связаться с модератором, вы можете либо послать письмо непосредственно ему, либо отправить сообщение на централизованный адрес модерируемой группы; в результате оно будет доставлено модератору. Примером централизованного адреса может служить %s@uunet.uu. net.

- domain. С помощью данной опции задается имя домена, например threeroomco. com. Оно предназначено для внутреннего использования компонентами INN.

- fromhost. Когда локальный пользователь передает сообщение, INN создает поле заголовка From, идентифицирующее отправителя. Значение данного поля интерпретируется как имя компьютера, поэтому вы можете указать в качестве значения данной опции имя домена или имя почтового сервера.

- complaints. Работая с группами новостей, некоторые пользователи совершают недопустимые действия, например, публикуют рекламные сообщения в группах, не предназначенных для этой цели, посылают двоичные файлы в группы, ориентированные на работу с текстовыми данными, передают сообщения, оскорбляющие других пользователей, и т. д. Опция complaints позволяет указать почтовый адрес, по которому пользователи смогут связаться с вами и сообщить о некорректном поведении других участников группы.

В файле inn. conf предусмотрено много других опций, но для них обычно принимают значения, установленные по умолчанию. Дополнительную информацию о назначении различных опций вы можете получить на страницах справочной системы, посвященных inn. conf.

Объявление групп новостей

В файле inn. conf отсутствуют объявления и описания групп новостей. Эта информация указывается в двух других конфигурационных файлах: active и newsgroups. Данные файлы хранятся в каталоге, указанном с помощью опции pathdb, которая находится в файле inn. conf (обычно это каталог /var/lib/news).

Файл active содержит список групп новостей, поддерживаемых системой. Каждой группе новостей посвящена строка этого файла. Порядок следования строк не важен. Строка, содержащая объявление группы, состоит из четырех полей:

имя_группы максимальный_номер минимальный_номер флаги

В первом поле указывается полное имя группы, например comp. os. linux. misc. Два следующих поля содержат соответственно максимальный и минимальный номера сообщений, присутствующих в группе. Для новой группы значения этих полей равны соответственно 0000000000 и 0000000001. (Сервер INN хранит сообщения, переданные в группу, как отдельные файлы, имена которых создаются на базе номеров сообщений в локальной группе. Эти номера не связаны с идентификаторами сообщений и могут

по-разному присваиваться на различных серверах.) Последнее поле содержит флаги, определяющие характеристики группы. Значения флагов описаны ниже.

- y. Данный флаг присутствует в объявлениях групп чаще других. Он указывает на то, что пользователи могут передавать сообщения в группу.
- n. Группа, помеченная этим флагом, может получать новые статьи с других серверов, но локальные пользователи не могут передавать в нее свои сообщения.
- m. Этот флаг определяет **модерируемую** группу новостей. Локальные сообщения, переданные в группу, пересылаются модератору для проверки.
- j. Группа, помеченная данным флагом, может принимать сообщения, но не обрабатывает их. Сервер INN лишь пересылает принятые сообщения на сервер, выполняющий роль поставщика данной группы новостей.
- x. Данный флаг определяет статическую группу. Новые сообщения не принимаются ни от локальных пользователей, ни от поставщика новостей.
- =*группа.новостей*. Сообщения, отправленные в группу, которая помечена данным флагом, передаются в указанную группу. Этот флаг можно использовать для автоматического перенаправления материалов.

Сервер новостей, который допускает только локальные операции, может поддерживать лишь несколько групп. Имена этих групп задаются произвольно, но они должны соответствовать схеме, принятой в Usenet. При желании вы можете назначать локальным группам имена, начинающиеся с имени вашей организации, например, на сервере новостей, расположенном в домене `threeroomco.com`, могут поддерживаться группы `threeroomco.support`, `threeroomco.support.bigproduct` и `threeroomco.accounting`. Если вы получаете материалы новостей из внешнего источника, соответствующий сервер должен предоставлять вам список групп новостей или даже полностью сформированный файл `active`.

В процессе работы сервера INN значения полей **максимальный_номер** и **минимальный_номер** в файле `active` могут изменяться. При добавлении нового сообщения значение поля **максимальный_номер** увеличивается, а по истечении срока действия имеющегося сообщения увеличивается число, указанное в поле **минимальный_номер**. При удалении сообщения значение поля **минимальный_номер** может остаться неизменным; как было сказано ранее, сообщения не обязательно удаляются в порядке их поступления.

Файл `newsgroups` используется сервером реже, чем файл `active`. Как и в файле `active`, первым полем каждой записи `newsgroups` является имя группы. После имени группы расположен один или несколько символов табуляции, за которым следует описание группы. Клиенты получают сведения из этого файла, чтобы предоставить пользователям информацию о группах.

Управление доступом

Чтобы предотвратить незаконные действия с материалами групп и сэкономить ресурсы, многие администраторы ограничивают доступ к серверам новостей. Рассматривая вопросы ограничения доступа, следует принять во внимание три задачи, выполняемые сервером: предоставление материалов групп другим серверам, получение материалов с других серверов и организация взаимодействия с клиентом. Первые две задачи необходимо

учитывать лишь тогда, когда ваш сервер обменивается новостями с другими серверами. Если же вы организуете независимый сервер, вам необходимо убедиться, что его конфигурация не предполагает обмена данными с другими серверами новостей. Взаимодействие с клиентами осуществляет любой сервер новостей (это необходимо предусмотреть, редактируя конфигурационные файлы).

Предоставление материалов групп другим серверам

Если вы хотите, чтобы сообщения, переданные вашими пользователями, достигали других узлов, тем более, если вы собираетесь предоставлять другим серверам материалы групп в полном объеме, вам надо соответствующим образом сконфигурировать ваш сервер новостей. Для этого необходимо отредактировать содержимое файла /etc/news/newsfeeds. В файле **/etc/news/newsfeeds** находятся записи, представленные в следующем формате:

идентификатор_узла:*шаблон*[,*шаблон*...]:*флаг*[,*флаг*...]:*параметр*

Длина подобных записей может быть достаточно большой. Для того чтобы разместить запись в нескольких строках, надо использовать символ \. Если данный символ располагается в конце строки, это означает, что следующая строка является ее продолжением. Размещение записи в нескольких строках делает запись более удобной для восприятия и упрощает редактирование файла. Назначение каждого из полей описано ниже.

- *идентификатор_узла*. В этом поле указывается идентификатор, или код узла. Код не обязательно должен соответствовать реальному имени узла, он лишь должен совпадать со значением соответствующего поля другого конфигурационного файла.

- *шаблон*. Шаблон определяет одну или несколько групп новостей. Если количество групп, поддерживаемых сервером, невелико, вы можете указывать имя каждой группы, в противном случае следует применять символ групповой операции (*). Например, comp.os.* определяет все группы в категории comp.os. Если перед шаблоном указан символ !, это означает, что материалы данных групп не должны передаваться на другой сервер; исключение составляют лишь сообщения, переданные одновременно в несколько групп. Аналогичный результат получается при использовании символа @, но при этом сообщения, переданные в несколько групп, также блокируются. Предположим, например, что вы задали в данном поле значение !comp.os.linux. Если сообщение направлено в группы comp.os.linux и comp.os.linux.hardware, оно появится лишь в составе группы comp.os.linux.hardware. Значение @comp.os.linux полностью запретит передачу данного сообщения. Сервер INN интерпретирует записи в файле newsfeeds последовательно одну за другой, поэтому если вы укажете comp.os.*, !comp.os.linux, INN разрешит передачу всех сообщений категории comp.os, за исключением группы !comp.os.linux. Изменив порядок следования записей на обратный, вы разрешите передачу всех групп, так как более общее выражение comp.os.* переопределит более конкретное выражение !comp.os.linux.

- *флаг*. В данном поле задается один или несколько флагов; эти флаги ограничивают типы сообщений, которые могут быть переданы на удаленный узел. Например, выражение <*размер* ограничивает набор передаваемых сообщений теми, размер которых меньше указанного, а выражение G*число* указывает на то, что сообщение, направленное в несколько групп, передается только в том случае, если количество

групп не превышает указанное. Описания всех флагов можно найти в справочной системе.

- *параметр*. Значение этого поля зависит от типа передаваемых данных. Обычно это имя файла, который содержит информацию, предназначенную для передачи на удаленный узел. В ряде случаев параметр не указывается. В файле newsfeeds, поставляемом в составе пакета, находится много закомментированных записей, которые могут быть использованы в качестве примеров.

Файл newsfeeds управляет созданием файла, который должен быть передан другому серверу. Информация, заданная в файле /etc/news/nntpsend.ctl, определяет порядок взаимодействия INN с этим сервером. Подобно newsfeeds, файл nntpsend.ctl содержит записи, состоящие из нескольких полей, разделенных двоеточиями. Формат записи приведен ниже.

идентификатор_узла:*имя_узла*:*максимальный_размер*:[*параметры*]

Значение в поле *идентификатор_узла* должно совпадать со значением, заданным в одноименном поле файла newsfeeds, а *имя_узла* — это реальное имя узла. Поле *максимальный_размер* позволяет ограничить объем данных, передаваемых в течение одного сеанса обмена; например, значение 2т ограничивает объем данных двумя мегабайтами. Последнее поле содержит необязательные параметры, которые могут передаваться программе innxmit, выполняющей реальную передачу данных. Сведения об этих параметрах можно найти в справочной системе.

Содержимое рассмотренных выше конфигурационных файлов вам придется менять, если вы собираетесь предоставлять материалы групп другим серверам или получать новости у них. Для того чтобы ваш сервер работал эффективно, серверы, поставляющие вам группы новостей, должны получать у вас сообщения, отправленные вашими пользователями. Без этого сообщения ваших пользователей будут доступны только в пределах локального сервера, а пользователи Internet не получат их. Таким образом, даже в том случае, когда вы лишь получаете группы новостей с внешнего сервера, вы должны предусмотреть в конфигурационном файле возможность передачи сообщений в эти группы.

Установка опций, управляющих доступом к серверу

Сервер INN может управлять доступом со стороны других серверов и клиентов. Основной демон, innd, принимает обращения от внешних серверов, предоставляющих материалы групп новостей, и от других программ, входящих в состав пакета INN. Несмотря на то что innd также принимает обращения от клиентов, он практически сразу перенаправляет их другой программе. Поэтому в конфигурационном файле /etc/news/incoming.conf, который управляет установлением соединений cinnd, указаны только локальный компьютер и серверы, выполняющие роль поставщиков новостей.

Атрибуты и их значения, задаваемые в файле incoming.conf, представлены в виде *ключ* : *значение*. Атрибуты могут быть объединены в *записи* (создаваемые с помощью ключевого слова peer); каждая запись описывает отдельный компьютер. (Глобальные атрибуты и их значения не включаются в записи.) Записи, в свою очередь, могут объединяться в *группы*. Для определения границ как записи, так и группы используются фигурные скобки. Пример файла incoming.conf для сервера, который получает материалы групп с одного сервера новостей, приведен в листинге 12.1.

Листинг 12.1. Пример файла **incoming.conf**

```
# Глобальные установки
streaming: true
max-connections: 50
# Allow NNTP posting from localhost
peer ME {
    hostname: "localhost, 127.0.0.1"
}
# Разрешение на передачу групп fiveroomco.com
peer fiveroom {
    hostname:news.fiveroomco.com
    patterns: *,!threeroomco.*
}
```

Наиболее важным является ключ hostname. Он задет имя узла, которому разрешено устанавливать соединение с данным сервером. Чтобы определить список групп новостей, которые могут быть переданы, используется ключ patterns; при указании имен групп используются те же соглашения, что и при формировании файла newsfeeds. По умолчанию сервер принимает все группы новостей, предлагаемые поставщиком. Другие ключи описаны в справочной системе, на страницах, посвященных **incoming.conf**.

Управление доступом клиентов

Возможно, вы захотите осуществлять проверку пользователей на право доступа к вашему серверу. Поскольку innd привлекает к решению этой задачи другую программу, необходимые установки выполняются в файле **/etc/news/nnrp.access**. Каждая строка этого файла состоит из пяти полей, разделенных двоеточиями:

имя_узла*:*полномочия*:*имя_пользователя*:*пароль*:*группы_новостей

Назначение каждого поля описано ниже.

- ***имя_узла***. В этом поле указывается имя или IP-адрес узла. В составе имени может использоваться символ *, определяющий групповую операцию. Так, например, выражение *.**threeroomco.**com описывает всех клиентов в пределах домена **threeroomco.**com. При указании IP-адреса можно задавать также маску подсети в виде **IP-адрес/маска** подсети, например 172.20.0.0/16.

- ***полномочия***. В этом поле может содержаться одна или несколько следующих опций: R (чтение сообщений разрешено), P (передача сообщений разрешена), N (клиент может использовать команду **NEWNEWS**) и L (клиент может передавать сообщения даже в ту группу, в которые не могут посылать статьи локальные пользователи). Последние две опции переопределяют глобальные установки для конкретного клиента.

- ***имя_пользователя***. Данное поле позволяет ограничить доступ к серверу отдельных пользователей. Если в поле содержится имя, то, перед тем, как доступ будет разрешен, выполняется аутентификация этого пользователя. Символ + указывает на то, что сервер должен предпринять попытку аутентификации с использованием базы паролей Linux. Следует заметить, что данная опция часто не работает, особенно

в тех случаях, когда в системе применяется "затененный" файл паролей. Если вы оставите это и следующее поле пустыми, аутентификация осуществляться не будет.

- *пароль.* Данное поле содержит пароль, требуемый для доступа к серверу.
- *группы_новостей.* Для указания групп новостей можно использовать шаблоны, подобные тем, которые применяются в файле newsgroups. Это поле позволяет ограничить доступ узла к конкретным группам. Если оставить данное поле пустым, это означает, что клиенту не доступна ни одна группа. Для того чтобы клиент мог обращаться ко всем группам, надо указать символ *.

Если в файле nnrp.access содержится несколько строк, последующие строки переопределяют предыдущие. Если вы собираетесь выполнить глобальные установки, а затем уточнить их для отдельных узлов, вы должны разместить более общие записи перед более конкретными.

Опции, управляющие удалением сообщений

Содержимое файла /etc/news/expire.ctl управляет автоматическим удалением сообщений. Записи в этом файле имеют тот же формат, что и записи других конфигурационных файлов INN. Каждая запись состоит из пяти полей:

шаблон:*флаг*:*хранение*:*время_по_умолчанию*:*удаление*

- *шаблон.* В данном поле помещается описание группы новостей. Подобно другим конфигурационным файлам, символ * определяет групповую операцию, т. е. выражение comp.os.* описывает всю категорию comp.os.
- *флаг.* В качестве флага используется символ, который указывает, что правило должно применяться только к модерируемым группам (M), только к немодерируемым группам (U) или ко всем группам (A).
- *хранение.* В составе сообщения может присутствовать поле заголовка Expires, в котором указывается срок действия этого сообщения. Значение, указанное в поле *хранение,* задает минимальный срок (число дней), в течение которого статья должна оставаться на сервере. Предположим, например, что в поле *хранение* содержится значение 6, а в поле заголовка Expires указано, что сообщение должно быть удалено через пять дней. В результате сообщение будет храниться в течение шести дней. Если бы при том же значении в поле *хранение,* в поле заголовка Expires был бы определен срок 7 дней, статья хранилась бы на сервере в течение семи дней. В рассматриваемом поле можно задавать не только целые числа, но и числа с плавающей точкой. Специальное значение never указывает на то, что срок действия статьи не ограничен. (Применяя значение never, будьте внимательны. Если оно будет указано для слишком большого количества статей, жесткий диск сервера быстро переполнится.)
- *время_по_умолчанию.* Данное поле чрезвычайно важно. В нем указывается срок действия для тех статей, в которых отсутствует поле заголовка Expires. Подобно полю *хранение,* значение в данном поле указывается в днях и может быть задано как целое число или число с плавающей точкой. Значение never указывает на то, что срок действия сообщения не ограничен.

- *удаление*. Если значение в поле хранения позволяет увеличить срок действия, указанный в поле заголовка Expires, то значение в поле *удаление* позволяет лишь уменьшить его. Например, если вы зададите в данном поле значение 10 и ваш сервер получит сообщение, поле Expires которого содержит значение 100, это сообщение будет удалено через десять дней. Подобно полям *хранение* и *время_по_умолчанию*, в поле *удаление* может быть задано число с плавающей точкой или специальное значение never.

Обеспечение выполнения сервера новостей

Сервер INN выполняется в режиме демона и запускается посредством сценариев SysV. Если вы инсталлировали сервер новостей с помощью пакета, поставляемого в комплекте с операционной системой, вы можете непосредственно вызывать соответствующий сценарий для запуска сервера.

Некоторые описанные выше задачи демон innd не решает. К таким задачам относится передача сообщений другим серверам или удаление статей по истечении срока их действия. Для выполнения требуемых действий используются утилиты и сценарии, вызываемые с помощью инструмента cron. Если сервер поставлялся в составе операционной системы, не исключено, что crontab-файлы, необходимые для автоматического вызова требуемых утилит уже сформированы и помещены в /etc/cron.d, /etc/cron.interval или другой каталог, используемый в подобных целях. Если вы хотите, чтобы соответствующие задачи выполнялись чаще или реже, вам надо найти crontab-файлы и изменить их.

Кроме того, в процессе сопровождения приходится изменять конфигурацию сервера. Например, вам может понадобиться добавить группу новостей, удалить существующую группу или временно запретить доступ к серверу. Выполнить подобные задачи поможет утилита ctlindd, поддерживающая большое количество опций сервера. Для просмотра возможностей данной программы задайте команду ctlindd -h.

Использование Leafnode

Сервер INN обычно используется крупными провайдерами или организациями, специализирующимися на предоставлении пользователям материалов групп новостей. Как было сказано выше в данной главе, в некоторых случаях необходимо иметь небольшой сервер новостей с ограниченными возможностями для ограниченного числа групп новостей. Кроме того, часто возникает необходимость организовать работу с группами новостей в рамках локальной сети при временном отсутствии связи с Internet. Сервер, решающий такую задачу, должен за время соединения скопировать материалы некоторых групп с внешнего сервера новостей и передать на этот сервер сообщения, отправленные в эти группы локальными пользователями. Такие сеансы взаимодействия обычно производятся один-два раза в день. Желательно выбрать для этого время, когда внешний сервер не слишком загружен. Необходимые для этого действия может выполнять сервер INN, но мощность данного инструмента несоизмерима с поставленной задачей. INN взаимодействует с другими серверами новостей как равноправный партнер, поэтому вам трудно будет найти поставщика новостей, который согласился бы планировать активность своего сервера в соответствии с графиком работы вашего сервера INN. Для решения поставленной задачи лучше использовать специальный сервер новостей с ограниченным

набором возможностей. На роль подобного сервера хорошо подходит продукт Leafnode (http://www.leafnode.org).

Leafnode — не единственный сервер, позволяющий организовать работу с группами новостей в автономном режиме. С такой задачей также хорошо справляются NNTPCache (http://www.nntpcache.org), Noffle (http://noffle.sourceforge.net), sn (http://infa.abo.fi/~patrik/sn/) и NewsCache (http://www.infosys.tuwien.ac.at/NewsCache/).

Возможности Leafnode

Подобно INN, функционирование продукта Leafnode обеспечивает несколько взаимодействующих между собой программ. Наиболее важные из них перечислены ниже.

- leafnode. Программа, реализующая сервер NNTP. Она запускается посредством суперсервера и обеспечивает взаимодействие с программой просмотра новостей, выполняющейся на том же или на другом компьютере.
- fetchnews. Программа, отвечающая за получение групп новостей с внешнего сервера. Для того чтобы периодически получать требуемые материалы, вы можете запускать данную программу с помощью инструмента cron. При необходимости fetchnews может быть вызвана вручную или из сценария, устанавливающего PPP-соединение.
- texpire. Подобно другим серверам новостей, Leafnode хранит сообщения групп в подкаталогах каталога /var/spool/news. Чтобы диск не переполнялся, приходится периодически удалять старые сообщения. Эту задачу решает программа texpire. Обычно она периодически запускается с помощью cron.
- newsq. Данная программа отображает сведения о сообщениях, отправленных в группы локальными пользователями, но еще не переданных на внешний сервер.

Leafnode осуществляет динамическую загрузку материалов групп. Если пользователь предпринимает попытку работы с группой, то при следующем запуске fetchnews материалы этой группы копируются с внешнего сервера. Если Leafnode обнаруживает, что в течение определенного периода времени ни один из пользователей не работал с этой группой, копирование материалов прекращается. Таким образом, у пользователей создается впечатление, что локальный сервер новостей поддерживает все группы, хотя на самом деле эту работу выполняет внешний сервер. Подобный подход к доставке данных приводит к тому, что при работе с новой группой сообщения поступают к пользователю с некоторой задержкой.

Одна из особенностей программы Leafnode состоит в том, что при ее использовании не требуется настройка для работы с источником новостей. Leafnode (если быть точным, программа fetchnews) взаимодействует с внешним сервером как обычная программа просмотра новостей. Благодаря этому можно организовать работу Leafnode с любым сервером новостей, например, сервером, расположенным на компьютере провайдера.

С одной стороны, Leafnode потребляет больше ресурсов, чем обычная программа просмотра новостей, но, с другой стороны, она экономит ресурсы. Получение материалов обычной группы не занимает много времени и может осуществляться в течение одного сеанса связи с провайдером по протоколу PPP. Чтобы пользователь мог работать с группой, обычная программа просмотра новостей должна поддерживать постоянное соединение с внешним сервером. Если Internet-услуги предоставляются на условиях поминутной оплаты, Leafnode, безусловно, экономит ресурсы. В то же время Leafnode копирует все сообщения группы, даже те, которыми не заинтересуется ни один локальный пользователь. Но если с Leafnode работают несколько пользователей, которые интересуются приблизительно одинаковым набором материалов групп, общий объем переданной информации получается меньше, чем если бы каждая клиентская программа непосредственно взаимодействовала с внешним сервером.

В начале 2002 г. последней версией Leafnode была версия 1.9.19. Следующая версия (2.0) в это время находилась в стадии разработки. В версии 2.0 планируется реализовать поддержку локальных групп новостей. В версии 1.9.x такая возможность отсутствует.

Важно помнить, что продукт Leafnode разрабатывался для небольших узлов. Масштабируемость данного пакета ограничена, поэтому при обслуживании большого количества пользователей и поддержке большого числа групп новостей производительность Leafnode резко снижается. Наилучшим образом данный пакет обслуживает 20-25 пользователей. Если большая нагрузка приводит к снижению эффективности работы Leafnode, целесообразно рассмотреть возможность перехода к использованию INN или другого полнофункционального сервера новостей.

Еще одна проблема, возникающая при работе с Leafnode, состоит в том, что данный продукт игнорирует ошибки в сообщениях. В результате в группе могут появиться статьи, которые никогда не посылали в Usenet, а существующие сообщения могут быть доставлены пользователям в искаженном виде. Специальная настройка fetchnews позволят частично решить данную проблему, но при этом, если внешний сервер работает ненадежно, некоторые сообщения будут утеряны.

Настройка Leafnode

Настройка пакета Leafnode сводится к настройке трех программ: leafnode, fetchnews и texpire. Опции, управляющие работой всех трех программ, содержатся в одном конфигурационном файле, но это не мешает настраивать каждую программу независимо от других. Если пакет Leafnode поставлялся в составе системы Linux, вам придется внести в конфигурационный файл лишь незначительные изменения.

Общие установки

Основной конфигурационный файл Leafnode называется *config;* обычно он хранится в файле `/etc/leafnode`. Строки данного файла, содержащие комментарии, начинаются с символа #. Помимо комментариев в конфигурационном файле находятся записи, представленные в следующем виде:

```
опция - значение
```

Минимальная конфигурация Leafnode предполагает наличие лишь двух опций: `server` и `expire`. Остальные опции необязательны; настраивая Leafnode, вы можете принять для

них значения, заданные по умолчанию. Наиболее важные опции, присутствующие в файле config, описаны ниже.

- server. Данная опция задает имя внешнего сервера, предоставляющего материалы групп, например server = news.abigisp.net. Задавая несколько опций server, вы можете организовать получение материалов групп с нескольких серверов.
- expire. Эта опция указывает время (количество дней), по истечении которого сообщения будут удалены.
- username. Если внешний сервер требует указывать имя пользователя, его можно задать посредством данной опции.
- password. Если внешний сервер требует ввода пароля, эта опция позволяет задать его.

ВНИМАНИЕ Следует заметить, что пароль хранится в незашифрованном виде. По умолчанию файл config доступен только пользователю root, поэтому опасность того, что пароль будет похищен и использован для незаконного доступа к серверу, относительно невелика. Однако по сети этот пароль также передается в незакодированном виде, поэтому его не следует использовать для других целей.

- port. Большинство серверов новостей использует по умолчанию порт 119. Данная опция позволяет вам указать другой порт.
- nodesc. Как правило, серверы новостей предоставляют описания групп, однако некоторые серверы не обеспечивают такой возможности. Наилучшим образом Leafnode работает в том случае, если в конфигурационном файле указана опция nodesc = 1.
- timeout. При соединении с сервером новостей программа fetchnews обычно выжидает десять секунд, а затем прекращает попытки. Данная опция позволяет изменить значение тайм-аута.
- groupexpire *имя_группы*. Если вы хотите задать для разных групп различное время хранения сообщений, вы можете воспользоваться этой опцией. При указании имени группы можно использовать символ групповой операции. Например, все группы категории comp.os.linux задаются с помощью значения comp.os.linux.*.
- maxfetch. С помощью данной опции Leafnode позволяет ограничить число новых сообщений, копируемых с внешнего сервера. Не следует задавать слишком малое значение опции maxfetch, так как в этом случае Leafnode не сможет скопировать все сообщения группы. Старые сообщения будут вытесняться новыми, и в результате пользователь не получит их,
- initialfetch. Когда пользователь начинает работать с новой группой, копирование всех ее материалов может занять много времени. Опция initialfetch позволяет ограничить число сообщений новой группы, которые могут быть скопированы с внешнего сервера.

- delaybody. По умолчанию Leafnode копирует с внешнего сервера как заголовки, так и тело сообщений. Число сообщений может быть ограничено с помощью maxfetch и других опций. Leafnode может также работать и в другом режиме — копировать только заголовки сообщений. В этом случае тело сообщения будет скопировано лишь в том случае, если пользователь активизирует соответствующий заголовок в программе просмотра. После щелчка на заголовке сообщения оно помечается для копирования, и тело сообщения доставляется при следующем сеансе получения данных. Если вы зададите значение 1 опции delaybody, пользователь будет получать тело выбранного сообщения с задержкой, но при этом уменьшится внешний трафик.

- maxcrosspost. Данная опция предназначена для борьбы со спамом. Если одно сообщение направлено в несколько групп, причем число групп превышает значение опции maxcrosspost, это сообщение удаляется. По умолчанию количество групп, в которые может быть передано одно и то же сообщение, не ограничено.

- maxage. Если сервер новостей сконфигурирован неправильно, получаемые сообщения будут снова отправляться в Usenet, увеличивая тем самым трафик в сети. Данная опция указывает Leafnode на то, что сообщения, с момента создания которых прошло время больше указанного, должны игнорироваться. По умолчанию опция maxage не используется.

- maxlines. Если в конфигурационном файле задана опция maxlines, сообщения, содержащие большее число строк, чем указано в качестве значения данной опции, должны игнорироваться. По умолчанию это ограничение не используется.

- minlines. Если в конфигурационном файле задана опцияminlines, сообщения, содержащие меньшее число строк, чем указано в качестве значения данной опции, должны игнорироваться. По умолчанию эта опция не используется.

- maxbytes. С помощью данной опции вы можете запретить копирование сообщений, размер которых в байтах превышает значение данной опции. По умолчанию данная опция не используется.

- timeout_short. По умолчанию Leafnode продолжает получать сообщения в течение двух дней после единичного обращения к группе. Данный параметр позволяет переопределить значение по умолчанию.

- timeout_long. По умолчанию Leafnode продолжает получать сообщения в течение семи дней после окончания работы с группой. Данный параметр позволяет изменить значение по умолчанию.

- timeout_active. Leafnode периодически обновляет список групп, предоставляемых внешним сервером. Данный параметр указывает на то, как часто должно проводиться такое обновление. По умолчанию список обновляется каждые 90 дней.

- filterfile. Значением данной опции является путь к файлу, выполняющему фильтрацию. (Вопросы фильтрации сообщений будут рассмотрены ниже в этой главе.) По умолчанию фильтрация не производится.

- hostname. Некоторые программы просмотра новостей не устанавливают идентификатор создаваемых сообщений. В этом случае идентификатор устанавливает Leafnode, задавая имя компьютера, на котором выполняется данный пакет. Если вы хотите, чтобы в состав сообщений включалось другое имя, вы должны указать его с помощью данной опции.

Перечисленные выше опции имеют отношение ко всем трем основным программам Leafnode: `leafnode`, `fetchnews` и `texpire`. Несмотря на то что все три программы используют один и тот же конфигурационный файл, они запускаются по-разному.

Запуск программы leafnode

Как было сказано ранее, сервер `leafnode` запускается с помощью суперсервера `inetd` или `xinetd`. Ниже приведена соответствующая запись в конфигурационном файле `inetd.conf`.

```
nntp stream tcp nowait news /usr/sbin/tcpd /usr/sbin/leafnode
```

В дистрибутивных пакетах, в которых используется суперсервер `xinetd`, обычно уже содержится файл, необходимый для запуска `leafnode`; он помещается в каталог `/etc/xinetd.d`. Независимо от того, используете ли вы `inetd` или `xinetd`, для того, чтобы сервер Leafnode смог начать обслуживание клиентов, вам надо перезапустить суперсервер. После того как вы сделаете это, программа `leafnode` будет отвечать на запросы клиентов так же, как INN или другой полнофункциональный сервер новостей.

ВНИМАНИЕ! В конфигурационном файле Leafnode не предусмотрены опции контроля доступа. Для того чтобы управлять взаимодействием с компьютерами локальной сети и внешними узлами, вы можете установить соответствующую конфигурацию TCP Wrappers.

Получение материалов групп

При каждом запуске программы `fetchnews` материалы групп копируются с внешнего сервера; эта же программа отвечает за передачу на сервер сообщений, составленных вашими пользователями. (Для получения информации о сообщениях, ожидающих обработки, надо запустить программу `newsq`.) Чтобы это происходило, необходимо указать имя внешнего сервера в файле `/etc/leafnode/config`. Вероятнее всего, что при первом запуске `fetchnews` ее выполнение займет достаточно длительное время, так как программа должна скопировать с внешнего сервера список предоставляемых групп новостей.

При вызове программы `fetchnews` можно задавать описанные ниже опции.

- -v. Данная опция позволяет управлять выводом информации в процессе выполнения программы. Чем больше символов v вы укажете при вызове программы, тем подробнее она будет комментировать выполняемые ею действия. Максимальный объем информации выводится в том случае, когда указаны четыре символа v (`-vvvv`). Эта опция может использоваться в качестве инструмента диагностики в тех случаях, когда программа `fetchnews` выполняется не так, как вы того ожидаете.

- -x *число*. Если вы встретились с проблемами при копировании материалов групп, вызов программы с указанием данной опции позволит скопировать сообщения с предшествующими номерами.

- -l. Как было сказано ранее, Leafnode позволяет получать материалы групп с различных серверов. Данная опция указывает на то, что данные должны быть скопированы только с первого сервера.

- -п. Данная опция сообщает о том, что сообщения групп, с которыми пользователи перестали работать, должны по-прежнему копироваться с сервера.

- `-f`. Если вы считаете, что список групп, предоставляемых внешним сервером, устарел, вы можете задать с помощью данной опции обновление списка. (По умолчанию программа автоматически копирует с сервера новый список групп один раз в 90 дней.) Для выполнения этой операции может потребоваться достаточно длительное время.

- `-P`. Данная опция указывает программе fetchnews нато, что сообщения, составленные локальными пользователями, должны быть переданы на внешний сервер, но копировать с сервера материалы групп не следует.

СОВЕТ ✔ В обычных условиях, для того, чтобы пользователь увидел в составе группы переданное им сообщение, необходимо дважды вызвать программу fetchnews. Чтобы новые сообщения стали доступны после очередного выполнения fetchnews, надо предварительно вызвать fetchnews с опцией `-P`. В Leafhode 2.0 задержка при получении собственных сообщений не возникает, поэтому предварительный вызов `fetchnews`не требуется.

Принимая меры для организации работы Leafnode, необходимо решить, каким способом следует вызывать программу `fetchnews`. Вы можете задать периодическое выполнение данной программы посредством инструмента `cron` либо включить вызов `fetchnews` в состав сценария, посредством которого устанавливается PPP-соединение (примером такого сценария является `ppp-on-dialer`, рассмотренный в главе 2). Вызов `fetchnews` посредством `cron` имеет смысл, если у вас есть постоянное соединение с Internet либо если вы хотите автоматически устанавливать соединение с Internet и получать данные с внешнего сервера новостей в то время, когда этот сервер наименее загружен, например рано утром. Ответить на вопрос о том, насколько часто следует вызывать программу `fetchnews`, можно лишь, зная потребности ваших пользователей и возможности внешнего сервера по предоставлению данных. Вызывая `fetchnews` посредством сценария установки PPP-соединения, вы предоставите вашим пользователям наиболее новые сообщения (насколько это позволяет график установления соединений с Internet).

Удаление старых сообщений

Программа `texpire` анализирует сообщения, хранящиеся на компьютере, и удаляет те из них, которые в соответствии с установками в файле `/etc/leafnode/config` считаются устаревшими. Удаление старых сообщений должно проводиться регулярно, иначе жесткий диск компьютера переполнится. Как правило, программа `texpire` вызывается с помощью инструмента `cron`. В некоторых пакетах Leafnode предусмотрен специальный сценарий, который помещается в `/etc/cron.daily` или другой подобный каталог.

Если такого сценария нет, вам надо создать его самостоятельно или спланировать вызовы texpire с помощью утилиты crontab.

Принимая решение об удалении сообщений, программа texpire учитывает данные о потоках. (Потоком называется исходное сообщение и все ответы на него.) Сообщение удаляется только в том случае, если в течение времени, превышающего срок действия сообщения, не было обращений к потоку. Если кто-либо из пользователей недавно просматривал содержимое потока, к которому принадлежит сообщение, оно может храниться на компьютере дольше, чем это предусмотрено в конфигурационном файле.

Подобно fetchnews, при вызове texpire может быть указано от одной до четырех опций -v. Среди других опций следует особо отметить опцию -f. В обычных условиях, чтобы убедиться в том, что данные потока не просматривались, texpire анализирует время последнего обращения к файлам. Опция -f сообщает texpire о том, что эту информацию следует игнорировать. Дело в том, что многие программы-архиваторы, в частности tar, изменяют дату последнего доступа к архивируемым файлам. Если вы часто архивируете материалы групп новостей, создается неверное впечатление о том, что сообщения недавно просматривались. Избежать этого позволяет опция -f.

Фильтрация сообщений

Leafnode позволяет удалять сообщения, соответствующие определенным критериям. Решение об удалении принимается исходя из информации, содержащейся в заголовке сообщения. Предположим, например, что в статьях, получаемых от пользователя obnoxious@annoying.edu, постоянно встречаются высказывания, оскорбляющие ваших пользователей. Указанное имя присутствует в заголовке в качестве значения поля From. На основе этой информации Leafnode может "отфильтровать" сообщения данного пользователя. Для этого вам надо включить соответствующее выражение в файл /etc/leafnode/filters, содержащий правила фильтрации. Правила в файле /etc/leafnode/filters имеют вид регулярных выражений. Например, если вы хотите удалять сообщения, поступающие от пользователя obnoxious@annoying.edu, необходимое для этого выражение будет иметь следующий вид:

```
^ From:.*obnoxious@annoying\.edu
```

Данное выражение начинается с символа ^, за которым следует имя заголовка (в данном случае From:). Символы .*, используемые совместно, означают любое число произвольных символов. Строка obnoxious@annoying.edu указывается непосредственно, но так как точка имеет в языке регулярных выражений специальное значение, перед ней указывается обратная косая черта (\).

Более подробно регулярные выражения будут рассмотрены в главе 19.

Для фильтрации сообщений вам надо указать Leafnode расположение файлов фильтров. Для этого можно использовать опцию filterfile в файле /etc/leafnode/config, о которой шла речь ранее в данной главе. Несмотря на то что фильтры обычно располагаются в каталоге /etc/leafnode/filters, вы можете указать любое имя файла и любой путь к нему.

Резюме

Серверы новостей потребляют значительные ресурсы. Для поддержки групп новостей Usenet необходимо ежедневно передавать большой объем данных, кроме того, эти данные приходится хранить в течение нескольких дней. Чтобы инсталлировать полнофункциональный сервер новостей, необходим отдельный компьютер с дисковым пространством в десятки и даже сотни гигабайт. Если ваши потребности более скромны, вы можете установить сервер новостей, не реализуя взаимодействие его с Usenet. С помощью такого сервера можно обеспечить работу нескольких локальных групп новостей и даже групп, доступных для всего мира. Как для поддержки Usenet в полном объеме, так и для организации локальных групп используется сервер INN, который может выполняться в среде Linux. Функционирование этого сервера обеспечивает несколько взаимодействующих друг с другом программ; для их настройки используется несколько конфигурационных файлов. С помощью этих файлов вы можете описывать группы новостей, которые должны поддерживаться на сервере, задавать политику взаимодействия с другими серверами и клиентами, а также определять прочие характеристики сервера.

Если число пользователей, которым необходим доступ к материалам групп новостей, невелико и если в круг их интересов попадает лишь небольшое число групп, с задачей обслуживания этих пользователей может справиться сервер с ограниченными функциональными возможностями. В качестве такого сервера можно использовать пакет Leafnode. Он копирует с внешнего сервера лишь материалы тех групп, которые интересуют локальных пользователей, и применяет для передачи сообщений подмножество протокола NNTP. Такой сервер хорошо подходит для небольших офисов и даже для домашнего использования. Leafnode и другие подобные продукты обладают важным преимуществом: они могут работать при отсутствии постоянного соединения с Internet. Во время сеанса связи с провайдером они копируют с внешнего сервера требуемые материалы и предоставляют пользователям возможность работать с группами новостей в автономном режиме.

Глава 13

Удаленная регистрация на сервере

Многие серверы предоставляют доступ лишь к отдельным ресурсам компьютера, на котором они выполняются. Например, временной сервер, рассмотренный *в* главе 10, сообщает о показаниях системных часов, а сервер шрифтов, который будет обсуждаться в главе 15, передает клиентам битовые карты символов того или иного шрифта. Существуют, однако, серверы, которые предоставляют более или менее полный доступ к ресурсам компьютера и называются *серверами удаленной регистрации* (remote login server). Такие серверы позволяют пользователям регистрироваться в системе и запускать в ней различные программы так же, как это происходит при работе за консольным терминалом. Сервер удаленной регистрации организует работу нескольких пользователей на одном компьютере.

Существует несколько типов серверов удаленной регистрации. Выбор типа сервера зависит от конкретной ситуации. В данной главе обсуждаются серверы, которые обеспечивают работу с системой в текстовом режиме. При этом пользователь может запускать программы с алфавитно-цифровым интерфейсом, например, почтовые клиенты `pine` и mutt, компилятор дсс, текстовые редакторы Vi и Emacs и т. д. Рассматриваемые здесь серверы не позволяют выполнять программы, предназначенные для работы в среде X Window, например KMail или Nedit; для этой цели используются средства доступа, поддерживающие графический интерфейс, которые будут рассматриваться в главе 14.

Данная глава посвящена рассмотрению трех инструментов, предназначенных для поддержки удаленной регистрации: `rlogind`, Telnet и SSH. Каждый из них обладает своими уникальными характеристиками и используется для решения конкретных типов задач. Различия между серверами удаленной регистрации в основном связаны с вопросами защиты и наличием специальных возможностей. Например, среди рассматриваемых в данной главе инструментов `rlogind` реализует минимальный уровень защиты, а SSH обеспечивает наибольшую степень безопасности. Следует заметить, что керберизованные версии `rlogind` и Telnet сравнимы по степени защиты с SSH. (Система Kerberos рассматривалась в главе 6.)

Использование сервера удаленной регистрации

Основное назначение сервера удаленной регистрации состоит в том, чтобы предоставить пользователям возможность запускать произвольные текстовые программы на других компьютерах. В зависимости от конфигурации аппаратных средств и специфики выполняемых программ на компьютере под управлением Linux могут одновременно работать до тысячи удаленных пользователей. Компьютер среднего уровня способен реально обслуживать несколько десятков пользователей, при условии, что программы, которые они запускают, не потребляют слишком много ресурсов.

ВНИМАНИЕ Серверы удаленной регистрации более критичны с точки зрения безопасности по сравнению с другими типами серверов. Предположим, например, что злоумышленник сумел получить пароль, позволяющий пользоваться услугами сервера POP. Если считать, что сервер POP не имеет недостатков в системе защиты, неавторизованный пользователь не сможет причинить существенного вреда компьютеру. Если злоумышленнику удастся зарегистрироваться на сервере удаленного доступа, в его распоряжении окажутся тысячи утилит и приложений. В защите многих программ существуют недостатки, воспользовавшись которыми можно получить доступ к ресурсам компьютера с привилегиями системного администратора. Поэтому необходимо внимательно относиться к настройке сервера удаленного доступа и принимать меры к обеспечению секретности паролей. Если на компьютере не предполагается работа удаленных пользователей, на нем следует запретить выполнение всех серверов регистрации.

Настройка rlogind

Сервер `rlogind` — одна из нескольких программ, поддерживающих так называемые *r-команды*. Эти команды были реализованы для обеспечения различных типов удаленного доступа к системе UNIX. При запуске клиента `rlogin` он старается установить соединение с сервером `rlogind` или `in.rlogind`. Одно из преимуществ `rlogind` состоит в том, что настройка данного сервера осуществляется очень просто. Однако используемый протокол чрезвычайно примитивен, поэтому контролировать обращения к системе посредством `rlogind` очень трудно.

Запуск rlogind

Сервер `rlogind` обычно запускается посредством суперсервера. Во многих дистрибутивных пакетах в конфигурационном файле `/etc/inetd.conf` уже присутствует запись для `rlogind`, но чаще всего она закомментирована. Для того, чтобы сервер выполнялся на компьютере, вам надо убрать символ комментариев из соответствующей строки и перезапустить суперсервер. Если в системе используется `xinetd`, в ней обычно создается файл для запуска `rlogind`, который помещается в каталог `/etc/xinetd.d`. Файл запуска, как правило, настраивается так, чтобы по умолчанию выполнение сервера было запрещено. Поэтому, чтобы обеспечить работу `rlogind`, надо специально разрешить его запуск. Необходимые для этого действия описаны в главе 4.

На выполнение `rlogind` влияют перечисленные ниже опции.

- -п. В обычных условиях rlogind периодически проверяет наличие клиента, даже если он длительное время не передает данные. Опция -п отменяет такое поведение сервера.
- -a. Данная опция была введена для поддержки расширенной процедуры аутентификации, но во многих системах она не работает.
- -h. В обычных условиях rlogind не использует файл .rhosts суперпользователя. Опция -h указывает на то, что данный файл должен использоваться.
- -1. Данная опция запрещает использование файла .rhosts для аутентификации пользователей. Исключение составляет суперпользователь, взаимодействие с которым определяет опция -h.
- -L. Эта опция запрещает аутентификацию на основе данных, содержащихся в файлах .rhosts и hosts.equiv.

ВНИМАНИЕ Несмотря на то что опции -h, -1 и -L входят в официальный набор опций rlogind, в новых версиях Linux они обычно не оказывают влияния на работу сервера. Причина в том, что в новых версиях системы используются модули PAM; в результате действие указанных опций отменяется.

Средства защиты rlogind

Средства защиты всех утилит, реализующих r-команды, в лучшем случае могут считаться устаревшими. А если подходить к этому вопросу с позиции современных требований, следует признать, что защита в них вовсе отсутствует. В частности, работа сервера rlogind основана на принципе доверия, а это значит, что rlogind полагается на результаты процедуры аутентификации, выполненной на клиентской машине. Существуют способы несколько улучшить защиту rlogind, а использование системы Kerberos позволяет реально обезопасить ресурсы сети.

Рассмотрим принцип работы базовых средств защиты rlogind. Когда клиентская программа предпринимает попытки установить соединение с сервером rlogind, система выполняет для аутентификации пользователя следующие действия.

1. Сервер проверяет порт клиента, выступающего инициатором установления соединения. Обычно клиент-программы rlogin используют номер порта в диапазоне 512–1023. Если номер порта клиента лежит за пределами этого диапазона, rlogind отвергает попытки установить соединение. Такая мера предотвращает использование для взаимодействия подложного клиента rlogin, написанного обычным пользователем, поскольку номера портов ниже 1024 может использовать только root. Однако это не мешает злоумышленнику подключить к сети свой компьютер под управлением Linux и зарегистрироваться на нем как root. Кроме того, в некоторых операционных системах порты с номерами ниже 1024 доступны всем пользователям, поэтому описанная здесь мера защиты не очень эффективна.
2. Сервер удаленной регистрации обращается к серверу DNS, чтобы преобразовать IP-адрес клиента в доменное имя.

3. Если имя, полученное в результате DNS-преобразования, принадлежит тому же домену, что и сервер, или если при запуске rlogind была указана опция -a, сервер ищет IP-адрес по доменному имени. Если полученный в результате адрес не отличается от исходного IP-адреса и если опции -L и -l не были указаны, rlogind обращается к файлам ~/.rhosts и /etc/hosts.equiv и проверяет, объявлен ли данный клиент как пользующийся доверием. Если проверка дала положительный результат и если удаленный пользователь имеет учетную запись на сервере, rlogind осуществляет регистрацию без дальнейшей проверки.

4. Если IP-адрес, полученный в результате DNS-преобразования, и IP-адрес, указанный в запросе, не совпадают, либо если была задана опция -L или -l, либо если клиент не найден в списке клиентов, пользующихся доверием, программа rlogind запрашивает пользовательское имя и пароль. Если пользователь ввел корректный пароль, rlogind предоставляет доступ в систему. Ели пароль не совпадает с паролем, хранящимся в базе данных, пользовательское имя и пароль запрашиваются снова. Если пользователь не смог зарегистрироваться с нескольких попыток, соединение разрывается.

При выполнении описанной выше процедуры регистрации предполагается, что программа rlogind знает имя пользователя, по инициативе которого устанавливается соединение. Эта информация передается с одного компьютера на другой и скрыта от пользователя. При желании, вызывая клиент-программу rlogin, можно задать имя пользователя с помощью опции -l, например rlogin -l sjones.

Поскольку rlogind использует принцип доверия и пользователи на узлах, пользующихся доверием, могут самостоятельно изменять данные в файле .rhosts, защиту сервера rlogind можно обойти несколькими способами. Это можно сделать, воспользовавшись недостатками в защите клиента, включив в запрос фальшивый IP-адрес компьютера, пользующегося доверием, удалив клиентскую машину из сети и подключив к ней свой компьютер, либо включив в файл .rhosts нужную запись. Некоторые из указанных способов атаки осуществить достаточно сложно. Например, если администратор постоянно следит за состоянием сети, пользователю вряд ли удастся незаметно подключить к ней свой компьютер но само разнообразие способов делает сервер rlogind чрезвычайно уязвимым для злоумышленника. При всех своих недостатках сервер rlogind имеет одно неоспоримое преимущество: регистрация на сервере может осуществляться без указания пользовательского имени и пароля, поэтому соединение с удаленным узлом устанавливается достаточно быстро.

```
[rodsmith@nessus rodsmith]$ rlogin speaker
Last login: Mon Aug 12
14:48:58 2002 from nessus on 4
[rodsmith@speaker rodsmith]$
```

Еще одна особенность rlogind состоит в том, что при взаимодействии с этим сервером данные передаются по сети в незашифрованном виде, следовательно, их можно легко перехватить. Поэтому средства rlogind целесообразно применять лишь в небольшой внутренней сети, пользователям которой вы полностью доверяете.

Таким образом, если вам необходимо, чтобы соединение с сервером удаленной регистрации устанавливалось быстро и чтобы для установления соединений можно было использовать сценарий, имеет смысл рассмотреть целесообразность применения rlogind.

При этом следует учитывать, что защита данного сервера крайне несовершенна и ее легко обойти. В отличие от rlogind, Telnet практически во всех случаях требует указывать пароль, а средства SSH обеспечивают гораздо более высокую степень защиты.

Управление доступом к rlogind

Если вы обращаетесь к rlogind с узла, не относящегося к списку узлов, пользующихся доверием, вам придется ввести пользовательское имя и пароль. Однако не исключено, что вы захотите регистрироваться на удаленном узле, не задавая имя и пароль. Для этого вам придется определить компьютер, на котором вы работаете, как узел, пользующийся доверием. Это можно сделать двумя способами.

- Создать запись в файле **/etc/hosts.equiv.** Данный конфигурационный файл содержит установки для всей системы. Если компьютер указан в данном файле, любой пользователь, работающий на нем, может обращаться к службам, поддерживающим **r-команды.** Для того чтобы такие обращения поддерживались, необходимо, чтобы на сервере существовала учетная запись для текущего пользователя либо выполнялось отображение пользовательских имен. Если же соответствие имен установить не удалось (например, если пользователь julia хочет зарегистрироваться на удаленном сервере с помощью учетной записи fred), придется вводить пароль.
- Создать запись в файле **~/.rhosts.** Этот файл хранится в рабочем каталоге пользователя и содержит описания клиентов, которые пользуются доверием у этого пользователя. Если имя удаленного пользователя совпадает с именем пользователя, для которого на сервере существует учетная запись, удаленный пользователь получает доступ к ресурсам сервера. Кроме того, можно принять меры для отображения пользователей (средства отображения будут рассмотрены ниже). Если на сервере используется файл .rhosts, за его поддержку отвечает пользователь, для которого создан этот файл.

ВНИМАНИЕ Тот факт, что файл ~/.rhosts доступен для пользователя, означает, что вы как системный администратор делегируете вашим пользователям право настраивать средства защиты системы. Это одна из причин, по которым применять сервер rlogind не рекомендуется. Если же по каким-либо причинам вам приходится запускать на компьютере rlogind, используйте TCP Wrappers либо другие средства ограничения доступа.

Оба описанных выше файла управляют использованием всех **r-команд** на сервере, в частности rlogin, rcp и rsh. Если на компьютере поддерживается система печати BSD LPD (системы печати были рассмотрены в главе 9), эти файлы также осуществляют контроль доступа к принтерам.

В обоих файлах содержатся записи, представленные в одинаковом формате, однако некоторые элементы интерпретируют по-разному. Каждая запись занимает одну строку и описывает узел или группу узлов. Формат записи приведен ниже.

```
[+|-][имя_узла] [имя_пользователя]
```

Символ + или **-,** указанный перед именем узла, разрешает или запрещает доступ для конкретного клиента. По умолчанию предполагается, что доступ разрешен, поэтому в большинстве случаев символ + указывать не обязательно. Символ - запрещает доступ

клиента. Запрещающую запись имеет смысл использовать в том случае, если ей предшествует запись, которая разрешает доступ к серверу для группы клиентов.

ВНИМАНИЕ | Используя символ +, будьте внимательны. Если в строке указан только этот символ (а имя узла отсутствует), доступ разрешается для всех клиентов. Подобная политика защиты является одним из недостатков r-команд. Если вы по ошибке введете пробел между символом + и именем узла, система будет интерпретировать имя узла как имя пользователя и предоставит возможность обращаться к серверу со всех компьютеров.

Для идентификации узла может использоваться IP-адрес (например, 192.168.34.56) или имя (например, gingko.threeroomco.com). В качестве имени может быть указано полное доменное имя, а если и сервер, и клиентская машина принадлежат одному домену, достаточно указать лишь имя самого компьютера, например gingko. Если перед именем задан символ @, это имя определяет домен NIS (для работы с NIS ваша система должна быть специальным образом сконфигурирована).

Если вы включите в состав записи имя пользователя, то указанному пользователю будет предоставлен доступ к системе. Запись в файле .rhosts, содержащая пользовательское имя, означает, что этот пользователь эквивалентен пользователю, в рабочем каталоге которого находится файл .rhosts. Предположим, например, что в файле .rhosts, находящемся в рабочем каталоге пользователя julia, содержится следующая запись:

```
172.21.13.14 jbrown
```

В этом случае пользователь jbrown, который работает на узле с адресом 172.21.13.14, может регистрироваться на сервере под именем julia и получить при этом все полномочия данного пользователя. (Другими словами, работая на клиентском компьютере, jbrown может вызывать команду rlogin, указывая опцию -l julia.)

Записи в файле /etc/hosts.equiv распространяются на всю систему. Если в этом файле указано имя пользователя, это означает, что он имеет право регистрироваться на сервере с помощью любой учетной записи, за исключением root. Если бы запись, рассмотренная ранее в качестве примера, присутствовала в файле /etc/hosts.equiv, это означало бы, что пользователь jbrown, работающий на компьютере с адресом 172.21.13.14, имеет право регистрироваться не только под именем julia, но и под именем любого другого пользователя, кроме root. Таким образом, указывая имя пользователя в файле /etc/hosts.equiv, вы создаете угрозу безопасности системы. Исключением являются случаи, когда пользовательское имя указывается в запрещающих записях, которые начинаются с символа -.

Доступ к rlogind может ограничиваться не только с помощью записей в файлах ~/.rhosts и /etc/hosts.equiv и проверки имени пользователя и пароля. Существуют также другие механизмы, предназначенные для ограничения доступа. Поскольку сервер rlogind запускается с помощью inetd или xinetd, вы можете применять для этой цели TCPWrappers. Блокировать доступ можно также с помощью брандмауэра, указав при его настройке TCP-порт 513 (порт, используемый программой rlogind).

Настройка Telnet

Протокол Telnet часто используется для удаленной регистрации в сети Internet. Клиент-программа Telnet (как правило, она называется telnet) поставляется с большинством

версий Linux. Программы, реализующие Telnet-серверы, также широко распространены, однако они в основном включаются в состав тех операционных систем, которые обеспечивают одновременную работу нескольких пользователей, например в Linux, UNIX и VMS. Несмотря на то что протокол Telnet сложнее протокола, используемого rlogind, он, как и все протоколы семейства TCP/IP, достаточно прост. Выполняя настройку сервера Telnet для работы в системе Linux, необходимо обеспечить запуск соответствующей программы посредством суперсервера. В отдельных случаях приходится предусматривать запуск сервера Telnet с помощью сценария SysV. Настройка Telnet также предполагает дополнительные операции, например, создание сообщения, выводимого в качестве приглашения к регистрации.

С точки зрения современных стандартов защита Telnet очень слабая, но она все же лучше, чем соответствующие средства программы rlogind. Зная слабые стороны системы защиты Telnet, вы сможете принять обоснованное решение о том, в каких случаях применение Telnet оправдано, а в каких лучше использовать другие средства удаленной регистрации. Кроме того, вы сможете принять меры, минимизирующие опасность для системы при работе с сервером Telnet. Если в вашей сети установлены средства Kerberos, вы можете повысить уровень безопасности, применяя керберизованные версии клиента и сервера Telnet.

Опции, используемые при запуске сервера Telnet

В некоторых случаях сервер Telnet инсталлируется по умолчанию при установке операционной системы, но иногда приходится инсталлировать этот сервер самостоятельно. Пакет, содержащий сервер Telnet, в различных системах называется по-разному. Например, в Caldera он называется netkit-telnet, в Debian — telnetd, в Mandrake и Red Hat — telnet-server, в Slackware — tcpipl, в SuSE — nkitserv а в TurboLinux — telnet. Некоторые из этих пакетов, например telnetd, поставляемый в составе системы Debian, содержат только сервер Telnet, а в других, например в telnet системы TurboLinux, находится как серверная, так и клиентская программа. В большинстве случаев сервер Telnet устанавливается по умолчанию, но это не означает, что в системе разрешено выполнение данного сервера. Вопросы запуска серверов с помощью суперсервера были рассмотрены в главе 4 (программа, реализующая сервер Telnet, обычно носит имя telnetd или in.telnetd).

При запуске сервера Telnet могут быть указаны опции, определяющие особенности его функционирования. Некоторые из них предназначены для управления дополнительными средствами защиты, которые в большинстве стандартных версий Telnet отсутствуют. Опции, используемые наиболее часто, перечислены ниже.

- -D *режим_отладки*. Данная опция используется при отладке сервера. Она задается в тех случаях, когда запуск программы telnetd осуществляется вручную с консольного терминала. В зависимости от указанного режима отладки, сервер отображает информацию о соединении либо о данных, которыми он обменивается с клиентом. Режим отладки может быть задан с помощью значений options, report (оба эти значения отображают информацию об установлении соединения), netdata и ptydata (эти значения выводят соответственно сведения о входном и выходном потоках данных).

- -h. По умолчанию `telnetd` передает клиентской программе начальное сообщение, в котором содержится информация, предназначенная для пользователя. Опция -h подавляет вывод начального сообщения. Если вы опасаетесь взлома системы, но в то же время вынуждены поддерживать работу сервера Telnet, вы можете указать эту опцию для того, чтобы не предоставлять злоумышленнику дополнительные сведения о системе.
- -L *программа_регистрации*. По умолчанию `telnetd` использует для регистрации пользователей `/bin/login`. При желании вы можете указать посредством данной опции другую программу.
- -п. Подобно `rlogind`, `telnetd` проверяет наличие клиента, используя для этого специальные сообщения. Опция -п подавляет передачу данных сообщений.

При вызове сервера могут быть также указаны другие опции, большинство из которых управляет шифрованием данных и поддержкой прочих расширенных средств защиты. Поскольку в большинстве версий Telnet-серверов расширенные средства защиты отсутствуют, эти опции применяются крайне редко. Следует заметить, что разновидности Telnet-серверов, обеспечивающие шифрование данных, не пользуются большой популярностью, так как задача обмена закодированными данными гораздо лучше решается с помощью SSH. Если в вашей сети установлена система Kerberos, в ее состав обычно входят кербе-ризованные версии клиента и сервера Telnet, которые можно использовать для повышения уровня защиты при удаленной регистрации пользователей на сервере.

Редактирование начального сообщения Telnet

При получении запроса на установление соединения сервер `telnetd` читает содержимое файла `/etc/issue.net` и передает его клиенту. Данные, содержащиеся в этом файле, отображаются перед тем, как пользователь получает возможность зарегистрироваться на сервере. Опция -h, указанная при запуске `telnetd`, подавляет вывод данного сообщения. Как правило, в начальном сообщении приводятся некоторые сведения о компьютере, на котором выполняется Telnet-сервер. Ознакомившись с ними, пользователь может убедиться, что он обратился к нужному ему узлу сети. По умолчанию в начальном сообщении содержатся данные о версиях системы и ядра. Большинству пользователей эти сведения не нужны, но они наверняка заинтересуют злоумышленника, который собирается взломать систему. Прочитав начальное сообщение, он сможет определить, какое программное обеспечение выполняется на компьютере, и догадаться, какими недостатками в защите можно воспользоваться.

Аналогичные сведения отображаются при регистрации пользователя с консольного терминала (непосредственно подключенного к компьютеру). Они содержатся в файле `/etc/issue`. (При установлении соединения с помощью X Window данный файл не используется. Вопросы удаленной регистрации средствами X Window будут рассматриваться в главе 14.)

Многие системы позволяют непосредственно редактировать файл `/etc/issue.net`. Вы можете изменять текст в этом файле по своему усмотрению. В составе начального сообщения могут содержаться специальные переменные, которые `telnetd` заменяет данными о системе. Назначение этих переменных описано в табл. 13.1.

Предположим, что текст в файле `/etc/issue.net` выглядит следующим образом:

Таблица 13.1. Переменные, используемые в файле /etc/issue.net

Переменная	Описание
%t	Используемый терминал (число, описывающее устройство ввода-вывода текста)
%h	Полное доменное имя компьютера
%D	Имя домена NIS (если сервер NIS используется в сети)
%d	Текущая дата и время
%s	Имя операционной системы (Linux)
%m	Тип аппаратного обеспечения (процессора)
%r	Номер версии ядра
%v	Версия операционной системы (обычно не используется)
%%	Символ %

```
Welcome to %h.
Current time is %d.
Notice: For authorized users only!
```

Если ваш компьютер имеет имя maple.threeroomco.com, начальное сообщение будет выглядеть так:

```
$ telnet maple.threeroomco.com
Trying 172.21.32.43...
Connected to maple.threeroomco.com.
Escape character is '^]'.
Welcome to maple.threeroomco.com.
Current time is 10:57 on Monday, 12 August 2002.
Notice: For authorized users only!
```

В некоторых разновидностях Linux (в частности, в Caldera, Mandrake и некоторых версиях Red Hat) файлы /etc/issue и /etc/issue.net создаются в процессе загрузки. Формированием этих файлов занимается сценарий /etc/rc.d/rc.local. Код сценария, используемого в системе Mandrake 8.1, приведен ниже.

```
# Этот сценарий создает файл /etc/issue при каждой
# загрузке системы. Чтобы сохранить изменения, внесенные
# в файл /etc/issue, надо изменить код сценария.
if [ -x /usr/bin/linux_logo ];then
    /usr/bin/linux_logo -c -n -f > /etc/issue
    echo "" >> /etc/issue
else
    > /etc/issue
fi
echo "$R" >> /etc/issue
echo "Kernel $(uname -r) on $a $SMP$(uname -m) / \l" >> /etc/issue
if [ "$SECURITY" -le 3 ];then
    echo "Welcome to %h" > /etc/issue.net
    echo "$R" >> /etc/issue.net
    echo "Kernel $(uname -r) on $a $SMP$(uname -m)" >>
```

```
/etc/issue.net
else
    echo "Welcome to Mandrake Linux" > /etc/issue.net
    echo "————————————————————" >> /etc/issue.net
fi
```

Начиная с версии 7.2 в системе Red Hat используются статические файлы `issue` и `issue.net`. В Caldera 3.1 и Mandrake 8.1 эти файлы по-прежнему формируются с помощью сценария `/etc/rc.d/rc.local`.

В некоторых строках приведенного выше кода используются переменные, которые были определены ранее в сценарии. Такой подход позволяет системе включать в состав начального сообщения дополнительную информацию, например номер процессора. Если вы измените содержимое `/etc/issue` и `/etc/issue.net`, при следующей загрузке сценарий `/etc/rc.d/rc.local` восстановит исходный вид этих файлов. Для того чтобы изменить начальное сообщение, можно закомментировать строки сценария, которые отвечают за запись в файлы `/etc/issue` и `/etc/issue.net`, либо модифицировать код самого сценария.

Средства защиты Telnet

После отображения начального сообщения, содержащегося в файле `/etc/issue.net`, `telnetd` передает управление `/bin/login` либо программе, указанной с помощью опции -L. Программа `/bin/login` предоставляет возможность локальной и удаленной регистрации в текстовом режиме. Она отображает приглашения на ввод пользовательского имени и пароля (`login:` и `Password:`). Если регистрационные данные введены корректно, `/bin/login` отмечает время последней регистрации и вызывает оболочку, используемую по умолчанию.

Поскольку большинство серверов Telnet не шифрует данные, пересылаемые по сети, пользовательское имя и пароль передаются в незакодированном виде. Несмотря на то что при вводе пароля символы не отображаются на экране, пароль может быть перехвачен по пути следования от клиента к серверу. Если вы не указываете при вызове `telnetd` опцию -L, передача незашифрованного имени пользователя и пароля является единственным средством аутентификации, поддерживаемым `telnetd`. В отличие от `rlogind`, сервер Telnet не использует принцип доверия, т. е. не полагается на процедуру аутентификации, выполняемую на стороне клиента. (Следует заметить, что имя пользователя и пароль все же поступают на сервер с клиентского компьютера. Некоторые клиенты Telnet могут быть сконфигурированы для автоматической передачи регистрационных данных.)

Как известно, при передаче данных по Internet информационные пакеты проходят через несколько шлюзов. Если злоумышленник получит контроль хотя бы над одним из шлюзов, он сможет перехватить пароль, передаваемый при регистрации на сервере Telnet, и получит доступ к вашему компьютеру. Перехват данных можно организовать также в локальной сети, к которой подключен клиент или сервер. Компьютер, подключенный к локальной сети, но непосредственно не участвующий в процессе обмена данными, может быть переведен в режим сбора сетевых пакетов. Наряду с остальными пакетами пароль, переданный клиентом, окажется на этом компьютере.

Отсутствие шифрования передаваемых данных чревато не только перехватом пароля, но и утечкой другой важной информации. Злоумышленник может узнать содержимое

важного письма, которое вы читали, или файла, который вы редактировали. Если во время сеанса Telnet вы используете команду `su`, чтобы получить привилегии суперпользователя, вы рискуете раскрыть и этот пароль. Сказанное выше в равной степени относится также к `rlogind` и другим средствам удаленного доступа, не использующим кодирование данных. Передача незашифрованной информации при обмене с сервером Telnet или `rlogind` гораздо более опасна, чем, например, передача электронной почты, так как во время сеанса Telnet могут передаваться и пароль, и письма, и содержимое файлов, и другие важные сведения.

Что же можно предпринять для уменьшения риска потери данных? Очевидно, что радикальной мерой является переход к использованию протокола, обеспечивающего кодирование передаваемой информации. Если же это по каким-либо причинам невозможно или нежелательно, то, чтобы уменьшить вероятность утечки важных сведений, рекомендуется придерживаться ряда правил. Не просматривайте файлы, содержащие секретные данные, или важные письма при работе посредством Telnet. Не регистрируйтесь из сеанса Telnet на другом компьютере. Даже если второе соединение устанавливается по защищенному каналу, информация будет поступать на ваш компьютер в незакодированном виде и может стать доступна злоумышленнику. Не используете команду su для получения привилегий `root` и пароль, с помощью которого вы регистрируетесь на сервере Telnet, для других целей. Лучше всего Telnet подходит для взаимодействия в пределах небольшой локальной сети, не подключенной к Internet. Используя Telnet, желательно периодически изменять пароль, чтобы у взломщика, получившего пароль, было меньше возможностей нанести реальный вред вашей системе.

Настройка SSH

Из протоколов, обеспечивающих защиту передаваемых данных, среди пользователей Linux наиболее популярен SSH (Secure Shell — защищенная оболочка). Данные, предназначенные для передачи по сети посредством данного протокола, шифруются. Очевидно, что шифрованные данные могут быть перехвачены на маршрутизаторе или в локальной сети, но технология, позволяющая декодировать их, в настоящее время отсутствует. (Теоретически достаточно мощный компьютер способен справиться с задачей расшифровки информации, но для этого ему потребуется очень много времени. Кроме того, компьютеры необходимой мощности встречаются достаточно редко, и получить доступ к ним очень трудно.)

В последнее время серверы SSH используются все чаще, но они еще не стали универсальным инструментом взаимодействия. Подобно другим серверам удаленной регистрации, настройка SSH производится достаточно просто, но, чтобы установить требуемую конфигурацию системы, надо знать назначение опций, указываемых при запуске сервера, и структуру конфигурационных файлов.

В 2001 году бурно обсуждался вопрос использования названия SSH в качестве торговой марки. Не исключено, что в ближайшее время протокол SSH и одна из его реализаций (OpenSSH) изменят свое название. Что же касается коммерческого продукта, реализующего этот протокол (SSH), то он и далее будет поставляться под тем же названием. На момент написания данной книги этот вопрос еще не был решен, поэтому я использую здесь названия, указанные на Web-узлах соответствующих продуктов. Изменения, скорее всего, будут относиться к названию пакета и не затронут название программ, содержащихся в его составе.

Программное обеспечение для поддержки SSH

Существуют два основных пакета SSH, предназначенных для работы в системе Linux: коммерческий продукт SSH (`http://www.ssh.com/products/ssh/`), разработанный компанией SSH, и пакет OpenSSH, распространяемый в исходных кодах (`http://www.openssh.org`). Пакет OpenSSH входит в состав многих версий Linux, в частности, Caldera 3.1, Debian 2.2, Mandrake 8.1, Red Hat 7.2, Slackware 7.0 и SuSE 7.3. Если версия пакета, поставляемая в составе операционной системы, вас не устраивает, вы можете обратиться на Web-узел OpenSSH. (Перед использованием коммерческой версии SSH необходимо ознакомиться с лицензионным соглашением.) Далее в этой главе я буду использовать термин SSH для обозначения любой реализации данного протокола.

Проект OpenSSH имеет непосредственное отношение к операционной системе OpenBSD. Двоичные коды OpenSSH различаются для разных систем. В документе `http://www.openssh.org/portable.html` содержится информация об использовании OpenSSH в системах, отличных от OpenBSD, в том числе в различных версиях Linux. При желании вы можете скопировать с сервера исходные коды данного продукта и скомпилировать их самостоятельно.

В декабре 2001 года была создана версия 3.1 продукта SSH. В том же месяце был выпущен пакет OpenSSH 3.0.2. В версиях 3.0.x этих продуктов поддерживался приблизительно одинаковый набор возможностей и использовалась одна и та же технология кодирования. В версии 3.1 SSH была добавлена поддержка PKI (Public Key Infrastructure — инфраструктура открытого ключа), позволяющая использовать для идентификации участников взаимодействия сертификаты, реализована возможность аппаратной идентификации, а также включены другие дополнительные средства. Продукт SSH несколько опережает по своему развитию OpenSSH; это вполне объяснимо, так как протокол SSH был разработан компанией SSH.

Программы SSH и OpenSSH могут взаимодействовать между собой, поэтому, организуя обмен по протоколу SSH, не слишком важно, какой тип клиента и сервера используется на компьютерах. Существуют незначительные несоответствия между конкретными версиями пакетов, которые нетрудно учесть. Протокол SSH разработан так, что компьютеры, на которых установлено программное обеспечение, поддерживающее различные версии SSH, могут использовать средства, присутствующие в обеих версиях. Например, если на одном компьютере поддерживается SSH 2, а на другом — SSH 3, они могут взаимодействовать по протоколу SSH 2.

В большинстве случаев пакет OpenSSH поставляется в виде нескольких файлов. Наиболее важными являются `openssh`, поддерживающий базовые средства SSH, а также

openssh-client и openssh-server, которые реализуют соответственно клиентскую программу и сервер.

Протокол SSH распространен не так широко как Telnet, поэтому вам необходимо специально установить клиентские программы SSH на компьютеры ваших пользователей. Такие программы существуют для различных операционных систем. Информация о бесплатно распространяемых клиентах SSH находится на сервере `http://www.freessh.org`. Поддержку SSH обеспечивают также многие коммерческие терминальные программы, ориентированные на работу в системах Windows и MacOS. Инсталлировать клиент SSH нетрудно; сложнее доставить эти программы на все компьютеры и научить пользователей работать с ними.

Возможности SSH

Основное отличие SSH от большинства протоколов удаленной регистрации заключается в том, что SSH обеспечивает шифрование передаваемых данных. Кроме того, данный протокол поддерживает перенаправление, или **туннелирование**, сетевых портов между клиентом и сервером. Это значит, что посредством **SSH-соединения** могут передаваться пакеты других протоколов. Возможна конфигурация, при которой SSH-соединение будет автоматически использоваться для обмена данными по некоторому протоколу, например, такое взаимодействие реализовано для X Window. (Подробно вопрос установления соединений средствами X Window будет рассмотрен в главе 14.) Затратив определенные усилия для настройки системы, можно реализовать туннелирование сообщений любого протокола посредством защищенного соединения. Вы можете даже использовать средства PPP для создания сетевого интерфейса, предполагающего туннелирование средствами SSH. В результате можно получить *виртуальную частную сеть* (VPN — Virtual Private Network). Необходимые действия по настройке такой сети описаны в документе VPN HOWTO (`http://www.linuxdoc.org/HOWTO/VPN-HOWTO.html`).

Заслуживает внимания еще одна возможность протокола SSH — непосредственная реализация инструментов, не связанных с регистрацией в системе. В частности, в состав пакета SSH входит программа под названием scp, которая используется для копирования файла с одного компьютера на другой. Формат вызова этой команды приведен ниже.

```
scp [[пользователь1]узел1:]имя_файла1 \
[[пользователь2]узел2:]имя_файла2]
```

Вызов `scp` очень похож на вызов программы `rcp`, реализующей r-команду, с помощью которой осуществляется копирование файлов. Программа scp создана для замены `rcp`. В отличие от `rcp` и многих других инструментов передачи файлов (например, FTP), `scp` копирует данные в зашифрованном виде, кроме того, она шифрует имя пользователя и пароль. Такую программу удобно использовать для передачи важных данных по сетям, в которых не гарантируется защита информации.

Дополнительные интерактивные возможности при передаче файлов обеспечивает программа `sftp`, которая работает подобно традиционной программе `ftp`, но защищает содержимое файлов и регистрационные данные посредством кодирования. Некоторые FTP-**клиенты** с графическим интерфейсом, например gFTP (`http://gftp.seul.org`), также поддерживают передачу данных на базе SSH. Таким образом, средства SSH, по сути, дублируют функции Telnet и FTP.

Стандартный сервер SSH (программа `sshd`) поддерживает как работу SSH-клиента (в системе Linux это программа `ssh`), так и обмен данными с программами scp и `sftp`.

Этот сервер также обеспечивает туннелирование портов. Весь трафик проходит через стандартный SSH-порт 22.

Опции, используемые при запуске сервера SSH

Независимо от того, какую реализацию протокола SSH вы используете, сервер обычно запускается с помощью сценария SysV. Запуск сервера может осуществляться также посредством суперсервера, но такая возможность используется крайне редко. Дело в том, что запуск старых версий сервера происходил с большой задержкой, так как некоторые операции, выполняемые при запуске, создавали большую нагрузку на процессор. Если сервер выполняется на современном оборудовании, задержка практически не заметна, поэтому при желании вы можете сконфигурировать sshd для запуска посредством суперсервера. При этом необходимо указывать опцию `-i`, назначение которой будет рассмотрено ниже.

При вызове сервера SSH могут быть заданы различные опции, изменяющие поведение программы. Опции, предусмотренные в пакете OpenSSH 3.0.2, перечислены ниже.

- `-d`. В обычных условиях сервер выполняется в режиме демона. Данная опция задает режим отладки, при котором сервер выполняется в качестве задачи переднего плана, поддерживает лишь одно соединение и выводит дополнительную отладочную информацию. Указывая дополнительные опции -d (sshd поддерживает до трех символов d), вы можете задать вывод дополнительных сведений о работе программы.

- -D. Эта опция отменяет режим демона, но, в отличие от опции -d, она не переводит сервер в режим отладки.

- -e. Данная опция указывает на то, что сообщения об ошибках, генерируемые sshd, не должны записываться в файл протокола, а должны выводиться в стандартный поток ошибок.

- `-f` *конфигурационный_файл*. *В* качестве конфигурационного файла сервер обычно использует `/etc/ssh/sshd_config`, но с помощью данной опции вы можете указать другой файл.

- `-i`. Данная опция сообщает программе о том, что программа была запущена посредством суперсервера (inetd или xinetd). Если для запуска sshd используется суперсервер, надо обязательно указывать данную опцию.

- `-p` *порт*. Эта опция задает порт для сервера. По умолчанию используется порт 22.

- `-q`. Данная опция подавляет протоколирование. (Обычно в файл протокола записывается информация об установлении соединения, выполнении аутентификации и разрыве соединения.)

- -4. По умолчанию sshd поддерживает соединения с компьютерами, адреса которых соответствуют либо протоколу IPv4, либо протоколу IPv6. Данная опция указывает на то, что sshd должен устанавливать соединения только с клиентами, имеющими адреса IPv4.

- `-6`. Данная опция разрешает установление соединений только с клиентами, имеющими адреса IPv6.

При запуске sshd могут задаваться и другие опции, большинство из которых определяют особенности кодирования данных. Дополнительную информацию об этих опциях можно получить на страницах справочной системы, посвященных sshd.

Перед первым запуском sshd необходимо сгенерировать файлы, содержащие ключи кодирования. При выполнении алгоритма SSH эти ключи используются для идентификации участников взаимодействия и кодирования данных. В большинстве случаев в сценариях SysV, осуществляющих запуск сервера, предусмотрен код, который поверяет наличие файлов с ключами кодирования и при необходимости генерирует их. Если в вашей системе подобный код отсутствует, вы можете использовать для генерации файлов следующие команды:

```
# ssh-keygen -q -t rsa1 -f /etc/ssh/ssh_host_key -C '' -N ''
# ssh-keygen -q -t rsa -f /etc/ssh/ssh_host_rsa_key -C '' -N ''
# ssh-keygen -q -t dsa -f /etc/ssh/ssh_host_dsa_key -C '' -N ''
```

Каждая из приведенных выше команд генерирует два ключа: *закрытый,* или *личный,* ключ (private key), используемый только на сервере, и *открытый,* или *общий,* ключ (public key). Открытый ключ передается клиенту, чтобы он мог кодировать данные, передавая их на сервер. Закрытый и открытый ключи помещаются в файлы, имена которых отличаются друг от друга лишь тем, что к имени файла с открытым ключом добавляется суффикс .pub. Перед вызовом этих команд необходимо проверить наличие шести файлов: (ssh_host_key, ssh_host_key.pub, ssh_host_rsa_key, ssh_host_rsa_key.pub, ssh_host_dsa_key и ssh_host_dsa_key.pub (обычно эти файлы размещаются в каталоге /etc/ssh). Если вы измените существующие ключи, вам придется переконфигурировать клиентские программы, настроенные на работу со старыми ключами. Поэтому ключи следует изменять только в том случае, когда это действительно необходимо.

Редактирование файла sshd_config

Работой сервера sshd управляет файл sshd_config, который обычно находится в каталоге /etc/ssh. (Не следует путать файл sshd_config с конфигурационным файлом клиента ssh_config, который размещается в том же каталоге.) В файле sshd_config указываются опции и их значения. Каждая опция задается в отдельной строке в следующем формате:

Опция значение

Подобно другим конфигурационным файлам, строка, начинающаяся с символа #, содержит комментарии. Многие опции в файле sshd_configдублируют опции командной строки, которые указываются при вызове sshd, но некоторые из опций могут присутствовать только в конфигурационном файле. Конфигурация, установленная по умолчанию, чаще всего обеспечивает нормальную работу сервера, но иногда приходится изменять значения некоторых опций, например PermitRootLogin. Наиболее важные из опций, содержащихся в файле sshd_config, приведены ниже.

- Port. Данная опция позволяет задать порт для сервера. По умолчанию используется порт 22.
- HostKey. Эта опция сообщает серверу о том, где следует искать ключи кодирования. Ключи кодирования содержатся в файлах, которые должны быть сгене-

рированы перед первым запуском программы. Примером такого файла является /etc/ssh/ssh_host_key. При настройке сервера можно указать несколько файлов с ключами.

- KeyRegenerationInterval. При установлении соединения участники SSH-взаимодействия ведут переговоры об использовании ключей кодирования, а затем время от времени договариваются о замене ключей. Периодическая замена ключей уменьшает опасность повреждения системы в случае, если по каким-либо причинам ключ будет расшифрован. (Обратите внимание на то, что здесь речь идет о ключах, сгенерированных в дополнение к ключам, которые создаются перед первым запуском программы. Ключи, сформированные в процессе переговоров, никогда не записываются на диск.) Данная опция задает время (в секундах) использования сгенерированных ключей. По истечении этого времени формируются новые ключи.
- PermitRootLogin. В большинстве случаев при инсталляции пакета устанавливается значение yes данной опции. По умолчанию sshd позволяет пользователю root регистрироваться на сервере. Безопаснее, однако, задать для этой опции значение по, так как в этом случае злоумышленник, пытающийся незаконно проникнуть в систему, должен знать два пароля (пароль обычного пользователя и пароль root). Значение по опции PermitRootLogin не исключает возможность удаленного администрирования системы, но для этого вам придется сначала зарегистрироваться как обычный пользователь, а затем получить привилегии root с помощью команды su.
- IgnoreRhosts. По умолчанию устанавливается значение yes данной опции, в результате чего сервер sshd игнорирует файл ~/.rhosts. Если опция IgnoreRhosts имеет значение по и если значение опции RhostsAuthentication равно yes, sshd, подобно rlogind, будет поддерживать аутентификацию по принципу доверия. Установка значения по опции IgnoreRhosts создает реальную опасность для системы.
- RhostsAuthentication. Для поддержки аутентификации по принципу доверия сервер SSH использует две опции: IgnoreRhosts и RhostsAuthentication. Опция
  RhostsAuthentication разрешает работу с узлами, пользующимися доверием. Желательно установить для данной опции значение по.
- RSAAuthentication. В версии *1* протокола SSH был предусмотрен метод аутентификации с применением открытого ключа, при котором пароль не передавался по сети. Вместо этого использовались открытый ключ и фраза пароля. Для того чтобы разрешить данный способ аутентификации, надо установить значение yes опции RSAAuthentication (это значение принимается по умолчанию).
- PubkeyAuthentication. Данная опция выполняет те же действия, что и опция RSAAuthentication, но применяется при работе с версией 2 протокола SSH.
- PasswordAuthentication. Значение yes данной опции позволяет пользователям регистрироваться, вводя пароль в ответ на приглашение. Этот способ аутентификации широко используется в настоящее время, поэтому желательно принять значение опции PasswordAuthentication, установленное по умолчанию.

- `X11` Forwarding. Как было сказано ранее, протокол SSH может быть использован для туннелирования X-соединений. Чтобы это стало возможным, соответствующая конфигурация должна быть установлена как для сервера, так и для клиентской программы. Значение yes опции `X11Forwarding` указывает на то, что сервер SSH должен перенаправлять соединения X Window. Опция аналогичного назначения предусмотрена и для клиента SSH. Имя этой опции — ForwardXll; она указывается в файле `/etc/ssh/ssh_config`.

При настройке сервера SSH могут быть использованы дополнительные опции. Одни из них определяют альтернативные способы аутентификации, другие уточняют действие перечисленных выше опций, а третьи задают детали функционирования сервера. После установки пакета, реализующего SSH-взаимодействие, внимательно просмотрите конфигурационный файл, который поставляется в составе пакета, и выясните, подходит ли вам конфигурация, установленная по умолчанию. Дополнительную информацию о назначении опций можно найти на страницах справочной системы, посвященных sshd.

Аутентификация при SSH-взаимодействии

В процессе обмена сервер и клиент SSH используют различные способы кодирования передаваемых данных. Упрощенно это выглядит так. Участники взаимодействия договариваются о временном использовании метода кодирования с помощью открытого ключа. Ключ представляет собой достаточно большое число и применяется для шифрования информации, передаваемой по сети. При получении данных система декодирует их с помощью закрытого ключа. Такой способ кодирования используется для передачи другого типа ключа, называемого секретным ключом. Секретный ключ используется в другом методе шифрования и обеспечивает более высокое быстродействие по сравнению с кодированием посредством открытого и закрытого ключей. По завершении передачи секретного ключа компьютеры, обменивающиеся по протоколу SSH, начинают использовать для кодирования данных секретный ключ. Помимо передачи секретного ключа, открытый и закрытый ключи применяются также при аутентификации; идентификатор пользователя, зашифрованный с помощью закрытого ключа, позволяет убедиться в том, что пользователь — именно тот, за кого он себя выдает.

Действия, выполняемые при SSH-аутентификации

В протоколе SSH предусмотрено несколько способов аутентификации пользователей. Детали процесса аутентификации различаются в зависимости от версии протокола. В общих чертах алгоритм аутентификации выглядит следующим образом.

1. Клиент предпринимает попытку аутентификации, основанной на принципе доверия, однако в большинстве случаев значения опций `RhostsAuthentication` и IgnoreRhosts запрещают этот способ аутентификации. Если же попытка оказывается успешной, клиент получает доступ к ресурсам сервера; при этом пользователю не приходится указывать имя и пароль.

2. Клиент пытается использовать способ аутентификации, представляющий собой сочетание принципа доверия и RSA-аутентификации. В большинстве случаев такая попытка тоже не имеет успеха.

3. Клиент пытается использовать RSA-аутентификацию, которая предполагает передачу специального файла. Если такой файл присутствует на сервере и если по-

следующие действия в рамках этого метода оказывается успешными, пользователь получает доступ к системе. В зависимости от конфигурации, от пользователя может потребоваться ввод фразы пароля. Для того чтобы этот метод мог быть использован, и клиент, и сервер должны быть настроены соответствующим образом.

4. Если все предшествующие попытки аутентификации окончились неудачей, пользователь должен ввести пароль. Этот пароль передается серверу в закодированном виде и применяется для аутентификации пользователя.

Сервер хранит файлы, содержащие ключи, в каталоге /etc/ssh и применяет их при работе со всеми пользователями. SSH-клиенты получают открытые ключи от серверов и могут использовать их для аутентификации этих серверов. Открытые ключи хранятся в каталоге ~/.ssh, и за их поддержку отвечает пользователь. При первом обращении клиента к серверу по протоколу SSH клиентская программа оповещает пользователя о том, что в файл записан новый ключ. (Конфигурация программы может быть установлена так, что для выполнения этого действия потребуется подтверждение пользователя.) Если сервер изменит ключ, ssh отобразит предупреждающее сообщение, которое выглядит следующим образом:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!         @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
```

Клиент может отображать дополнительную информацию в составе сообщения, но в любом случае соединение с сервером не устанавливается. Если пользователь согласен работать с новыми ключами, он должен удалить запись, соответствующую серверу, из файла ~/.ssh/known_hosts или ~/.ssh/known_hosts2 (имя файла зависит от версии протокола).

В отличие от telnetd, sshd не передает управление программе /bin/login; вместо этого sshd самостоятельно осуществляет регистрацию пользователя. (Если вы хотите, чтобы sshd использовала программу login, необходимо задать соответствующее значение опции UseLogin в файле sshd_config.) Таким образом, sshd можно рассматривать как сочетание программ telnetd и login. Исключив программу login из процесса регистрации, sshd получает возможность выполнять аутентификацию с помощью открытого ключа.

Ключи для автоматической регистрации и повышения безопасности системы

Пользователи могут генерировать ключи и использовать их для регистрации на сервере. Эти ключи хранятся на клиентской машине в каталоге ~/.ssh. Как известно, по умолчанию при аутентификации на сервере SSH пользователь должен задавать пароль. Если вы передадите открытый ключ на сервер, то будете избавлены от необходимости указывать пароль при регистрации. Для повышения уровня защиты при регистрации с помощью открытого ключа может также быть предусмотрен ввод специальной фразы пароля; это еще больше усложнит процедуру незаконного проникновения в систему. Чтобы аутентификация с помощью открытого ключа стала возможной, выполните следующие действия.

1. Зарегистрируйтесь на компьютере, выполняющем роль клиента при SSH-обмене.

2. Введите команду генерации ключей, соответствующих версии 2 SSH. Эта команда приведена ниже; для ее выполнения потребуется несколько секунд.

```
$ ssh-keygen -q -t rsa -f ~/.ssh/id_rsa -C '' -N ''
```

Если вы не укажете опцию -N '', утилита ssh-keygen запросит фразу пароля. Если перед нажатием клавиши <Enter> вы введете какие-либо символы, вам придется указывать их при каждом соединении с сервером. С точки зрения пользователя использование фразы пароля при регистрации принципиально не отличается от указания пароля.

3. Скопируйте файл ~/id_rsa.pub в свой рабочий каталог на сервере. (Этот файл содержит открытый ключ. Его имя отличается от имени файла, которое вы задали при вызове предыдущей команды, суффиксом .pub.) Для копирования можно использовать команду scp.

```
$ scp ~/.ssh/id_rsa.pub server:.ssh/id_rsa.client
```

4. Зарегистрируйтесь на сервере. Для этого вы можете использовать ssh, но вам придется ввести пароль.

5. Сделайте каталог ~/ . ssh текущим. Если вы выведете список файлов, то увидите, что в этом каталоге присутствует файл id_rsa.client.

6. Добавьте открытый ключ в файл authorized_keys2. Это можно сделать с помощью следующей команды:

```
$ cat id_rsa.client >> authorized_keys2
```

С этого момента вы можете устанавливать соединение с сервером, используя протокол SSH 2. Если вы не указали фразу пароля, при регистрации вам не придется вводить какие-либо идентификационные данные. Опция -2 указывает SSH-клиенту на то, что взаимодействие должно осуществляться с использованием версии 2 протокола SSH.

```
$ ssh -2 server
```

ВНИМАНИЕ Если вы применяете открытый ключ для аутентификации, необходимо принять меры для того, чтобы закрытый ключ не стал доступен посторонним. Если злоумышленник сможет получить ваш закрытый ключ, он сможет обращаться к серверу под вашим именем. В протоколе SSH ключ связывается с определенным IP-адресом, и регистрироваться с помощью открытого ключа можно только с одного компьютера, но тот, кто пытается проникнуть в систему, сумеет без труда обойти это ограничение (именно поэтому защита rlogind не соответствует современным требованиям). Применение фразы пароля повысит уровень защиты, так как, чтобы проникнуть в систему, надо не только получить в свое распоряжение открытый ключ, но и узнать нужную фразу. Однако, если при каждой регистрации на сервере пользователю придется вводить идентификационные данные (в частности, фразу пароля), работа посредством протокола SSH будет гораздо менее удобной.

Для того чтобы обеспечить RSA-аутентификацию при использовании протокола SSH 1, ваши действия должны несколько отличаться от приведенных выше. Новая процедура и процедуры, описанные ранее, имеют следующие отличия.

- В п. 2 вместо `-t rsa -f ~/.ssh/id_rsa` следует указать `-t rsa1 -f ~/.ssh/identity`. При этом будет сгенерирована пара ключей RSA по соглашениям версии 1. Аналогичным образом надо изменить имена файлов на других стадиях процедуры.
- В п. 6 открытый ключ из `identity.pub` копируется не в `authorized_keys2`, а в файл `authorized_keys`.
- При установлении соединения, вызывая ssh, не надо указывать опцию -2.

Обе описанные здесь процедуры предполагают, что сервер сконфигурирован для выполнения аутентификации с помощью открытого ключа. Как вы уже знаете, для этой цели используются опции `RSAAuthentication`(версия 1) и `PubkeyAuthentication` (версия 2), задаваемые в конфигурационном файле `/etc/ssh/sshd_config`.

Заметьте, что пользоваться SSH-соединением можно, не настраивая программы для аутентификации с помощью открытого ключа. Подобная аутентификация лишь исключает необходимость ввода пароля либо обеспечивает дополнительную степень защиты за счет использования фразы пароля.

Применение ssh-agent

SSH-аутентификацию можно также организовать с помощью инструмента, который называется ssh-agent. Программа ssh-agent осуществляет управление SSH-ключами так, что фразу пароля приходится вводить лишь один раз. Для того чтобы работа с ssh-agent стала возможной, выполните следующие действия.

1. Создайте закрытый и открытый ключи, выполнив описанную выше процедуру, и скопируйте открытый ключ в свой рабочий каталог на сервере SSH. При вызове `ssh-keygen` не следует задавать опцию `-N ''`, чтобы закрытый ключ был защищен фразой пароля.
2. На компьютере, на котором выполняется клиентская программа SSH, задайте команду ssh-agent /bin/bash, и она запустит на выполнение программу ssh-agent и новую оболочку Bash. В результате ssh-agent будет контролировать все процессы, порожденные новой оболочкой. (При необходимости вы можете использовать вместо Bash другую оболочку.)
3. Чтобы добавить RSA-ключ SSH к кэшу ключей ssh-agent, вызовите команду `ssh-add ~/.ssh/id_rsa`. (При использовании версии 1 SSH задавать ~/.ssh/id_rsa не следует.) Если ключ защищен фразой пароля, ssh-add запросит ее.

С этого момента вы можете обращаться к серверу SSH с помощью SSH-клиента; причем вам не придется вводить ни пароль, ни фразу пароля. Программа ssh-agent хранит ключи в памяти и устанавливает переменные окружения так, что клиент SSH взаимодействует с ssh-agent и получает значения ключей. Доступ к ключам имеют только программы, являющиеся дочерними по отношению к ssh-agent, но если ssh-agent запускает новую оболочку, то все программы, вызванные из этой оболочки, в том числе ssh, также становятся дочерними для ssh-agent.

Если вам необходимо установить одно соединение, данный подход не оправдывает себя, так как перед вызовом ssh вам необходимо запустить ssh-agent и включить ключи

в кэш посредством **ssh-add**, кроме того, придется один раз ввести фразу пароля. Применение ssh-agent позволит сэкономить время в том случае, если вы регистрируетесь на нескольких компьютерах с помощью одного ключа либо если вам часто приходится повторно регистрироваться на одном компьютере. Существует несколько способов, позволяющих упростить работу с ssh-agent.

- Вы можете внести изменения в файл /etc/passwd так, чтобы оболочка вызывалась посредством ssh-agent. Например, если в /etc/passwd указана оболочка /bin/bash, вы можете задать в соответствующем поле **/usr/bin/ssh-agent** /bin/bash. (При необходимости вы можете изменить путь к ssh-agent или использовать другую оболочку.) После этого вам не придется вручную задавать команду ssh-agent /bin/bash; достаточно будет зарегистрироваться в системе, ввести ssh-add ~/.**ssh/id_rsa** и использовать ssh для регистрации на удаленном компьютере. Данный подход применим только в том случае, когда регистрация в системе осуществляется в текстовом режиме. Если вы используете графическую среду, то для каждого нового окна **xterm** будет создаваться новое окружение ssh-agent, что приведет к непроизводительному расходу ресурсов.

- Если вы регистрируетесь в текстовом режиме и запускаете X Window с помощью startx, вы можете вместо этой команды задать ssh-agent startx. При этом ssh-agent становится родительской программой по отношению ко всем процессам X Window и может предоставлять им информацию о ключах.

- Если вы используете инструменты регистрации с графическим интерфейсом, вам надо сохранить файл **.xsession** (или другой файл аналогичного назначения) под именем **.xsession-nosshagent**, а затем создать новый файл .xsession, включив в него единственную команду ssh-agent **~/.xsession-nosshagent**. В результате ssh-agent станет родительской программой для всех процессов X **Window**, и вам не придется повторно вводить ключи, добавленные в кэш программой ssh-add, даже если вы будете запускать клиент SSH из различных окон.

После запуска ssh-agent и указания ключей вы можете просмотреть введенные ключи, задав команду ssh-add **-l**. Для удаления ключей надо использовать команду ssh-add -d. Если после вызова ssh-add -d вы захотите установить SSH-соединение, вам придется повторно ввести ключ (или указать пароль).

Одно из преимуществ использования ssh-agent состоит в том, что для взаимодействия с различными серверами SSH вам надо лишь скопировать открытый ключ на каждый сервер. Устанавливая соединение с несколькими серверами, вы можете использовать один и тот же открытый ключ и программу ssh-agent. Если вы предпочитаете применять различные ключи для работы с разными серверами, вам придется хранить ключи в отдельных файлах и при загрузке каждого из них вводить соответствующую фразу пароля. Если вам понадобится установить соединение с компьютером, на котором отсутствует открытый ключ, вам придется ввести пароль так же, как вы это делаете при обычном обращении без использования ssh-agent.

Резюме

Сервер удаленной регистрации позволяет пользователям запускать программы и вызывать команды из оболочки во время работы на удаленном компьютере. Такой режим работы очень удобен для пользователя, но, поддерживая средства удаленной регистрации, приходится принимать дополнительные меры по обеспечению безопасности системы.

Наиболее часто в системе Linux используются серверы удаленной регистрации `rlogind`, Telnet и SSH. Из них наилучшую защиту обеспечивают средства SSH, поэтому именно их желательно использовать для установления соединения с Internet. (Существуют также разновидности Telnet, реализующие повышенный уровень защиты, но они применяются чрезвычайно редко.) Использование `rlogind` и Telnet оправдано в небольших локальных сетях, полностью контролируемых системными администраторами. В средах, в которых могут предприниматься попытки взлома систем, особенно в Internet, применять эти серверы не рекомендуется, так как их средства защиты не отвечают современным требованиям. Все три сервера достаточно просты в настройке. Если конфигурация, предложенная по умолчанию, не устраивает вас, вы можете без труда установить нужные параметры, изменяя значения опций. В особенности это относится к серверу SSH, в котором реализовано несколько механизмов аутентификации.

Глава 14

Организация удаленного доступа с помощью X Window и VNC

В главе 13 рассматривались серверы удаленной регистрации **rlogind**, Telnet и SSH. Совместно с клиентскими программами, выполняющимися на других компьютерах, эти серверы позволяют пользователям регистрироваться в системе Linux и выполнять приложения и утилиты, работающие в текстовом режиме. Поскольку в Linux (а также в UNIX) существует большое количество разнообразных программ, предоставляющих пользователю алфавитно-цифровой интерфейс, серверы удаленной регистрации дают возможность выполнять самые различные задачи. Однако многие пользователи предпочитают работать с инструментами, поддерживающими графический интерфейс. Подобные инструменты не могут выполняться в текстовом режиме, поэтому, зарегистрировавшись на удаленном компьютере с помощью **rlogind**, Telnet или SSH, нельзя запускать такие приложения, как The GIMP, Netscape Navigator или StarOffice. (Существуют программы, например Emacs, которые поддерживают как графический, так и текстовый интерфейс, но наличие такой поддержки является скорее исключением, чем правилом.) Чтобы предоставить доступ к программам с графическим интерфейсом, выполняющимся на удаленном компьютере, нужен специальный графический сервер. Наиболее часто для обеспечения графического пользовательского интерфейса в Linux используется система X Window. Являясь частью современных версий Linux, изначально она была создана для работы в сети, поэтому, для того, чтобы приложения с графическим интерфейсом могли выполняться, вам надо лишь установить на пользовательском компьютере соответствующие программы. Помимо X Window, для обеспечения работы приложений с графическим интерфейсом применяется также система VNC (Virtual Network Computing — вычисления в виртуальной сети), которая работает подобно X Window, но использует другие протоколы. Обе системы будут рассмотрены в данной главе.

Использование серверов удаленного доступа, поддерживающих графический интерфейс

Серверы удаленного доступа, поддерживающие графический интерфейс, в основном нужны тогда, когда компьютер должен выполнять роль рабочей станции для нескольких пользователей, работающих на удаленных компьютерах. Например, для рабочей группы, насчитывающей около десяти сотрудников, может быть приобретен один мощный компьютер и несколько компьютеров малой мощности, которые будут выполнять функции терминалов центральной машины. На центральном компьютере можно запускать приложения, интенсивно потребляющие ресурсы, например StarOffice, The GIMP, KMail и др. Пользователи, работающие на маломощных машинах, могут регистрироваться в центральной системе и запускать приложения на удаленном компьютере. Такая структура имеет ряд преимуществ по сравнению с конфигурацией, при которой каждый пользователь запускает необходимые ему программы на своей рабочей станции. Эти преимущества описаны ниже.

- Централизованное администрирование программного обеспечения. Для добавления, удаления или обновления приложений необходимо изменить конфигурацию лишь одного компьютера. (Альтернативой такому подходу является хранение приложений на файловом сервере и выполнение их на рабочих станциях.)

- Простая конфигурация рабочих станций. При использовании центрального компьютера аппаратное и программное обеспечение рабочих мест пользователей может быть очень простым. Для поддержки пользовательских компьютеров приходится затрачивать минимальные усилия. Администрирование подобных рабочих станций часто осуществляется с удаленного компьютера. По сути, в такой системе можно использовать X-терминалы — чрезвычайно простые компьютеры, поддерживающие X Window.

- Упрощенная процедура обновления аппаратных средств. Если новые приложения требуют дополнительных аппаратных ресурсов, изменения затрагивают только центральный компьютер. При этом затрачивается намного меньше средств, чем при обновлении десятка рабочих станций. С другой стороны, при таком подходе работа всей группы зависит от исправности одного компьютера. Кроме того, центральная машина должна иметь мощность, намного превышающую мощность любой рабочей станции.

- Централизованное хранение данных. Системному администратору гораздо проще обслуживать один центральный компьютер, чем десяток рабочих станций. В частности, такой подход упрощает создание резервных копий. Возможна конфигурация пользовательских компьютеров, при которой резервное копирование содержащейся на них информации не понадобится. Администратору потребуется создать лишь одну резервную копию, стандартную для всех рабочих станций. Она не займет много места и может поместиться, например, на одном компакт-диске.

- Централизованная поддержка учетных записей пользователей. При работе на одном центральном компьютере пользователю нужна учетная запись лишь на одной машине. Это существенно упрощает администрирование сети. Среда, предоставляемая пользователю, не зависит от того, с какого рабочего места он зарегистрировался.

При замене пользовательского компьютера его либо вовсе не придется настраивать, либо такая настройка сведется к минимуму.

Рассматриваемая конфигурация сети имеет существенный недостаток. Поскольку все пользователи выполняют программы на одном компьютере, выход этого компьютера из строя означает остановку работы всей группы. Без центрального компьютера остальные машины оказываются практически бесполезными. Если вы решите сконфигурировать сеть подобным образом, вам придется уделять большое внимание созданию резервных копий центральной системы и держать их под рукой на случай выхода системы из строя. Возможно, вы даже будете вынуждены создать резервный сервер, готовый в любую минуту вступить в строй вместо основного сервера.

Даже если вы не собираетесь организовывать работу пользователей большой сети на центральном сервере, вам все равно понадобятся инструменты регистрации, поддерживающие графический интерфейс. При необходимости они позволят обращаться к серверам, решающим частные задачи, либо регистрироваться на компьютерах других пользователей. Если каждый пользователь будет иметь собственную рабочую станцию достаточной мощности, ее можно будет сконфигурировать для обслуживания обращений извне. В результате пользователь сможет вызвать приложение, находясь дома или будучи зарегистрированным на другом компьютере.

Средства удаленной регистрации, поддерживающие графический интерфейс, чаще всего применяются в локальных сетях. Поскольку для отображения графических элементов приходится передавать большой объем данных, использование соответствующих протоколов при работе в Internet существенно снизит скорость отображения информации. Даже в сети с пропускной способностью 100 Мбод применение протоколов удаленной регистрации заметно уменьшает быстродействие программ по сравнению с выполнением их на локальной машине, однако скорость работы остается в допустимых пределах. Подобно инструментам, которые осуществляют регистрацию в текстовом режиме, серверы, поддерживающие графический интерфейс, предоставляет пользователю все привилегии, необходимые для работы на удаленном компьютере. Однако при регистрации производится передача пароля в незакодированном виде, что создает опасность для системы. (Инструмент VNC предусматривает шифрование паролей; это несколько снижает риск, но остальные данные передаются в незашифрованном виде. Использование SSH в дополнение к средствам удаленной регистрации обеспечивает кодирование как паролей, так и всей информации, передаваемой в течение сеанса.)

Обеспечение удаленного доступа средствами X Window

Графическая среда, реализуемая средствами X Window, очень часто используется в системе Linux. Как было сказано ранее, система X Window поддерживает взаимодействие по сети. Даже если все компоненты X-программы выполняются на одном и том же компьютере, для отображения окон, меню, диалоговых окон и других элементов графического интерфейса используются сетевые протоколы. Чтобы правильно сконфигурировать клиент и сервер для обеспечения удаленного доступа с поддержкой графического интерфейса, необходимо ясно представлять себе работу этих протоколов.

Взаимодействие клиента и сервера в системе X Window

Пользователи, не искушенные в вопросах применения вычислительной техники и сетевых протоколов, представляют себе сервер как большой мощный компьютер, находящийся в отдельной комнате. Пользователи работают за клиентскими машинами и время от времени обращаются к серверу. Такое представление, хотя и не идеально с технической точки зрения, не слишком далеко от истины. Действительно, в качестве сервера обычно используется мощный компьютер, а мощность клиентских компьютеров, взаимодействующих с сервером, часто бывает ограничена. Однако по отношению к системе X Window данная схема неприменима; в X Window все обстоит совершенно по-другому. Пользователи работают за компьютерами, представляющими собой серверы X Window, причем эти компьютеры могут быть низкоуровневыми. Клиент-программы X Window, как правило, выполняются на компьютерах, которые значительно превышают по мощности машины, выполняющие роль X-серверов.

Для того чтобы понять, почему так происходит, надо рассмотреть процесс обмена данными с точки зрения клиентской программы. Когда клиентская программа работает в сети, она устанавливает соединения с серверами, предназначенные для передачи данных. Программа-сервер предоставляет некоторые услуги клиенту. Рассмотрим в качестве примера текстовый процессор WordPerfect и его взаимодействие с сервером NFS. С помощью интерфейсных средств WordPerfect пользователь задает команду на получение файла из каталога, экспортируемого средствами NFS. В результате WordPerfect инициирует передачу данных по сети, т. е. передает серверу NFS запрос, означающий, что тот должен доставить файл. (На самом деле роль клиента NFS выполняет модуль ядра Linux, WordPerfect лишь задает модулю команду передачи данных.) В ответ на запрос сервер NFS предоставляет клиенту необходимый ресурс, в данном случае файл. Теперь представьте себе, что WordPerfect выполняется на удаленном узле и взаимодействует с пользователем посредством сервера X Window. Чтобы отобразить окно выбора файла, текстовый редактор должен запросить у сервера X Window услуги по выводу диалогового окна, текста и других данных. С точки зрения WordPerfect сервер X Window принципиально не отличается от других серверов, поддерживающих ввод-вывод данных. Тот факт, что выводимые данные будут представлены пользователю, не имеет значения для программы. Взаимодействие клиента X Window с серверами условно показано на рис. 14.1.

Сервер X Window представляет устройство отображения и одно или несколько устройств ввода (обычно клавиатуру и мышь). Сам X-сервер может выступать в роли кли-

Рис. 14.1. Сервер X Window обеспечивает интерфейс между пользователем и программой, выполняющейся на удаленном компьютере

ента по отношению к другим серверам, например серверу шрифтов (он будет рассмотрен в главе 15). В ответ на запрос X-сервера сервер шрифтов доставляет ему шрифты, необходимые для отображения данных.

Принцип взаимодействия клиента и сервера VNC противоположен принципу, реализованному в системе X Window. Детально работа VNC будет рассмотрена далее в этой главе. Отличие между данными протоколами приводит к тому, что одна система может успешно использоваться там, где вторая работает неэффективно либо не работает вовсе. В качестве примера можно привести конфигурацию сети, в которой между клиентом и сервером расположен брандмауэр. Если на пользовательских компьютерах выполняются X-серверы, необходимо специально сконфигурировать брандмауэр. При использовании VNC специальная настройка таблиц брандмауэра не требуется. Средства SSH создают иллюзию того, что клиент и сервер X Window "поменялись местами" и взаимодействие между ними осуществляется так же, как и в большинстве сетевых служб.

X-серверы в различных операционных системах

Программы, реализующие сервер X Window, доступны не только в Linux и UNIX, но и в других операционных системах. Если вы хотите использовать в качестве терминала компьютер под управлением Windows, OS/2 или MacOS, вы должны лишь инсталлировать в системе X-сервер. Для этого хорошо подходят продукты XFree86 (`http://xfree86.cygwin.com` — для Windows, `http://ais.gmd.de/~veit/os2/xf86os2.html` — для OS/2 и `http://mrcla.com/XonX/` — для MacOS X), MI/X — для Windows и MacOS Classic (`http://www.microimages.com/freestuf/mix/`), Exceed — для Windows (`http://www.hcl.com/products/nc/exceed/`), Xmanager — для Windows (`http://www.netsarang.com/products/xmanager.html`) и Xtools — для MacOS X (`http://www.tenon.com/products/xtools/`). Данный список не охватывает все доступные продукты. Так, например, только для Windows существует множество серверов, специально предназначенных для получения доступа к системе Linux. Материалы дискуссии, позволяющей оценить возможности различных серверов X Window, находятся по адресу `http://www.microimages.com/mix/prices.htm`.

Как было сказано ранее, существуют специальные устройства, называемые X-терминалами. Они поддерживают функции сервера X Window, обеспечивают обмен по сети TCP/IP, и этим их возможности практически исчерпываются. X-терминал по сути представляет собой выделенный X-сервер. X-терминалы поставляют некоторые компании, например Network Computing Devices (NCD; `http://www.ncd.com`) и Hewlett Packard (`http://www.hp.com`). Для работы этих устройств, как правило, необходимо, чтобы в сети выполнялся сервер TFTP (Trivial File Transfer Protocol — простой протокол передачи файлов); он нужен для передачи загрузочных файлов. (Сервер TFTP не обязательно должен находиться на компьютере, использующем X-терминал.) Кроме того, X-терминалы также требуют, чтобы на компьютере, для работы с которым они предназначены, присутствовал сервер регистрации, поддерживающий графический интерфейс. В качестве X-терминала может быть использован компьютер старой модели. На этом компьютере надо установить минимальные средства Linux и изменить конфигурационные файлы так, чтобы система обеспечивала регистрацию на удаленном компьютере.

Большинство дистрибутивных пакетов Linux позволяет после установки операционной системы на компьютер установить требуемую конфигурацию X-сервера. Этот сервер может использоваться для поддержки доступа локального компьютера к средствам отображения. Такая конфигурация позволяет одному и тому же компьютеру выступать в роли как клиента, так и сервера X Window, т. е. на компьютере, выступающем в роли X-сервера (рис. 14.1), могут выполняться свои X-программы. X-сервер в системе Linux может также применяться для взаимодействия с X-клиентами, выполняющимися на других компьютерах. Для того чтобы подобное взаимодействие стало возможным, надо выполнить несколько команд, которые будут описаны далее.

Для того чтобы компьютер выполнял роль X-клиента, дополнительное программное обеспечение не требуется; клиентами X Window являются сами программы, которые запускаются в системе. В большинстве случаев этим клиентам в процессе работы требуются различные графические библиотеки, например Qt или GTK+. При инсталляции программ в системах, базирующихся на RPM или Debian, вы получите информацию о том, какие библиотеки нужны им для работы. Теоретически вам нет необходимости устанавливать X-сервер на компьютере, выполняющем роль клиента, однако на практике проще сделать это, так как из-за взаимной зависимости пакетов вам все равно приходится инсталлировать большинство компонентов X Window. X-сервер представляет собой отдельную программу, обеспечивающую интерфейс с монитором, мышью, клавиатурой и другими устройствами. Наличие локального X-сервера упрощает проверку X-программ. Для того чтобы программные средства X-сервера вступили в действие, надо запустить X-программу либо локально, либо на удаленном компьютере.

Настройка X-сервера для взаимодействия с X-клиентом

Подобно другим типам серверов, сервер X Window отвечает на запросы клиентов. В большинстве дистрибутивных пакетов Linux по умолчанию не предусмотрена работа в качестве X-терминала; считается, что компьютер будет выполнять функции рабочей станции или сервера, поддерживающих другие протоколы. Поэтому, чтобы использовать компьютер под управлением Linux как X-сервер для программ, выполняющихся на других узлах сети, необходимо изменить конфигурацию системы. Сделать это можно двумя способами: с помощью программ `xhost` и `xauth`.

Использование программы xhost

Программа `xhost` позволяет указанному при ее вызове удаленному компьютеру обращаться к X-серверу. Для того чтобы воспользоваться данной программой, надо в окне `xterm` или другом окне, поддерживающем командную строку, ввести следующую команду:

```
$ xhost +biggie.threeroomco.com
```

В результате ее выполнения X-сервер получает указания о том, что он должен принимать обращения от компьютера `biggie.threeroomco.com`. Любой пользователь, работающий на этом компьютере, сможет использовать X-сервер для отображения окон, получать данные, введенные с клавиатуры, принимать информацию о перемещениях мыши и выполнять другие действия с удаленными системами отображения. Если вы не укажете имя узла (а ограничитесь лишь вводом команды `xhost +`), X-сервер будет принимать обращения от любого источника.

Большинство X-серверов для Windows, MacOS и других систем настроены так, что с ними может взаимодействовать любой узел сети. В системе Linux подобная конфигурация задается посредством команды `xhost +`.

Программа `xhost` проста и удобна в применении, но использовать ее можно лишь в той сети, в которой не требуются высокоуровневые средства защиты. Дело в том, что `xhost` не предпринимает попыток идентификации пользователей, работающих на удаленных компьютерах. Если доступ к серверу разрешен для узла сети, его автоматически получают все пользователи на этом узле. Любой пользователь удаленного компьютера может открывать окна на X-сервере и даже читать символы, вводимые с клавиатуры. Для ограничения доступа пользователей к X-серверу может быть использована программа `xauth`.

Использование программы xauth

Программа `xauth` обеспечивает средства аутентификации, скрытые от пользователя. Данная утилита автоматически применяется при регистрации в системе X Window, но при желании вы можете вызывать ее вручную. Несмотря на то что программа xauth менее удобна в работе, чем `xhost`, она обеспечивает более высокий уровень защиты.

В процессе работы `xauth` использует файл `.Xauthority`, расположенный в рабочем каталоге пользователя. Этот файл должен находиться и на клиентской машине, и на сервере. Если данный файл отсутствует, xauth автоматически создает его. В отличие от большинства конфигурационных файлов Linux, `.Xauthority` не является текстовым файлом. Для изменения его содержимого используется утилита xauth. С помощью `xauth` можно добавлять, удалять ключи и выполнять с ними другие необходимые действия. Некоторые методы регистрации на удаленном сервере предполагают автоматическую проверку содержимого `.Xauthority` и добавление необходимого ключа. X-сервер принимает обращения от любого клиента, который обладает соответствующим ключом. (Поскольку `.Xauthority` содержится в рабочем каталоге пользователя, ключ генерируется тогда, когда данный пользователь регистрируется на компьютере или запускает X-программу. Различным пользователям могут соответствовать различные файлы .Xauthority.) Чтобы X-клиент мог работать с X-сервером, необходимо скопировать ключ из пользовательского файла `.Xauthority` на сервере в файл `.Xauthority` на клиентской машине. При обращении к серверу клиент автоматически использует этот ключ. Процедура передачи ключа описана ниже.

1. На компьютере, на котором выполняется сервер X Window, введите команду `xauth`. При этом утилита xauth будет запущена от имени пользователя, который применит систему для взаимодействия с удаленным узлом. Несмотря на то что `xauth` формально является X-утилитой, она выполняется в текстовом режиме.

2. Введите команду `list`. При ее выполнении будет выведена информация о ключах, содержащихся в файле `.Xauthority`. Каждый ключ начинается с имени дисплея, которое представляет собой имя узла, а за ним следует номер дисплея, например `term.threeroomco.com:0`. Имена некоторых компьютеров сопровождаются символами `/unix`, кроме того, по команде `list` будут также выведены записи для `localhost`. Оба типа записей можно не принимать во внимание. В некоторых записях номера дисплеев будут отличаться от 0. Эти записи соответствуют второму, третьему и последующим сеансам работы с X-сервером, которые поддерживаются одновременно с первым сеансом. Вас интересует имя основного дисплея? Вероятнее

всего, оно будет состоять из имени вашего компьютера, за которым следует номер 0. После имени дисплея в строке будут отображаться также тип кодировки (например, MIT-MAGIC-COOKIE-1) и 32-байтовое шестнадцатеричное число. Несмотря на то что эти данные предназначены для передачи, их можно не учитывать.

3. Введите команду extract *имя_файла имя-дисплея*. Здесь имя файла может быть любым, а имя дисплея — это имя, которое вы выяснили на предыдущем шаге процедуры. Например, вы можете задать команду extract xfer-auth term.threeroomco.com:0. В результате запись файла .Xauthority для дисплея будет скопирована в указанный файл. Файл используется для передачи ключа на клиентский компьютер.

4. Введите команду exit, чтобы завершить работу с программой xauth.

5. Скопируйте файл, созданный при выполнении команды extract, да клиентский компьютер (удаленный компьютер, на котором расположена программа, предназначенная для выполнения). Сделать это можно различными способами: использовать средства FTP или NFS, перенести файл на дискете и т. д.

6. Зарегистрируйтесь на компьютере, выполняющем функции X-клиента.

7. Задайте команду xauth, чтобы запустить утилиту xauth на клиентской машине.

8. Введите команду merge *имя_файла*. В этой команде должно быть указано имя файла, которое вы сгенерировали посредством команды extract и скопировали на клиентский компьютер. (Возможно, вам придется указать путь к файлу.)

9. Задайте команду list. Данная команда, помимо прочих сведений, должна отобразить запись для X-сервера, которую вы только что включили. Если такая запись отсутствует, это значит, что какие-то из предшествующих действий были выполнены неправильно.

10. Введите команду exit, чтобы завершить работу xauth и сохранить внесенные изменения. (Заметьте, что в xauth также предусмотрена команда quit, которая не сохраняет изменения. Команду quit надо использовать в том случае, если при выполнении данной процедуры были допущены ошибки.)

Если на обоих компьютерах инсталлированы средства SSH, вы можете вместо описанной выше процедуры выполнить единственную команду.

```
# xauth list x_сервер :0 | sed -e 's/^ /add /' | ssh \
  x_клиент -x xauth
```

В данном случае xauth вызывается в командной строке, sed используется для включения команды add в начало выходных данных, кроме того, утилита xauth запускается также на стороне X-клиента. При вызове данной команды необходимо учитывать следующее.

- Вместо *x_сервер* надо указать имя компьютера, за которым вы работаете, а вместо *x_клиент* — имя компьютера, на котором должна выполняться клиент-программа.

- Между add и последующей косой чертой (/) должен быть пробел. Эта команда передается утилите **xauth** на клиентском компьютере, и пробел должен отделять add от имени дисплея.

- Если конфигурация SSH предполагает ввод пароля либо фразы пароля, вам придется ввести соответствующие данные.

С этого момента X-сервер будет принимать обращения от X-клиентов, но, чтобы эти программы могли работать совместно, вам придется установить на клиентском компьютере опцию, позволяющую взаимодействовать с X-сервером (этот вопрос будет подробнее рассмотрен в следующем разделе). При установлении соединения клиент X Window обратится к файлу **.Xauthority** за ключом, соответствующим серверу.

Поскольку работа xauth основана на применении ключа, который известен только серверу и авторизованному клиенту, она обеспечивает более высокий уровень защиты, чем xhost. Кроме того, при использовании xauth доступ к серверу предоставляется только отдельным пользователям. X-сервер становится более устойчивым к атакам, осуществляемым путем подмены IP-адреса. Недостатком данного способа является передача ключей в незакодированном виде. Если локальная сеть не обеспечивает безопасность передаваемых данных либо если клиент с сервером взаимодействуют по Internet, ключ может быть похищен и злоумышленник получит доступ к X-серверу. Если при обмене данными в системе X Window необходимо обеспечить высокий уровень защиты, надо использовать SSH-соединение. Вопросы поддержки X-взаимодействия посредством SSH будут подробно рассмотрены в следующем разделе.

Не все X-серверы сконфигурированы для работы с xauth. Соответствующая опция обычно устанавливается в том случае, когда X-сервер запускается посредством XDM, GDM или KDM. Если вы запускаете X-сервер с помощью **startx**, поддержка xauth в ряде систем будет отсутствовать. В некоторых случаях вам придется отредактировать сценарий **startx** (он обычно располагается в каталоге **/usr/X11R6/bin**) так, чтобы в нем присутствовала опция **-auth** *файл_авторизации*; в качестве файла авторизации обычно указывается файл **.Xauthority**, находящийся в рабочем каталоге. Часто в редактировании **startx** нет необходимости.

Настройка X-клиента для работы с X-сервером

Независимо от того, используете ли вы xhost или xauth, вы должны сконфигурировать клиентскую систему для работы с нужным X-сервером. Если, например, вы работаете за компьютером **term.threeroomco.com**, зарегистрировались на узле **biggie.threeroomco.com** и хотите, чтобы программа использовала компьютер **wrongone.threeroomco.com** в качестве X-терминала, вам не удастся сделать это. По умолчанию многие версии Linux сконфигурированы так, что даже если пользователь зарегистрировался с внешнего узла, они будут работать с локальным X-сервером.

При запуске X-программа читает значение переменной окружения **DISPLAY** и определяет, какой X-сервер следует использовать. Чтобы определить текущее значение этой переменной, надо на компьютере, выполняющем роль X-клиента, вызвать следующую команду:

```
$ echo $DISPLAY
```

```
biggie.threeroomco.com:0.0
```

Если отображаемая с помощью этой команды строка (в данном случае `biggie.threeroomco.com:0.0`) соответствует вашему серверу, вам не надо предпринимать никаких действий. (Первый дисплей обычно имеет номер 0 или 0.0; эти два значения эквивалентны.) Если же значение переменной DISPLAY указывает на X-клиент или другую систему либо если оно вовсе не определено, вам надо задать новое значение данной переменной. Необходимая для этого команда выглядит следующим образом:

```
$ export DISPLAY=term.threeroomco.com:0
```

Очевидно, что имя узла должно определять ваш X-сервер. При последующих запусках X-программа будет пытаться взаимодействовать с указанным сервером. Чтобы эти попытки были успешными, вам надо настроить X-сервер для работы с X-клиентом, т. е. запустить программу `xhost`, или создать запись `xauth` для клиента.

Туннелирование X-соединений через SSH

Из материала, рассмотренного ранее в данной главе, следует, что для инициализации X-соединения используются два отдельных, независимых друг от друга протокола. Во-первых, работая на компьютере, выполняющем роль X-сервера, вы используете клиент-программу удаленной регистрации, работающую в текстовом режиме, например программу `telnet`. Во-вторых, после установления соединения вы инициируете взаимодействие X-клиента с X-сервером. Выполнив основные действия по установке соединений, вы можете вызвать на своем компьютере любую команду, например `xclock`, и соответствующая программа (в данном случае `xclock`) будет выполняться на удаленном компьютере. Данная конфигурация подходит для решения многих задач, но при ее использовании могут возникать проблемы. Одна из этих проблем связана с тем, что в сеансе X-взаимодействия данные передаются в незакодированном виде и могут быть перехвачены. Тот факт, что на каждом из взаимодействующих компьютеров должен выполняться сервер, также можно считать недостатком. Если эти компьютеры разделены брандмауэром или маршрутизатором, осуществляющим маскировку пакетов, эти программы должны быть сконфигурированы специальным образом, иначе обмен по протоколу X Window станет невозможным. Одно из возможных решений обеих проблем состоит в использовании протокола SSH. Этот протокол может применяться как для установления начального соединения между X-клиентом и X-сервером, так и для туннелирования данных, передаваемых в рамках этого соединения.

Основные вопросы, связанные с настройкой и использованием SSH, рассматривались в главе 13. Для того чтобы туннелирование протокола X Window посредством SSH стало возможным, надо соответствующим образом сконфигурировать средства поддержки SSH.

- В конфигурационном файле `/etc/ssh/ssh_config` клиентской программы SSH (эта программа выполняется на том компьютере, на котором расположен X-сервер) следует задать значение yes опции ForwardXll. Аналогичный результат можно получить, указав при запуске ssh опцию -X. (Обратите внимание на регистр символа; если в опции -x вы зададите символ нижнего регистра, туннелирование будет запрещено.)

- В файле `/etc/ssh/sshd_config` на компьютере, на котором выполняется SSH-сервер (эта машина играет роль клиента в X-взаимодействии), опция

X11Forwarding должна иметь значение yes. Эта опция сообщает серверу SSH о том, что локальные вызовы X-сервера должны перехватываться и направляться SSH-клиенту.

При туннелировании X-соединения сервер SSH "подменяет" локальный X-сервер. Если конфигурация установлена правильно, средства поддержки SSH устанавливают переменную окружения DISPLAY таким образом, что X программы передают данные через порт локального X-сервера (по умолчанию это X-сервер 10, или TCP-порт 6010). С этим портом связывается сервер SSH. Вместо того чтобы отображать информацию на локальной машине, сервер SSH кодирует ее и передает клиенту SSH. Клиент, в свою очередь, запрашивает локальный X-сервер (он определяется значением переменной окружения DISPLAY на этом компьютере), а данные, полученные в результате этого обращения, передает серверу SSH, который доставляет информацию X-клиенту. Таким образом, сервер SSH "выдает себя" за X-сервер, а клиент SSH "подменяет" клиентскую программу X Window.

Такой подход имеет ряд преимуществ по сравнению с обычным X-обменом. Для взаимодействия двух компьютеров используется лишь одно соединение, что упрощает настройку сетевых средств. Если вы хотите спрятать X-сервер за брандмауэром или маршрутизатором, выполняющим NAT-преобразование, это проще сделать при работе посредством SSH, чем в случае, когда используется Telnet или другой протокол удаленной регистрации. Кроме того, протокол SSH предполагает кодирование передаваемых данных. Поэтому если информация будет перехвачена по пути от одного компьютера к другому, использовать ее вряд ли удастся. Туннелирование X-соединения средствами SSH имеет один недостаток. Как известно, для кодирования информации приходится выполнять большой объем вычислений, поэтому использование SSH создает дополнительную нагрузку на процессоры компьютеров на обоих концах соединения. В результате скорость обмена X-клиента с X-сервером снижается. Замедление работы вследствие использования SSH заметно при скорости процессора около 200 МГц, однако во многих случаях повышение уровня защиты оправдывает уменьшение производительности компьютеров. Чтобы снизить требования к пропускной способности линий, можно использовать сжатие информации. С другой стороны, сжатие данных создает дополнительную нагрузку на процессор, что может еще больше уменьшить скорость обмена. Желательно испробовать обмен данными со сжатием и без сжатия и экспериментально определить, какой режим более благоприятен для компьютера и сетевых средств.

Сказанное выше предполагает, что на обоих концах соединения используются компьютеры под управлением Linux или UNIX. Если X-сервер выполняется в среде Windows, MacOS, OS/2 или в другой операционной системе, следует выяснить, поддерживает ли клиент SSH туннелирование X-соединений. Если клиент SSH предоставляет такую возможность, вам надо уметь активизировать данные средства. Подробную информацию вы получите, прочитав документацию на программы поддержки SSH.

Основные действия по организации X-взаимодействия

В данной главе были рассмотрены самые разнообразные средства установления X-соединения. В ваше распоряжение предоставляется настолько много возможностей, что разобраться с ними бывает достаточно трудно. Ниже описаны действия по установлению типичного соединения. Это описание является своеобразным итогом предыдущего обсуждения.

1. **Запуск X-сервера.** Если вы используете систему Linux, средства X-регистрации могут быть предусмотрены при загрузке. Некоторые конфигурации системы предполагают запуск X-сервера по команде `startx`. В Windows, MacOS и других средах X-сервер надо запускать вручную либо настраивать операционную систему для автоматического запуска соответствующей программы.

2. **Настройка X-сервера для установления соединения.** Для того чтобы X-взаимодействие могло осуществляться, необходимо сообщить серверу X Window о том, что он должен обрабатывать запросы от удаленных компьютеров на установление соединений. Это можно сделать, запуская программу `xhost` на компьютере, на котором установлен X-сервер, либо передавая ключ `xauth` клиентской системе. Если вы используете SSH для туннелирования соединения, действия, выполняемые на этом шаге, не обязательны, но при этом вам необходимо сконфигурировать клиент SSH и сервер SSH.

3. **Установление соединения с X-клиентом.** Для соединения с компьютером, на котором выполняется X-клиент, вы можете использовать любой протокол удаленного доступа, например Telnet или SSH. Заметьте, что на удаленном компьютере выполняется сервер удаленной регистрации и клиент X Window.

4. **Настройка X-клиента для работы с требуемым X-сервером.** Чтобы определить, какой компьютер должен использоваться в качестве X-сервера, X-клиент использует переменную окружения `DISPLAY`. В некоторых системах значение этой переменной устанавливается автоматически, в остальных случаях вы должны сделать это самостоятельно, вызывая команду наподобие следующей: `export DISPLAY=term.threeroomco.com:0`.

5. **Запуск X-программы.** Для того чтобы запустить X-программу, достаточно ввести ее имя в окне, посредством которого вы осуществляли удаленную регистрацию. Например, если вы регистрировались в окне `xterm`, в нем же следует запускать требуемую программу.

В зависимости от способа соединения и аутентификации, некоторые стадии данной процедуры, например этапы 2 и 4, могут быть пропущены. В системах Windows и MacOS ряд действий выполняется автоматически. Например, в состав некоторых X-серверов входят минимальные средства удаленной регистрации с использованием Telnet или других протоколов. Эти средства автоматически вызываются при выводе xterm. Если при настройке подобного сервера были указаны пользовательское имя и пароль, то после щелчка на соответствующей кнопке будет запускаться X-сервер и отображаться окно `xterm`. Подробные сведения о каждом типе X-сервера можно найти в документации на него.

В результате выполнения описанной выше процедуры программа на удаленном компьютере выводит свои данные на экране X-сервера. Как правило, это происходит в окне, которое X-сервер открывает для отображения рабочего стола клиента. Обычно в локальной системе выполняется диспетчер окон. Если же вы хотите, чтобы на локальном компьютере поддерживались минимальные системные средства, а диспетчер окон и окружение рабочего стола реализовывались на удаленной машине, вам нужно соответствующим образом изменить стартовые сценарии X Window на локальном и удаленном компьютерах. Еще один подход состоит в использовании удаленного X-сервера регистрации;

в этом случае основная нагрузка по поддержке окон и среды рабочего стола ложится на удаленную систему.

Использование сервера XMCP

Если в сети не используется брандмауэр или маскирующий маршрутизатор, способный повлиять на обмен данными между клиентом и сервером, то каждый компьютер, на котором выполняется X-сервер, может быть использован как дисплей для любого X-приложения. Однако в некоторых случаях этого недостаточно; необходимо, чтобы X-сервер работал как локальный сервер удаленной системы, отображая ее окружение. Процедура регистрации, использующая Telnet, SSH или другой протокол удаленной регистрации, не всегда удобна для управления удаленной системой. Часто бывает удобнее работать с протоколом регистрации, входящим в систему X Window. Таким протоколом является XDMCP (X Display Manager Control Protocol — протокол управления диспетчером X-отображения). В состав большинства систем Linux входит программное обеспечение, необходимое для организации работы сервера XDMCP (сервер XDMCP располагается на том же компьютере, что и X-клиент), но, как правило, эти системы сконфигурированы так, что доступ к серверу XDMCP ограничен локальной системой. Если вы измените конфигурацию сервера, он будет обслуживать различные клиенты XDMCP (они располагаются на тех же компьютерах, что и X-серверы). Компьютер под управлением Linux можно также использовать как клиент XDMCP, но для этого надо модифицировать X-конфигурацию так, чтобы средства поддержки XDMCP представляли окно регистрации, генерируемое на удаленном компьютере.

Принцип действия XDMCP

В предыдущих разделах был рассмотрен принцип использования X-соединений, предполагающий применение протокола удаленной регистрации, например Telnet. Сервер Telnet располагался на том же компьютере, что и X-клиент; с его помощью пользователь регистрировался на удаленном компьютере, после чего устанавливалось X-соединение между клиентом и сервером. Сервер XDMCP успешно заменяет Telnet, SSH и другие протоколы удаленной регистрации, обеспечивающие взаимодействие в текстовом режиме. Когда пользователь обращается к удаленной системе с помощью Telnet, сервер Telnet регистрирует его и предоставляет текстовую оболочку. Средства XDMCP действуют подобным образом, но вместо текстовой оболочки сервер XDMCP инициирует процедуру регистрации X Window; при этом отображается окно для ввода пользовательского имени и пароля и загружается диспетчер окон, среда рабочего стола и другие компоненты графического интерфейса. Эти программы запускаются посредством X-соединения; XDMCP автоматически конфигурирует клиентскую и серверную программы, используя `xauth`. Другими словами, XDMCP автоматизирует описанную ранее процедуру регистрации.

Сервер XDMCP применяется не только для установления X-соединений. Он также обеспечивает обращение системы Linux к X-серверу при загрузке. Отображаемое при этом окно регистрации показано на рис. 14.2. Внешний вид окна зависит от конфигурации сервера XDMCP.

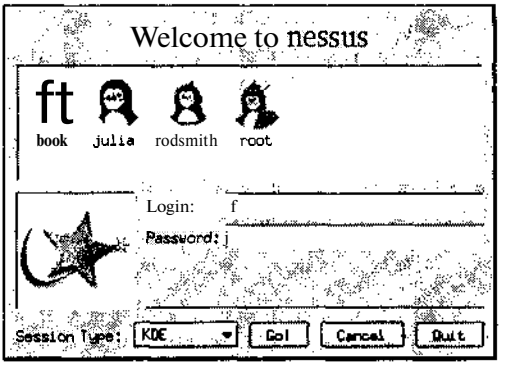


Рис. 14.2. В окне регистрации указываются пользовательское имя и пароль. В некоторых случаях пользователь может задать дополнительную информацию, например, указать среду рабочего стола и диспетчер окон

Настройка сервера регистрации для установления соединения

Для настройки сервера XDMCP используются конфигурационные файлы, которые обычно находятся в каталогах `/etc` и `/etc/X11`. Исходя из сообщений защиты, большинство дистрибутивных пакетов сконфигурировано так, что обращения к серверу возможны только с локального узла. Чтобы разрешить удаленную XDMCP-регистрацию, необходимо снять данные ограничения. Кроме того, надо обеспечить выполнение сервера XDMCP. В настоящее время в системе Linux используются три сервера XDMCP: X Display Manager (XDM) и более новые продукты — KDM (KDE Display Manager) и GDM (GNOME Display Manager).

Настройка XDM

XDM — наиболее простой среди серверов XDMCP; он был разработан раньше других продуктов подобного назначения. В отличие от GDM и KDM, XDM не связан с окружением рабочего стола Linux. Он дает возможность пользователям вводить имена и пароли, но не позволяет указывать дополнительные сведения, например задавать диспетчер окон. Для управления регистрационными параметрами пользователя служит файл `.xsession`, расположенный в его рабочем каталоге. (Данный сценарий запускается из сценария `Xsession`, который располагается в каталоге `/etc/X11` или `/etc/X11/xdm`.) Файл `.xsession`, соответствующий конкретному пользователю, обычно заканчивается строкой, которая запускает диспетчер окон или среду рабочего стола. По завершении сценария (это происходит, когда пользователь выходит из системы, прекращая тем самым работу диспетчера окон) завершается X-сеанс и XDM прекращает удаленное взаимодействие, либо при использовании локального дисплея XDM повторно отображает приглашение для регистрации.

Обеспечение доступа к XDM

Доступом удаленных компьютеров к XDM управляет основной конфигурационный файл `/etc/X11/xdm/xdm-config`. В большинстве дистрибутивных пакетов в срставе этого файла содержится следующая строка:

```
DisplayManager.requestPort: 0
```

Данная запись указывает XDM на то, что он не должен принимать обращения через UDP-порт 177. Если вы хотите, чтобы пользователи с других компьютеров могли регистрироваться посредством XDMCP, вам надо закомментировать данную строку (включить в ее начало символ #).

Помимо редактирования файла `xdm-config`, вам, возможно придется внести изменения в файл `/etc/X11/xdm/Xaccess`. В этом файле указаны компьютеры, которым разрешен доступ к серверу XDM. Данный файл состоит из набора записей, каждая из которых занимает одну строку. В составе записи указываются имя узла и тип доступа, разрешенный для него. (Символ # в начале строки является признаком комментариев.) Если тип доступа не указан, клиенту разрешены непосредственные обращения к серверу. Значение CHOOSER определяет действия сервера при получении косвенного запроса от клиента, а значение BROADCAST, которое чаще всего используется в сочетании с CHOOSER, указывает на то, что при получении косвенного запроса он должен передаваться остальным серверам XDMCP в широковещательном режиме. Если вместо имени узла указан символ *, это означает, что любой клиент имеет право устанавливать соединение с сервером XDM. Например, приведенные ниже записи позволяют любому клиенту обращаться к серверу непосредственно или использовать его для передачи косвенного запроса.

```
*
* CHOOSER BROADCAST
```

Если вы хотите, чтобы право обращаться к серверу имели лишь некоторые узлы сети, вам надо вместо символа * задавать конкретные имена компьютеров. Данный символ может использоваться в составе имени; в этом случае право доступа к серверу получают все компьютеры, принадлежащие к указанному домену. Например, приведенные ниже записи разрешают доступ к серверу всем компьютерам домена `threeroomco.com`, а также двум внешним узлам. Кроме того, указанные здесь внешние узлы имеют право передавать косвенные запросы.

```
*.threeroomco.com
bronto.pangaea.edu
stego.pangaea.edu
bronto.pangaea.edu CHOOSER BROADCAST
stego.pangaea.edu CHOOSER BROADCAST
```

Ограничить доступ к XDMCP можно не только средствами, предусмотренными в конфигурационном файле XDM. Блокировать обращения к серверу может также брандмауэр.

Управление отображением

В файле `/etc/X11/xdm/Xservers` указывается список устройств, которыми может управлять сервер XDM. При запуске XDM пытается непосредственно обратиться к этим дисплеям и вывести окно регистрации. По умолчанию в данный файл помещается

следующая строка (в зависимости от особенностей системы конкретная запись может несколько отличаться от приведенной ниже):

```
:0 local /usr/X11R6/bin/X
```

Данная запись указывает на то, что сервер должен управлять локальным дисплеем (:0). При необходимости XDM запускает локальный X-сервер. Если вы хотите, чтобы XDM непосредственно управлял отображением на удаленных машинах, без использования окна регистрации, вы должны указать требуемые системы следующим образом:

```
term.threeroomco.com:0 foreign
```

В этом примере `foreign` указывает на то, что заданная система является удаленной. Система может быть сконфигурирована так, что соединение с сервером XDMCP и отображение окна регистрации будет разрешено. Редактируя файл Xservers, вы можете также удалить включенную по умолчанию запись local. Если вы сделаете это, компьютер не будет загружать локальный X-сервер при запуске XDM. Такая конфигурация может потребоваться, если вы организуете работу с мощным центральным компьютером через X-терминалы. В таком случае на компьютере будет выполняться много X-программ, но ни одна из них не будет взаимодействовать с локальным X-сервером.

Задавать управление устройством отображения следует в том случае, если X-сервер должен взаимодействовать только с компьютером, на котором выполняется XDM. В частности, данный подход применяется тогда, когда X-терминал соединяется лишь с одним узлом сети.

Настройка KDM

Сервер KDM был разработан для замены существующего XDM. По сравнению со своим предшественником, KDM предоставляет дополнительные возможности, наиболее важными из которых являются список Session Type, в котором пользователи могут выбирать диспетчер окон или среду рабочего стола, и кнопка Quit (или Shutdown), позволяющая завершать выполнение локального X-сервера (при удаленном выполнении) или останавливать работу всей системы (при локальном запуске). Окно регистрации, отображаемое посредством KDM, показано на рис. 14.2.

KDM использует те же конфигурационные файлы, что и XDM, поэтому рекомендации по обеспечению удаленного доступа, изложенные в предыдущем разделе, подходят и для KDM. Кроме того, в KDM предусмотрены дополнительные средства настройки по сравнению с XDM. Соответствующие опции указываются в файле `kdmrc`, расположение которого зависит от дистрибутивного пакета. Чаще всего он находится в каталоге `/opt/kde2/share/config` или `/usr/share/config`. В этом файле содержатся средства управления размером окна регистрации, стилем интерфейса и другими параметрами отображения данных. Одной из наиболее важных является опция `SessionTypes`. Она определяет тип пользовательского сеанса, в частности диспетчеры окон для выбора. Если вы добавляете сеанс в список, вы также должны учесть его в файле Xsession или `Xsession.d`, находящемся в каталоге `/etc/X11` или `/etc/X11/xdm`. К сожалению, разобраться в структуре файла достаточно трудно, кроме того, эта структура изменяется в зависимости от дистрибутивного пакета. Для добавления сеанса служит переменная SESSION либо другая переменная с подобным именем. В состав некоторых пакетов входит инструмент chksession, который автоматически учитывает диспетчер окон или среду рабочего стола в конфигурации KDM или GDM. Чтобы это стало возможным,

диспетчер окон или среда рабочего стола должны поставляться с соответствующими конфигурационными файлами. В большинстве случаев пользователю проще настроить окружение, отредактировав файл .xsession, находящийся в его рабочем каталоге. Для того чтобы система использовала этот файл, пользователь должен выбрать при работе с KDM специальный пункт под названием Default.

Настройка GDM

Подобно KDM, сервер GDM предоставляет пользователям возможность выбрать окружение рабочего стола, а также завершить работу локального компьютера или сеанс удаленного взаимодействия. Однако, в отличие от KDM, GDM использует собственные конфигурационные файлы, которые обычно хранятся в каталоге /etc/X11/gdm. Наиболее важным из них является файл gdm.conf.

Как и большинство систем, использующих серверы XDMCP, системы, включающие GDM, конфигурируются так, чтобы пользователи не могли регистрироваться с удаленных узлов. Чтобы отказаться от этого ограничения, надо изменить одну или две записи в разделе [xdmcp] файла gdm.conf. Строку Enable=0 следует заменить на Enable=1. Если вы хотите, чтобы GDM предоставлял X-терминалу список других компьютеров, работающих с XDMCP, вам надо заменить HonorIndirect=0 на HonorIndirect=1.

Если вы хотите, чтобы GDM был доступен для удаленных узлов, а локальный X-сервер не запускался, вам надо закомментировать строку, соответствующую локальным серверам в разделе [servers]. Как правило, в этом разделе находится следующая запись:

```
0=/usr/bin/X11/X
```

Эта запись указывает GDM на то, что X-сервер (программа /usr/bin/X11/X) должен быть запущен для управления первым X-сеансом. Если в начало этой строки будет включен символ комментариев, GDM не будет управлять локальным устройством отображения или запускать X-сервер.

Подобно KDM, GDM позволяет пользователю выбрать диспетчер окон или окружение рабочего стола. (В GDM для этого предназначено меню Session.) Для добавления или удаления сеансов надо создать соответствующие сценарии в каталоге /etc/Xll/gdm/Sessions. По умолчанию обычно используется сценарий /etc/X11/xdm/Xsession. Вам необходимо отредактировать данный сценарий или создать новый, выполняющий подобные действия, реализовав в нем возможность загрузки другого диспетчера окон или окружения рабочего стола. В большинстве систем пользователь может настраивать среду посредством редактирования файла .xsession.

Запуск сервера XDMCP

Чтобы обеспечить запуск сервера XDMCP, надо настроить компьютер для работы с X Window и приема запросов на регистрацию. В большинстве дистрибутивных версий для этой цели зарезервирован уровень выполнения 5, но в некоторых случаях используются и другие уровни. Например, в версиях SuSE, предшествующих версии 7.2, регистрация с применением графического интерфейса производится на уровне 3, а в Slackware для этого используется уровень 4. В Debian и системах, созданных на ее основе, средства X Window выполняются на всех уровнях, допускающих работу нескольких пользователей.

Уровень по умолчанию задается с помощью специальной записи в файле /etc/inittab, которая выглядит приблизительно следующим образом:

```
id:5:initdefault:
```

Во многих дистрибутивных пакетах данной записи предшествуют комментарии, объясняющие ее назначение. Номер уровня содержится во втором поле; именно на этот уровень система переходит после загрузки. Если на компьютере установлены средства X Window, сервер XDMCP также будет загружен.

Для изменения уровня выполнения служит утилита `telinit`. Например, по команде `telinit 5` происходит переход на уровень 5. Система будет находиться на указанном уровне до следующего вызова `telinit` либо до перезагрузки компьютера.

Если вы вносите изменения в конфигурацию сервера XDMCP, вам надо обеспечить, чтобы новая конфигурация была учтена при работе сервера. Сделать это вы можете, перейдя на уровень, обеспечивающий работу только в текстовом режиме, а затем вернувшись на уровень, допускающий выполнение X-программ. Для перехода на другой уровень используется утилита `telinit`. Кроме того, вы можете остановить работу сервера XDMCP, вызвав команду `kill` или `killall`, а затем запустить сервер снова. Для того чтобы сервер XDMCP повторно прочитал содержимое конфигурационных файлов, ему можно передать сигнал `SIGHUP`; в этом случае завершать работу сервера нет необходимости.

В каждом дистрибутивном пакете используется свой сервер XDMCP, но при желании вы можете переконфигурировать систему для работы с нужным вам сервером. Средства для выбора сервера XDMCP, используемые в различных версиях Linux, описаны ниже.

- **prefdm.** В некоторых дистрибутивных пакетах Linux, например, в системах Red Hat и Mandrake, для загрузки сервера XDMCP применяется сценарий с именем prefdm (он находится в каталоге `/etc/X11`). Для выбора среды рабочего стола и сервера XDMCP данный сценарий читает файл `/etc/sysconfig/desktop`. Обычно в этом файле содержатся значения KDE, GNOME и AnotherLevel, которые задают в качестве XDMCP-сервера соответственно KDM, GDM и XDM.

- **Сценарии запуска** SysV. В Debian и системах, созданных на ее основе, запуск сервера XDMCP осуществляется посредством сценария SysV, например `/etc/init.d/xdm`. Заменив или отредактировав этот файл, вы можете задать использование другого сервера XDMCP. Аналогичный способ применяется в системе SuSE, но тип сервера XDMCP, запускаемого с помощью сценария `xdm`, определяет значение переменной окружения `DISPLAYMANAGER`, которое задается в файле `/etc/rc.config`.

- **Прочие сценарии запуска.** Для запуска сервера XDMCP в системе Slackware применяется сценарий `/etc/rc.d/rc.4`. Как было сказано в главе 4, в Slackware в явном виде не используется механизм уровней выполнения, но сценарий `rc.4` выполняет те же функции, что и сценарий xdm в системах Debian и SuSE. В Caldera применяется тот же подход, но сценарий запуска называется `/etc/rc.d/rc.gui`. Код сценария для Slackware составлен так, что сначала предпринимается попытка запустить KDM, затем GDM, а потом XDM. Сценарий в системе Caldera запускает только KDM. Отредактировав код сценария, вы можете изменить порядок вызова серверов.

Настройка клиента удаленной регистрации

Подобно другим серверам, XDMCP-сервер сам по себе абсолютно бесполезен; его использование имеет смысл только тогда, когда он взаимодействует с одним или несколькими клиентами. Как правило, клиенты XDMCP встраиваются в состав X-серверов. Клиент XDMCP может либо непосредственно взаимодействовать с сервером XDMCP, либо предоставлять список доступных X-серверов. (X-сервер для Windows, отображающий список компьютеров, показан на рис. 14.3.) Если вы выберете компьютер и щелкнете на кнопке Connect (или активизируете другой интерактивный элемент аналогичного назначения), вы увидите окно регистрации, представленное на рис. 14.2. После окончания регистрации X-сервер отобразит рабочий стол компьютера. В зависимости от конфигурации X-сервера, изображение рабочего стола либо займет весь экран либо будет выведено в отдельном окне.

Большинство X-серверов, предназначенных для работы в системе Windows или MacOS, предоставляют диалоговое окно, которое позволяет задать особенности выполнения операций XDMCP. Диалоговое окно, предоставляемое одним из X-серверов, выполняющихся в системе Windows, показано на рис. 14.4. Основными элементами являются переключатели опций, расположенйые в верхней части окна. С их помощью задается способ, которым клиент XDMCP, встроенный в X-сервер, устанавливает соединение с сервером XDMCP. В различных программах имена опций могут различаться. Назначение этих опций описано ниже.

- Do Not Use XDM (Passive). Данная опция предполагает, что соединение с X-сервером устанавливается вручную с использованием Telnet или другого инструмента регистрации либо что сервер XDMCP настроен для управления отображением на X-сервере (это можно сделать с помощью записи `foreign`, включаемой в файл `/etc/X11/xdm/Xservers`). В последнем случае сервер XDMCP отобразит на экране компьютера, выполняющего функции X-сервера, окно регистрации. Если вы перезапустите X-сервер, окно регистрации исчезнет и отобразится только после перезапуска сервера XDMCP.

- XDM Query. Если задана эта опция, X-сервер передает запрос на регистрацию тому узлу, имя или IP-адрес которого вы зададите. Если на указанном компьютере выполняется сервер XDMCP, будет выведено окно регистрации, подобное изобра-

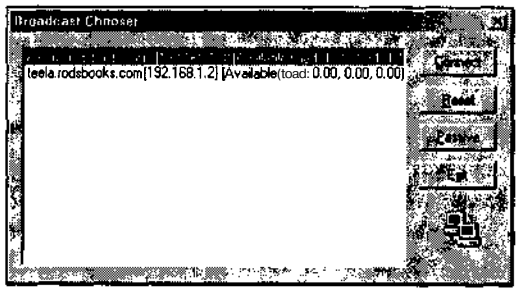


Рис. 14.3. Выбрав сервер XDMCP из списка, вы можете запустить X-программу на этом компьютере

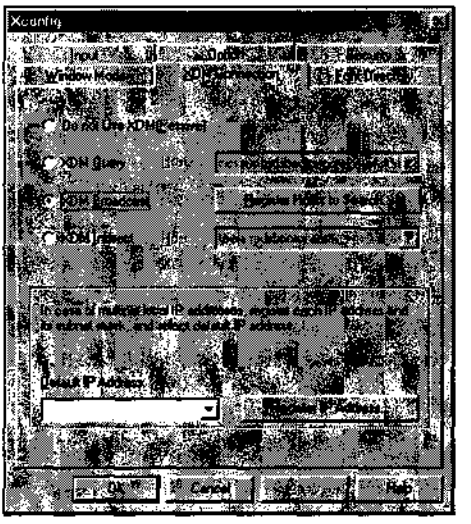


Рис. 14.4. Большинство клиентов XDMCP позволяют выбирать способ взаимодействия с серверами XDMCP

женному на рис. 14.2. Используя данную опцию, вы не можете непосредственно регистрироваться на другом компьютере. XDM Query заставляет X-сервер при каждом запуске передавать запрос серверу XDMCP. Такое поведение более приемлемо для пользователя, чем ситуация, когда сервер XDMCP управляет X-сервером.

- XDM Broadcast. Данную опцию лучше всего задавать, когда в локальной сети выполняется несколько X-серверов. В этом случае X-сервер передает широковещательный запрос по сети, определяет расположение всех серверов XDMCP и выводит их список, как показано на рис. 14.3. Некоторые серверы позволяют ограничить широковещательный запрос определенными адресами (для этого предназначена кнопка Register Hosts to Search, показанная на рис. 14.4).

- XDM Indirect. Данную опцию удобно применять, если вы хотите, чтобы пользователи могли регистрироваться на одном из нескольких компьютеров, принадлежащих внешней сети. Введите имя или IP-адрес сервера XDMCP в сети; в результате X-сервер обратится к этой системе для получения списка серверов. В данном случае сервер XDMCP должен быть сконфигурирован для обработки косвенных запросов.

Способностью отображать список доступных серверов XDMCP обладают не только X-серверы, предназначенные для выполнения в системе Windows. То же самое может делать XFree86, предназначенный для Linux. Режим работы данного продукта определяется опциями, задаваемыми при его запуске. Вы можете указывать опции `-query` *`имя_узла`*, `-broadcast` и `-indirect`. Пример вызова X-сервера приведен ниже.

```
$ /usr/X11R6/bin/X -indirect xdmcp-server.threeroomco.com
```

Приведенные выше опции действуют так же, как и опции X-сервера для Windows, за одним исключением. Опция `-broadcast` не приводит к отображению списка доступных

узлов; клиент устанавливает соединение с первым сервером XDMCP, который отвечает на запрос.

При желании вы можете сконфигурировать компьютер под управлением Linux как выделенный X-терминал. Конфигурацию следует задать так, чтобы X-сервер не запускался автоматически посредством сервера XDMCP. Затем следует создать сценарий запуска, который вызывал бы X-сервер с указанием требуемой опции: `-query`, `-broadcast` или `-indirect`. Если вы хотите отобразить список доступных локальных серверов, вам надо сконфигурировать один из серверов XDMCP так, чтобы он обрабатывал косвенные запросы, и указать при запуске X-сервера опцию `-indirect`. Таким образом, можно реализовать X-терминал даже посредством компьютера с процессором 386.

Обеспечение удаленного доступа с помощью сервера VNC

Средства X Window очень часто применяются для создания графической среды в системе Linux. Поскольку система X Window непосредственно разрабатывалась для работы в сети, ее очень удобно использовать в качестве инструмента удаленной регистрации. Однако X Window — не единственный инструмент, предоставляющий графический интерфейс для взаимодействия с удаленным компьютером. Эту задачу решают также средства VNC. Сетевая модель VNC отличается от X Window, и ей присущи свои преимущества и недостатки. Процедура инсталляции и настройки сервера VNC отличается от установки и выбора конфигурации X-сервера. Клиенты этих систем также различаются между собой.

Взаимодействие клиента и сервера VNC

Излагая материал данной главы, я пытался обратить внимание на роль различных клиентских и серверных программ в процессе сетевого взаимодействия. Как вы уже знаете, один и тот же компьютер может действовать как сервер по одному протоколу и выполнять функции клиента по другому протоколу. Рассматривая VNC, понять принцип взаимодействия программ несколько проще, так как в этом случае компьютер, за которым работает пользователь, является клиентом VNC, а удаленный компьютер — сервером VNC.

Если вы вспомните недавнее обсуждение принципов работы X Window, вам, наверное, покажется странным, как клиент VNC может обеспечивать работу пользовательского компьютера. Если X-сервер управляет клавиатурой, мышью и дисплеем, то как пользователь, не работающий за X-сервером, может выполнять X-программы? Дело в том, что VNC предполагает дополнительный уровень сетевого взаимодействия, скрытый от пользователя. На узле сети, выполняющем роль сервера VNC, присутствует X-клиент, который взаимодействует с X-сервером локального компьютера. X-сервер обменивается данными с сервером VNC так, как будто X-сервер работает с реальными устройствами ввода-вывода, однако вместо локальной клавиатуры, мыши и дисплея сервер VNC взаимодействует по сети с удаленным клиентом VNC, который поддерживает обмен с устройствами ввода-вывода. На рис. 14.5 показано, как работают компоненты сервера VNC. Для сравнения

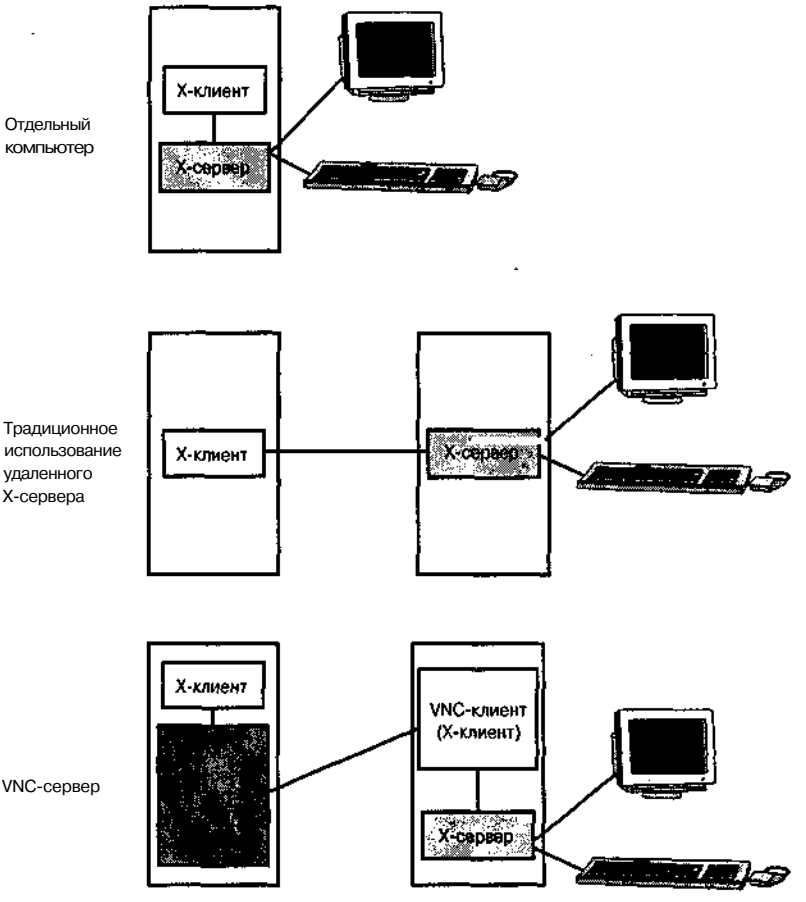


Рис. 14.5. Сервер VNC взаимодействует с локальным X-сервером и удаленным клиентом VNC. Клиент, в свою очередь, взаимодействует с X-сервером и поддерживает обмен данными с клавиатурой, мышью и дисплеем

на этом же рисунке изображено взаимодействие X-клиента и X-сервера, работающих на отдельном компьютере и в сети.

X-сервер, работающий совместно с сервером VNC, поддерживает свое текущее состояние даже в том случае, когда VNC соединение разрывается. Например, если в работе сервера VNC возникнет сбой или если пользователь закроет клиент-программу, не завершив сеанс, сервер VNC продолжит свою работу, и когда пользователь возобновит соединение, приложения, работающие с сервером VNC, останутся открытыми. Такая возможность во многих случаях упрощает работу, например, она может быть очень полезна тогда, когда сеть функционирует ненадежно. Однако не стоит пользоваться ею без необходимости. Если вы надолго прервете сеанс взаимодействия, в работе сервера VNC может возникнуть ошибка; не исключено также, что соединением воспользуется злоумышленник, пытающийся получить доступ к важной информации. (Заметьте, что текущее состояние не поддерживается, когда средства VNC работают совместно с XDMCP.)

VNC обеспечивает уровень защиты выше, чем в Telnet, но ниже, чем XDMCP при X-взаимодействии посредством SSH. VNC кодирует пароль, но остальные данные передаются в незашифрованном виде. Таким образом, при использовании VNC существует опасность перехвата данных, особенно если соединение устанавливается через Internet.

Средства X Window разрабатывались специально для работы в сети. При обмене X-клиента с X-сервером информация пересылается в виде символьных строк; битовые карты формируются на том компьютере, на котором осуществляется вывод на дисплей. При работе X Window по сети часто передаются небольшие фрагменты данных. В отличие от X Window, средства VNC ориентированы на работу с битовыми картами. При работе сервер VNC поддерживает сравнительно небольшое число транзакций, но в рамках каждой транзакции передается значительный объем данных. Это означает, что *в* некоторых сетях и для некоторых приложений VNC будет работать медленнее обычного X-соединения. Например, если вы работаете с текстовым редактором, он передает X-серверу отдельные символы или слова, получив которые X-сервер генерирует битовую карту на локальной машине. Если же взаимодействие с редактором будет осуществляться посредством VNC, по сети будут передаваться битовые карты, объем которых значительно превышает объем самого подробного описания текста. Различия в скорости работы незначительны в быстродействующих сетях, но становятся все более заметными при уменьшении пропускной способности линий. Способ обмена между компьютерами практически не имеет значения при работе с графическими программами, генерирующими растровые изображения. VNC может работать быстрее на линиях с большой задержкой, например при связи через спутник; в этом случае частые обмены информацией, типичные для системы X Window, являются недостатком и замедляют взаимодействие. Если скорость обмена, обеспечиваемая клиентом и сервером VNC, не устраивает вас, вы можете использовать модифицированный вариант VNC, в котором используется кодирование данных, например TightVNC (`http://www.tightvnc.com`) или TridiaVNC (`http://www.developvnc.org`). Кодирование с целью уменьшения объема передаваемых данных занимает ресурсы процессора, поэтому TightVNC или TridiaVNC желательно использовать тогда, когда на обоих концах соединения находятся быстродействующие компьютеры. Сжатие данных может также осуществляться при туннелировании VNC через SSH. При этом, в зависимости от пропускной способности линий и быстродействия процессора, эффективность работы может увеличиться либо уменьшиться.

Одно из преимуществ VNC перед X Window состоит в том, что VNC можно использовать для управления системами Windows и MacOS. Сервер VNC получает контроль над устройствами ввода-вывода и передает информацию клиенту VNC. Функционирование серверов VNC для Windows и MacOS здесь описываться не будет, достаточно лишь сказать, что они работают подобно серверу VNC в Linux, но предоставляют ограниченные возможности, а настройка их осуществляется с помощью диалоговых окон. Обращение к серверу VNC, работающему в среде Windows или MacOS, выполняется так же, как и обращение к серверу VNC в Linux. Поскольку в системе Windows или MacOS сервер VNC перехватывает управление экраном, в каждый момент времени работать с компьютером может только один пользователь.

Инсталляция сервера VNC

Программу, реализующую сервер VNC, можно получить с Web-узла VNC `http://www.uk.research.att.com/vnc/`. Сервер и клиент VNC поставляются со многи-

ми версиями Linux (VNC распространяется в исходных кодах). Иногда и клиент, и сервер входят в состав одного пакета (такой пакет обычно называется vnc), а иногда оформляются в виде разных пакетов (в этом случае пакеты называются vncserver и vnc). Процедура инсталляции Tight VNC и TridiaVNC ничем не отличается от той, которая будет описана ниже.

Если VNC входит в состав вашей версии Linux либо если вы имеете в своем распоряжении архив, содержащий двоичные файлы, инсталляция VNC сводится к выполнению следующих действий (предполагается, что вы инсталлируете версию 3.3.3r2 VNC).

1. Распакуйте архив, вызвав команду `tar xvfz vnc-3.3.3r2_x86_linux_2.0.tgz`. При выполнении этой команды будет создан каталог `vnc_x86_linux_2.0`.

2. Скопируйте файлы vncviewer, vncserver, vncpasswd, vncconnect и Xvnc в один из каталогов, указанных в переменной окружения PATH. При желании вы можете поступить и по-другому: скопировать весь каталог `vnc_x86_linux_2.0` в подходящую для вас позицию файловой системы (например, в каталог /opt) и указать этот каталог в переменной окружения PATH. Если необходимо, вы можете обеспечить доступ к каталогу с помощью символьной ссылки.

3. Создайте в рабочем каталоге пользователя, который должен работать с VNC, подкаталог с именем `.vnc`. Владельцем этого каталога должен быть сам пользователь. В этом каталоге будут содержаться конфигурационные файлы, в том числе файл пароля. Чтобы предотвратить утечку информации, необходимо установить права доступа 700 (`rwx------`).

4. От имени пользователя, работающего с VNC, введите команду vncpasswd. Как нетрудно догадаться, с помощью утилиты vncpasswd задается пароль. В отличие от большинства других серверов регистрации, VNC не полагается на результаты процедуры аутентификации, проведенной средствами Linux. (Если VNC работает совместно с сервером XDMCP, за аутентификацию отвечает Linux, поэтому вы можете не выполнять пп. 3 и 4 данной процедуры.)

Выше были описаны действия, которые обычно выполняются при инсталляции сервера и клиента VNC. VNC также обеспечивает работу в режиме Java-сервера. Этот режим позволяет пользователю обращаться к серверу VNC с любого Web-броузера, поддерживающего Java. Необходимые классы Java находятся в подкаталоге `classes`. Дополнительная информация об инсталляции и использовании этих классов находится в файле README.

Запуск сервера VNC

Для того чтобы запустить сервер VNC, надо зарегистрироваться как обычный пользователь на том компьютере, на котором инсталлирован этот сервер. Обычно пользователи регистрируются с того узла сети, на котором они собираются работать, но при необходимости вы можете подготовить сервер VNC к взаимодействию с удаленной системой, с помощью консольного терминала. Для запуска сервера надо задать от имени обычного пользователя следующую команду:

```
$ vncserver
```

```
New 'X' desktop is vncserv.threeroomco.com:1

Starting applications specified in /home/rodsmith/.vnc/xstartup
Log file is /home/rodsmith/.vnc/vncserv.threeroomco.com:1.log
```

Обратите внимание на данные, отображаемые при выполнении команды; особенно для вас важен номер рабочего стола. В приведенном выше примере это номер 1 — число, которое отображается после имени узла (`vncserv.threeroomco.com:1`). В процессе работы VNC запускает X-сервер (программа Xvnc). Этот X-сервер можно рассматривать как сервер, запускаемый посредством команды `startx`; он формирует среду рабочего стола или диспетчер окон. Если несколько пользователей запускают серверы VNC с одного компьютера, необходимы средства, позволяющие идентифицировать их. В качестве идентификатора используется номер X-сервера. Номер 0 обычно выделяется для X-сервера, связанного с консолью, поэтому первому серверу VNC, вероятнее всего, будет соответствовать номер 1. В последующих сеансах VNC будут использоваться номера 2, 3 и т. д.

ВНИМАНИЕ Если вы зарегистрируетесь на удаленном узле средствами SSH и попытаетесь вызвать сервер VNC, вы, возможно, обнаружите, что выполняется только сервер VNC, а остальные программы (в том числе диспетчеры окон) не работают. В результате вы увидите экран, заполненный фоновым цветом без окон. Так происходит потому, что SSH пытается установить конфигурацию `xauth` в соответствии с настройкой своих средств **туннелирования** X-взаимодействия. Чтобы избавиться от этой проблемы, нужно перед запуском `vncserver` задать команду `export XAUTHORITY=~/.Xauthority`, в результате выполнения которой будут восстановлены установки по умолчанию. Можно также скопировать записи из файла, используемого по умолчанию, во временный файл SSH.

Закончив работу с сервером VNC, надо завершить сеанс взаимодействия, указав для этого опцию `-kill`:

```
$ vncserver -kill :1
```

Число в составе данной команды определяет номер сеанса VNC; этот номер отображается при вызове `vncserver`. Завершать выполнение сервера VNC не обязательно, но работающий сервер напрасно занимает память компьютера. Вызывать данную команду целесообразно с точки зрения безопасности системы, поскольку невозможно воспользоваться недостатками в защите сервера, если он не выполняется. Перед тем как отключать сервер, убедитесь, что вы окончили работу со всеми программами и закрыли файлы, так как сервер VNC, заканчивая работу, не выводит предупреждающих сообщений.

Использование клиента YNC для взаимодействия с сервером

Программа, реализующая функции клиента VNC в системе Linux, называется `vncviewer`. Для вызова клиента надо ввести имя программы и, возможно, имя сервера VNC и номер дисплея.

```
$ vncviewer vncserv.threeroomco.com:1
VNC server supports protocol version 3.3 (viewer 3.3)
Password:
```

При вводе пароля символы не отображаются на экране. При выполнении клиент VNC выводит дополнительную информацию, например, число битов, используемых для представления цвета, и другие технологические характеристики. Если клиент-программа выполняется корректно, вы увидите на экране окно, отображающее рабочий стол Linux. Возможно, что конфигурация сервера VNC вас не устроит и вы захотите изменить ее.

Если вы не укажете номер дисплея, клиент VNC попытается подключиться к дисплею с номером 0, который в системе Linux не работает, так как используется локальным X-сервером. Без указания номера дисплея можно обращаться к серверу VNC, работающему в системе Windows или MacOS. Если вы не зададите имя узла, клиент VNC отобразит диалоговое окно, в котором предложит вам ввести имя узла и пароль.

Клиенты, предназначенные для работы в Windows и MacOS, действуют аналогично клиенту VNC Linux. Для запуска программы надо дважды щелкнуть на соответствующей пиктограмме, в результате чего клиент отобразит диалоговое окно для ввода имени сервера VNC и номера дисплея (например, `vncserv.threeroomco.com:1`). Если имя узла задано верно, у пользователя запрашивается пароль, после ввода которого будет выведено окно с рабочим столом сервера VNC.

Настройка сервера VNC

VNC представляет собой удобный инструмент удаленного доступа, но при его использовании могут возникать проблемы. В частности, многие пользователи сообщают об ошибках, возникающих при совместной работе редактора NEdit (`http://www.nedit.org`) и VNC. В моей системе NEdit не реагировал на нажатие клавиш, т. е. оказался совершенно непригоден к использованию. К счастью, серьезные ошибки, подобные этой, возникают достаточно редко. В большинстве случаев проблему удается решить с помощью настройки компонентов VNC. Характеристики VNC можно задавать, редактируя сценарий, используемый для запуска сервера, либо изменяя содержимое конфигурационных файлов.

Установка основных характеристик сервера

Программа, реализующая функции сервера VNC, называется Xvnc. Эта программа содержит X-сервер (взаимодействующий с локальными X-программами) и сервер VNC (который взаимодействует с клиентом VNC). Вы, вероятно, заметили, что при обсуждении работы сервера VNC программа Xvnc не упоминалась. Дело в том, что эта программа вызывается из сценария `vncserver`, используемого для запуска сервера VNC. Сценарий `vncserver` написан на языке Perl; изменяя его код, вы можете задавать характеристики сервера VNC, принимаемые по умолчанию. Некоторые из установок, которые можно осуществить, редактируя код сценария, описаны ниже.

- **Автоматическая установка параметров, используемых по умолчанию.** В последних версиях `vncserver` для определения размера дисплея, числа битов, используемых для представления цвета, и других параметров применялся вызов `&GetXDisplayDefaults()`. Однако при этом может быть получено значение размера, не подходящее для клиента. Если вы хотите изменить размер экрана, вам надо закомментировать данную строку, поместив в начале ее символ #, и указать размер экрана явным образом. В сценарии, поставляемом в составе пакета, размер экрана устанавливается до вызова `&GetXDisplayDefaults()`.

- **Размер экрана.** При запуске программа Xvnc создает виртуальный экран определенного размера. Если вы не используете опции по умолчанию, установите размер экрана с помощью переменной $geometry. Например, чтобы задать размер 900 x 675, надо включить в состав сценария следующую строку:

```
$geometry = "900x675";
```

СОВЕТ ✔ Поскольку клиент VNC отображает рабочий стол сервера VNC в окне, имеет смысл указать размер дисплея несколько меньше, чем реальный его размер. Оставшееся место понадобится для вывода обрамления. Если размер дисплея выбран слишком большим, для просмотра рабочего стола придется использовать элементы прокрутки.

- **Глубина цвета.** Включив в состав сценария переменную **$depth,** вы можете контролировать число битов, используемых для представления цвета. Во многих случаях для кодирования цвета бывает достаточно 16 битов, однако программы, воспроизводящие большое количество разнообразных цветов, могут искажать данные, отображаемые другими программами. Это правило не распространяется на VNC; 16-битовое представление может привести к некорректному отображению цвета. В будущем данная проблема, скорее всего, будет решена.

- **Шрифт, или путь к шрифту.** Сценарий, поставляемый в составе пакета, по умолчанию настроен для использования сервера шрифтов. Изменить эту настройку можно с помощью раздела Add font path and color database stuff here. Для добавления шрифта используется параметр **-fp** в строке $cmd, которая используется при вызове Xvnc. При необходимости вы можете сконфигурировать VNC для работы с сервером шрифтов. Использованием сервера шрифтов описано в главе 15.

- **Диспетчер окон, используемый по умолчанию.** Сценарий vncserver, поставляемый в составе дистрибутивного пакета, содержит переменную **$defaultXStartup,** определяющую содержимое пользовательского сценария запуска. При первом запуске сценарий vncserver помещает соответствующий файл в пользовательский каталог. По умолчанию задан диспетчер окон **twm,** который в настоящее время используется достаточно редко. Вы можете отказаться от значения, заданного по умолчанию, и заменить вызов twm на вызов другого диспетчера окон или среды рабочего стола, например startkde, sawmill или **icewm.** Изменения, внесенные в сценарий vncserver, повлияют на работу только тех пользователей, которые еще не запускали данный сценарий. Ниже будет рассмотрены средства установки конфигурации для существующих пользователей.

Даже если вы плохо знакомы с языком Perl, просмотрев данный сценарий, вы найдете сведения о многих характеристиках, которые, возможно, захотите изменить. В основном данный сценарий устанавливает опции, которые должны быть указаны при запуске Xvnc; они помещаются в строку **$cmd.** Разобравшись в том, как формируются опции, вы сможете легко модифицировать их. По команде Xvnc **-help &> Xvnc-help.txt** создается текстовый файл с именем **Xvnc-help.txt**, содержащий информацию о доступных опциях Xvnc.

Перед тем как вносить изменения в сценарий vncserver, необходимо создать его резервную копию. Она понадобится в случае, если вы допустите ошибку и вам придется вернуться к исходному варианту сценария.

Сценарии vncserver, входящие в состав некоторых пакетов, существенно отличаются от исходного варианта. Особенно это относится к сценарию, поставляемому в составе системы Debian. Тем не менее советы, приведенные выше, применимы ко всем разновидностям vncserver. Необходимо лишь перед внесением изменений ознакомиться с конкретными особенностями сценария. Например, сценарий для системы Debian создает для определения шрифта переменную `$fontpath`.

Изменение параметров для отдельных пользователей

Глобальные характеристики сервера VNC задаются с помощью сценария vncserver. Если пользователь захочет изменить некоторые установки, он может предпринять следующие действия.

- **Самостоятельно создать сценарий запуска сервера VNC.** Пользователь может скопировать сценарий в свой каталог, модифицировать его и использовать в дальнейшем для запуска сервера.
- **Организовать передачу опций сценарию.** Сценарий vncserver обрабатывает несколько опций, которые могут быть использованы для переопределения значений, заданных по умолчанию. Например, опция -geometry *ширина_и_высота* устанавливает размер рабочего стола. Эти опции в основном совпадают с опциями программы Xvnc.
- **Редактировать отдельные конфигурационные файлы.** Стандартный сценарий запуска сервера VNC перед окончанием своего выполнения вызывает сценарий ~/.vnc/xstartup. В нем содержатся команды запуска диспетчера окон и xterm. Пользователь может редактировать этот файл так же, как и обычный сценарий запуска X Window. В некоторых дистрибутивных пакетах имя и расположение этого сценария отличается от указанных здесь. Например, в системе Debian вызывается сценарий /etc/X11/Xsession, который, в свою очередь, запускает пользовательский сценарий ~/.xsession.

В большинстве случаев для организации передачи опций и редактирования конфигурационных файлов приходится затрачивать гораздо меньше усилий, чем для создания сценария запуска. Однако бывают ситуации, когда один из способов настройки оказывается намного удобнее остальных. Например, размер экрана проще всего задавать с помощью опции -geometry в сценарии vncserver, а диспетчер окон лучше всего настраивать, используя его сценарий запуска. Общее правило таково: содержимое сценария vncserver позволяет задать поведение X-сервера в составе VNC, а опции сценария запуска дают возможность настроить диспетчер окон и среду рабочего стола.

Совместная работа серверов XDMCP и VNC

Один из главных недостатков VNC заключается в следующем: для того, чтобы начать работу с сервером VNC, надо зарегистрироваться, используя один из стандартных протоколов, загрузить сервер и запомнить номер дисплея. Эта рутинная процедура мешает выполнению реальных задач. Решением данной проблемы может быть использование VNC X-сервера и сервера XDMCP на одном компьютере.

Подобно большинству X-серверов, VNC X-сервер позволяет серверу XDMCP управлять отображением данных. Для того, чтобы это стало возможным, вам надо указать при запуске VNC X-сервера опцию -query *имя_узла*. Если вы используете суперсервер xinetd, соответствующая запись в конфигурационном файле будет выглядеть следующим образом:

```
service vnc
{
    disable = no
    socket_type = stream
    protocol = tcp
    wait = no
    user = nobody
    server = /usr/local/bin/Xvnc
    server_args = -inetd -query vncserv -once
}
```

В данном случае важно правильно задать параметры сервера. В частности, опция -inetd сообщает Xvnc о том, что он запущен посредством суперсервера, -query vncserv означает, что необходимо обратиться к vncserv. Опция -once свидетельствует о том, что сервер должен быть вызван однократно, а затем прекратить свою работу; в результате, если пользователь завершит сеанс взаимодействия, соединение VNC будет разорвано. Вы можете также использовать и другие опции Xvnc, например -geometry или -fp. Кроме того, в файле /etc/services должно присутствовать описание порта.

```
vnc 5900/tcp
```

Для обычных соединений VNC использует номера портов 5900-5999, а порты 5800-5899 применяются для обработки обращений посредством Web-броузера (поддержки режима Java-сервера). Порт 5900 соответствует дисплею 0, порт 5901 — дисплею 1 и т. д. Таким образом, приведенное выше описание задает отображение приглашения к регистрации XDMCP и взаимодействие VNC через порт 0. Очевидно, что сервер XDMCP должен выполняться на компьютере, определенном посредством опции -query. Вы можете настроить систему так, чтобы она по-разному реагировала на обращения клиента через различные порты. Например, дисплею 0 может соответствовать размер рабочего стола 800 x 600, дисплею 1 — размер 1024 x 768 и т. д. Для идентификации таких серверов необходимо поместить в файл /etc/services несколько записей: по одной на каждый порт. Настроенный таким образом сервер VNC не требует ввода пароля — все детали взаимодействия обеспечивает сервер XDMCP. (Заметьте, что в отличие от традиционного VNC-взаимодействия, имя пользователя и пароль передаются в незакодированном виде.) Еще одна особенность сконфигурированного подобным образом сервера VNC состоит в том, что он может принимать обращения нескольких пользователей через один порт. Таким образом, совместное использование серверов VNC и XDMCP можно условно сравнить с применением сервера XDMCP и удаленного X-сервера. Однако эти системы имеют ряд отличий. Наиболее важные из них описаны ниже.

- При использовании VNC между двумя компьютерами устанавливается одно соединение. Это удобно в тех случаях, когда взаимодействующие компьютеры разделены брандмауэром; при этом уменьшается количество серверов, доступных извне.

- При работе с VNC на пользовательском компьютере вместо X-сервера выполняется клиент VNC. Сервер VNC распространяется в исходных кодах, поэтому он свободно доступен, в то время как большинство X-серверов для Windows и MacOS предоставляется на коммерческой основе.
- Протокол VNC имеет свои особенности. Если на пользовательском компьютере вы замените X-сервер клиентом VNC, качество системы может как повыситься, так и снизиться, в зависимости от потребностей пользователя и применяемого X-сервера.
- В большинстве случаев протокол VNC обеспечивает меньшее быстродействие по сравнению с X Window, однако в некоторых случаях применение VNC вместо X Window может повысить производительность системы.

Преимущества и недостатки различных технологий удаленной регистрации

В табл. 14.1 приведены наиболее важные характеристики различных технологий удаленной регистрации, рассмотренных в данной главе. Заметьте, что конкретные оценки могут различаться в зависимости от реализации протокола и конфигурации программных средств. Например, уровень безопасности при VNC-регистрации в текстовом режиме зависит от применяемого инструмента. Если вы используете для установления начального соединения SSH, то защита оценивается как отличная, если же кодирование передаваемых данных не осуществляется, информация может быть перехвачена, поэтому защита считается неудовлетворительной.

Итак, какие же средства удаленной регистрации следует применить, если пользователю должен быть предоставлен графический интерфейс? На этот вопрос нельзя дать однозначный ответ. X Window очень хорошо подходит для обмена данными между компьютерами под управлением Linux или UNIX в локальной сети. В этих операционных системах есть практически все необходимые программные компоненты, X Window обеспечивает высокое быстродействие, и работа с ней не вызывает затруднений. С помощью X Window удобно также запускать программы в системе Linux с компьютеров под управлением Windows и MacOS, но вам придется установить X-серверы на соответствующих машинах, а это может оказаться слишком дорого. Для установки VNC требуются меньшие средства, кроме того, в данной системе используется традиционное расположение клиента и сервера (в отличие от X Window, где X-сервер находится на компьютере, за которым работает пользователь, а X-клиент — на удаленной машине). VNC удобно применять в тех случаях, когда клиент и сервер разделены брандмауэром. Брандмауэр с большей вероятностью будет блокировать обмен данными посредством X Window, когда сервер находится на пользовательском компьютере.

Таблица 14.1. Характеристики средств удаленной регистрации, поддерживающих графический интерфейс

Характеристика	Регистрация в текстовом режиме	**SSH-регистрация**	XDMCP-регистрация	Регистрация в текстовом режиме с использованием VNC	Совместное использование VNC и XDMCP
Уровень защиты при регистрации	Очень низкий	Очень высокий	Очень низкий	От очень низкого до очень высокого	Очень низкий
Уровень защиты при передаче данных	Очень низкий	Очень высокий	Очень низкий	Очень низкий	Очень низкий
Вероятность возникновения проблем при наличии брундмауэра	Высокая	Низкая	Высокая	Низкая	Низкая
Способность к сохранению состояния сеанса (разрыв и восстановление соединения)	Низкая	Низкая	Низкая	Высокая	Низкая
Быстродействие	Высокое	Среднее	Высокое	От низкого до среднего	От низкого до среднего
Вероятность возникновения проблем при работе приложения	От низкой до средней	От низкой до средней	От низкой до средней	Средняя	Средняя

Резюме

Серверы удаленной регистрации — чрезвычайно полезные инструменты. В особенности они нужны в тех случаях, когда пользователи имеют учетные записи на нескольких компьютерах, или тогда, когда нужно обеспечить выполнение основных программ на центральной машине и организовать доступ к ним с менее мощных пользовательских компьютеров. Чаще всего на компьютерах под управлением Linux используется система X Window, которая позволяет пользователям регистрироваться на удаленной машине и предоставляет графический интерфейс. Организовать удаленный доступ посредством X Window можно различными способами: инициализировать X-взаимодействие посредством одного из протоколов, обеспечивающих работу в текстовом режиме, либо использовать сервер XDMCP для аутентификации удаленных пользователей. Для организации удаленного доступа можно также применять средства VNC. В этом случае в процесс обмена данными вовлекаются дополнительные компоненты, но с точки зрения пользователя работа с удаленным узлом упрощается.

Глава 15

Серверы шрифтов

В цивилизованном мире не найдется человека, который никогда не видел букв. Одна и та же буква может выглядеть по-разному. Например, буква "Р" в заголовке главы отличается от той же буквы в тексте абзаца. Буква "Р", отображаемая курсивом, отличается от их обеих. Рассмотренные здесь варианты буквы "Р" принадлежат различным шрифтам. Шрифт определяет внешний вид букв (как в верхнем, так и нижнем регистре), цифр и знаков пунктуации. Существуют шрифты, не содержащие букв; они состоят исключительно из специальных символов. Шрифты — это необходимые компоненты программного обеспечения современных компьютеров. Приступая к работе с текстовым процессором или Web-броузером, пользователь справедливо предполагает, что в его распоряжение предоставлен набор шрифтов и что он в любой момент может выбрать наиболее подходящий шрифт. Иногда, например при решении задач электронной публикации, наличие многих шрифтов является одним из основных требований к системе. В других случаях, например при просмотре электронной почты, возможность выбора шрифта лишь создает пользователю более комфортные условия для работы.

Один из способов управления шрифтами в системе Linux состоит в использовании сервера шрифтов. Этот сервер имеет доступ к набору шрифтов и может доставлять их клиентам, подключенным к сети. На первый взгляд сервер шрифтов может показаться абсолютно бесполезным компонентом, так как на большинстве компьютеров поддерживаются свои наборы шрифтов. Однако в некоторых ситуациях сервер шрифтов существенно упрощает процедуру администрирования сети. Для того чтобы грамотно использовать сервер шрифтов, необходимо хотя бы в общих чертах знать форматы файлов шрифтов и представлять себе преимущества и недостатки этих форматов. Большинство серверов шрифтов, работающих в системе Linux, выполняют ограниченный круг задач, в основном связанных с поддержкой работы системы X Window. Но возможности сервера шрифтов могут быть расширены. Такие серверы чаще всего предназначаются для работы с текстовыми процессорами и другими подобными приложениями.

Использование серверов шрифтов

Одним из параметров, задаваемых в конфигурационном файле X Window, является *шрифт,* или *путь к шрифту.* В пакете XFree86 для этого используется запись FontPath.

Она включается в файл `XF86Config`, который обычно хранится в каталоге /etc или `/etc/X11`. Данный параметр указывает X-серверу на расположение информации о шрифтах. В середине 1990-х годов в системе Linux в качестве пути к шрифту задавались каталоги локальной файловой системы. В этих каталогах содержались шрифты и вспомогательные файлы. Такая конфигурация до сих пор поддерживается, а в некоторых дистрибутивных пакетах даже устанавливается по умолчанию.

При работе X-сервера со шрифтами, хранящимися на жестком диске, возникает ряд проблем. Прежде всего, X-сервер должен поддерживать форматы файлов шрифтов, а это требование не всегда выполняется. Например, версии, предшествующие XFree86 4.0, не поддерживали популярный формат TrueType. Реализация средств обработки нового формата шрифтов в XFree86 — чрезвычайно трудная задача, которая, тем не менее, достаточно просто решается сервером шрифтов. Поэтому TrueType стал поддерживаться серверами шрифтов гораздо раньше, чем XFree86. Несмотря на то что задача обработки TrueType в XFree86 уже решена, время от времени возникают проблемы, связанные с появлением новых форматов шрифтов. Так, например, в настоящее время становится все более популярным формат Multiple Master, поддержку которого необходимо обеспечивать.

Еще одна проблема, препятствующая непосредственному использованию шрифтов, хранящихся на диске, имеет отношение к администрированию сети. Если вам необходимо, чтобы некоторый набор шрифтов стал доступен всем компьютерам в сети, установка его на каждую машину займет много времени. Ситуация станет еще хуже, если вам придется постоянно добавлять или удалять отдельные шрифты. Сервер шрифтов позволяет осуществлять централизованное управление шрифтами; вам достаточно сконфигурировать каждую клиентскую программу для работы с конкретным сервером, а все последующие изменения будут затрагивать лишь сервер. В результате администрирование компьютеров намного упростится.

И, наконец, не стоит забывать, что серверы шрифтов обеспечивают возможности, которые не может предоставить X-сервер, который работает со шрифтами, находящимися на диске. Эти возможности бывают необходимы для текстовых процессоров, издательских систем и других подобных программ. X-сервер в основном ориентирован на отображение информации на экране монитора и не удовлетворяет всем требованиям, которые предъявляют к нему специализированные программы, предназначенные для работы с текстом. Сервер шрифтов помогает разрешить эту проблему и обеспечивает работу программ обработки текста, в частности программ WYSIWYG (what-you-see-is-what-you-get).

Сервер шрифтов может работать на локальном компьютере и предоставлять шрифты выполняющимся на нем программам либо обслуживать всю сеть. Эти две конфигурации в основном совпадают и различаются между собой лишь деталями.

ВНИМАНИЕ В США шрифты не подпадают под закон об авторском праве; этот закон защищает лишь реализации шрифтов, выполненные в виде файлов, пригодных для обработки на компьютере. В других странах дело обстоит иначе. Перед тем как предоставлять доступ к шрифту посредством сервера шрифтов, необходимо внимательно ознакомиться с действующим законодательством. Возможно, чтобы получить право на использование шрифта, вам придется заплатить определенную сумму его владельцу; часто эта сумма зависит от числа компьютеров, обслуживаемых сервером.

Форматы файлов шрифтов

Существуют два типа шрифтов: *растровые* и *контурные* (контурные шрифты часто называют *масштабируемыми)*. Эти типы шрифтов имеют разные свойства и обрабатываются различными способами. Большинство серверов шрифтов, предназначенных для выполнения в системе Linux, поддерживают оба типа шрифтов. Администратор сети должен уметь различать эти типы шрифтов и оценивать преимущества и недостатки каждого из них.

Форматы растровых шрифтов

Область отображения большинства устройств, предназначенных для вывода шрифтов (в том числе дисплеев и принтеров), состоит из отдельных прямоугольных элементов, называемых *пикселями*. Каждый пиксель может быть окрашен в тот или иной цвет. На монохромном дисплее или на черно-белом лазерном принтере пиксель принимает одно из двух возможных значений, например, отображает черный или белый цвет. Цветные дисплеи и принтеры могут воспроизводить различные цвета, но символы стандартных шрифтов X Window формируются с помощью двух цветов. (Вы можете выбирать любые цвета, например, отображать текст с помощью черного и белого или красного и желтого цветов, но в формировании каждого символа будут участвовать только два цвета.) Каждый пиксель в составе растрового шрифта кодируется с помощью одного бита. Растровое изображение символа представляет собой набор битов — *битовую карту*. При отображении текста битовая карта символа копируется на устройство отображения.

В качестве примера рассмотрим рис. 15.1, на котором показано представление одного символа растрового шрифта. Даже если не принимать во внимание, какие именно пиксели закрашены черным, а какие белым цветом, данный рисунок иллюстрирует ряд характеристик растрового шрифта. *Знакоместо* (часть растровой сетки, предназначенная для отображения символа) имеет фиксированные размеры. В пропорциональных шрифтах, которые чаще всего используются в книгопечатании, различные символы могут иметь разную ширину. Соответственно различается ширина знакомест для разных символов одного и того же шрифта. Высота знакоместа фиксирована; фиксирована также высота большинства символов. (Существуют так называемые символы с подстрочными элементами. Подстрочный элемент располагается под нижней границей символа. Примерами подобных символов являются g, j, p, q и y. Заметьте, что на рис. 15.1 символ размещается в пределах знакоместа так, что остается место для подстрочного элемента.) Поскольку высота знакоместа фиксирована, размер шрифта на одном устройстве отображения остается постоянным. При переходе на другое устройство с другой разрешающей способностью размер символов изменится. Чтобы обеспечить отображение символов одинакового размера на различных устройствах или отображение символов разного размера на одном устройстве, необходимо иметь в наличии набор битовых карт.

Разрешающая способность дисплея обычно выражается в точках на дюйм (dpi — dots per inch), т. е. разрешение — это количество пикселей, помещающихся на отрезке в один дюйм. В большинстве устройств разрешающая способность по горизонтали и по вертикали совпадает, однако в некоторых случаях она может различаться. Мониторы компьютеров обычно имеют разрешение от 72 до 120 dpi, а разрешающая способность принтеров, как правило, лежит в пределах 144-1200 dpi. (Разрешающая способность, равная 144, характерна для матричных принтеров; кроме того, такое разрешение часто

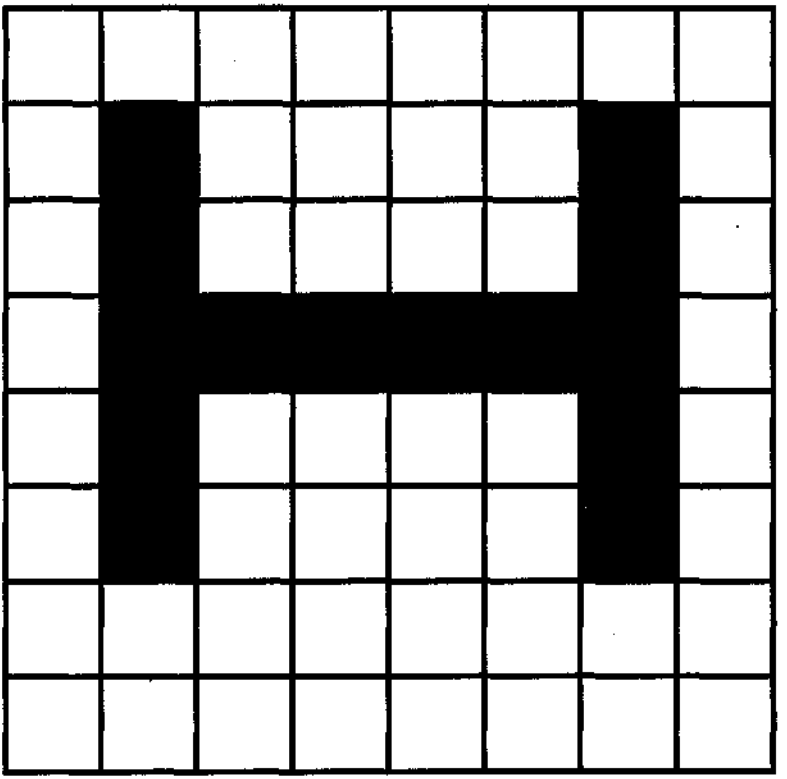


Рис. 15.1. Битовая карта определяет, какие пиксели в составе знакоместа должны отображаться черным цветом, а какие — белым

устанавливают, чтобы предельно ускорить процесс печати за счет снижения ее качества.) Разрешающая способность высокоуровневых принтеров и специализированных полиграфических устройств, как правило, намного превышает 1200 dpi. Разнообразие устройств печати и необходимость создания отдельного файла для каждого размера шрифта и для каждого значения разрешающей способности приводит к тому, что общее число файлов становится недопустимо большим.

Многие принтеры снабжены встроенными растровыми шрифтами. В 1980-х годах такие шрифты широко применялись при выводе информации на печать. В настоящее время лишь отдельные программы работают со встроенными шрифтами. В большинстве случаев используются файлы шрифтов, хранящиеся на компьютере.

Размеры шрифтов измеряются в *пунктах* (эта единица измерения широко применяется в полиграфии). Для отображения текста абзаца обычно используется шрифт размером 9-14 пунктов. Выбор размера зависит от шрифта и назначения текста. Растровые шрифты чаще всего создаются в том случае, когда необходимо отображать символы фиксированных размеров на устройстве с определенной разрешающей способностью, например, если нужен текст размером 12 пунктов на устройстве с разрешением 144 dpi. Один и тот

же шрифт позволяет отображать символы разного размера на устройствах с различной разрешающей способностью, однако шрифт, созданный с нуля и ориентированный на устройство с конкретным разрешением будет несколько отличаться от шрифта, перенесенного с другого устройства. При необходимости для отображения символов требуемого размера можно изменить битовые карты символов (например, уменьшить размер символов с 12 до 10 пунктов), однако в результате подобных действий отображается текст плохого качества.

Основным преимуществом растровых шрифтов является их простота, а следовательно, и возможность быстрого вывода на устройство отображения. Для вывода символа достаточно скопировать несколько битов из памяти компьютера на дисплей. Это было чрезвычайно важно в 1980-х годах, когда объем ресурсов, доступных пользователям, был крайне мал, но в середине 1990-х к внешнему виду отображаемых данных стали предъявляться все более строгие требования и качество растровых шрифтов перестало устраивать пользователей. В настоящее время быстродействие компьютеров достаточно велико и растровые шрифты применяются все реже.

Для представления шрифтов используются различные форматы. В ранних версиях X Window применялся формат SNF (Server Normal Format — обычный формат сервера), но сейчас он встречается редко. Как правило, в X Window используются шрифты в формате PCF (Portable Compiled Font — переносимый компилируемый шрифт). Формат BDF (Bitmap Distribution Format — формат распространения битовых карт) также часто применяется в X Window, но чтобы X-программы могли работать с ним, сервер шрифтов преобразует шрифты BDF в формат PCF. В других операционных системах используются другие форматы. Чтобы работать с ними в системе Linux, надо воспользоваться одним из преобразователей шрифтов.

XFree86 использует PCF-шрифты, сжатые посредством программы `gzip`. В большинстве дистрибутивных пакетов для экономии дискового пространства шрифты поставляются в сжатом виде. При этом имена PCF-файлов оканчиваются символами `.pcf.gz`. Чтобы использовать эти шрифты при работе X-программ, не обязательно распаковывать их.

Шрифты, применяемые в системе Linux, не ограничиваются используемыми в X Window. Некоторые программы работают с собственными наборами шрифтов. Одной из таких программ является система TeX, в которой применяется формат Packed Font (файлы шрифтов имеют расширение .pk). Поскольку система TeX в основном разрабатывалась для подготовки материалов к печати, а представлению текста на экране монитора уделялось не слишком большое внимание, число пикселей, составляющих знакоместа в шрифтах Packed Font, существенно превышает соответствующий показатель других форматов.

Форматы контурных шрифтов

Одна из основных проблем, возникающих при работе с растровыми шрифтами, состоит в том, что эти шрифты плохо масштабируются. Если вам необходимо отображать на одном устройстве символы разных размеров либо выводить текст одного и того же размера на устройства с различной разрешающей способностью, вам потребуется несколько файлов шрифтов. Учитывая разнообразие имеющихся в настоящее время устройств отображения и требования к масштабированию символов, предъявляемые современными программами (например, текстовыми процессорами), становится очевидно, что, для того,

Таблица 15.1. Контурное описание символа, представленного на рис. 15.1

Операция	Координата x	Координата y
Установка в начальную позицию	10000	10000
Прямая	10000	60000
Прямая	20000	60000
Прямая	20000	40000
и т. д.	...	...

чтобы отобразить высококачественный текст на разнообразном оборудовании, потребуется чрезвычайно большой набор файлов шрифтов. Решить эту проблему можно, используя контурные, или масштабируемые шрифты. Вместо битовых карт в контурных шрифтах символы представляются в виде описаний кривых, с помощью которых формируются их контуры. Вернемся к рис. 15.1. Если знакоместо 8 × 8 пикселей увеличить до гораздо больших размеров, например 80000 x 80000, контуры символа могут быть описаны набором прямых так, как это показано в табл. 15.1.

Замкнутый контур заполняется цветом. Основное преимущество контурных шрифтов состоит в том, что символ легко масштабировать для отображения на устройстве с любой разрешающей способностью. Для масштабирования символа достаточно перевести описание из исходной системы координат в систему координат, соответствующую конкретному устройству отображения. Чтобы описание было максимально точным, разрешающая способность в исходной системе координат принимается очень высокой. Описание большинства символов не исчерпывается прямыми линиями. Как правило, контур символа строится из набора кривых. Часто в описание символа включают закодированные специальным образом *рекомендации разработчика* (hint). Эти рекомендации позволяют повысить качество отображения символов на устройствах с низким разрешением, например на мониторах компьютеров.

Формально *шрифтом* называется представление символов определенного фиксированного размера, а для определения одинаковых шрифтов, отличающихся только размерами, используется термин *начертание,* или *семейство шрифтов.* Таким образом, растровый шрифт действительно является шрифтом, а к контурному шрифту лучше подходит термин *начертание.* Однако в литературе, посвященной компьютерам, этими различиями обычно пренебрегают. В данной книге я буду называть шрифтом как шрифт конкретного размера, так и все семейство шрифтов.

При отображении контурных шрифтов необходимо анализировать контур символа, чтобы определить, каким цветом должен быть закрашен тот или иной пиксель. В результате для вывода символов, представленных в контурном формате, требуется намного больше времени, чем для отображения тех же символов, представленных в растровом виде. Ускорить процесс отображения текста можно различными способами. Один из этих способов заключается в том, что шрифт определенного размера заранее *растрируется* (т. е. переводится в растровую форму) и хранится в виде битовой карты. Именно этим занимается сервер шрифтов. Клиент, обращающийся к серверу шрифтов, всегда получает шрифт в растровом виде, независимо от того, в каком формате был представлен исходный шрифт.

Существуют различные форматы представления контурных шрифтов. В качестве примеров можно привести Bitstream Speedo, Adobe Type 1, Type 3, Type 5, Type 42 и Apple TrueType. (Шрифты Type 42 на самом деле представляют собой шрифты TrueType, преобразованные для вывода на PostScript-принтеры.) Различия между разными форматами контурных шрифтов гораздо более существенны, чем в случае растровых шрифтов, так как в них используются различные типы описаний прямых и кривых линий. В большинстве случаев шрифты можно преобразовать из одного формата в другой, но при этом начертания символов будут незначительно отличаться друг от друга. В частности, в процессе преобразования могут быть потеряны рекомендации разработчика, что ухудшит внешний вид символов при выводе на устройства с низким разрешением. Поэтому рекомендуется использовать исходный формат шрифта, а если это невозможно, то следует приобрести у разработчика тот же шрифт, представленный в нужном вам формате.

В Linux (а точнее, в XFree86) реализована поддержка контурных шрифтов Speedo и Adobe Type 1. Шрифты Speedo используются достаточно редко, а шрифты Type 1 нашли широкое применение; они распространяются на компакт-дисках и доступны через Internet. Кроме того, некоторые шрифты Type 1 входят в поставку Linux. В Windows и MacOS шрифты TrueType намного более популярны, чем Type 1. В частности, TrueType является стандартным форматом шрифтов для системы Windows. Считается, что шрифты TrueType позволяют обеспечить более высокое качество отображения на устройствах с низким разрешением, чем Type 1, однако такое утверждение справедливо только в тех случаях, когда в состав шрифта включены подробные рекомендации разработчика. При отсутствии рекомендаций разработчика символы TrueType отображаются ничуть не лучше, чем символы Type 1.

Корпорация Microsoft разработала шрифты TrueType, снабженные подробными рекомендациями. Они предназначены для использования в Web-броузерах и доступны по адресу `http://www.microsoft.com/typography/fontpack/`. Если вы скопируете файлы, ориентированные на применение в Windows 3.1, то сможете непосредственно использовать их в системе Linux. Эти шрифты поставляются в виде самораспаковывающихся (в системе Windows) zip-файлов; в Linux вы можете извлечь их содержимое с помощью утилиты `unzip`. Порядок инсталляции шрифтов будет описан далее в этой главе. Разработчики многих Web-узлов предполагают, что шрифты, о которых идет речь, уже установлены на клиентской машине, поэтому, инсталлировав их, вы сможете наиболее корректно отобразить соответствующие Web-страницы.

В версиях XFree86, предшествующих 4.0, шрифты TrueType не поддерживались, поэтому единственным способом работы с ними было применение сервера шрифтов. В настоящее время есть возможность непосредственного использования шрифтов TrueType в X Window, однако в большинстве дистрибутивных пакетов для работы с TrueType традиционно применяется сервер шрифтов.

X Window — не единственная система, в которой используются контурные шрифты. С помощью специальных инструментальных средств можно, например, обеспечить работу TeX с этим типом шрифтов. Программа Ghostscript (PostScript-интерпретатор для принтеров, не поддерживающих PostScript) также использует растровые шрифты: чаще всего Ghostscript работает со шрифтами Type 1, но в некоторых случаях может применяться и формат TrueType. Растровые шрифты необходимы для обеспечения работы текстовых

процессоров. Упомянутые здесь программы, за исключением текстовых процессоров, не используют сервер шрифтов.

Обеспечение работы традиционного сервера шрифтов

В данной главе термин *традиционный сервер шрифтов* будет использоваться для обозначения программы xfs, поставляемой в комплекте с XFree86, и других подобных ей программ. Сервер шрифтов поставляет X-программам шрифты, представленные в растровом виде, используя для этого стандартный протокол. Исходные файлы шрифтов могут быть как растровыми, так и контурными. Серверы шрифтов в основном предназначены для поддержки вывода символов на экран монитора и не обеспечивают согласование экранных шрифтов с символами, выводимыми на принтер. В состав XFree86 входит xfs, поэтому, даже если сервер шрифтов не инсталлирован, на вашем компьютере скорее всего имеется в наличии все необходимое программное обеспечение для организации его работы. Настройка сервера шрифтов сводится к редактированию нескольких конфигурационных файлов. Если на компьютере уже установлен локальный сервер шрифтов, вы можете перенастроить его так, чтобы к нему могли обращаться X-серверы, работающие на других машинах.

Программы, реализующие сервер шрифтов в Linux

Чаще всего в качестве сервера шрифтов в Linux используется программа xfs, которая поставляется в составе XFree86. По сути эта программа представляет собой набор кодов X Window, используемых для обработки шрифтов и дополненный средствами поддержки сетевого взаимодействия. Как правило, данный сервер помещается в каталог `/usr/X11R6/bin`; пакет, используемый для инсталляции, обычно называется XFree86-xfs или xfs.

При работе с версиями XFree86, предшествующими версии 4.0, вам понадобится модифицированный вариант сервера шрифтов, в котором реализована поддержка TrueType. Два сервера, обеспечивающих такую поддержку, описаны ниже.

- **xf**stt. Данный сервер ориентирован исключительно на работу с TrueType. Type 1, BDF и другие форматы шрифтов не поддерживаются. Этот продукт удобен для обеспечения поддержки TrueType в версиях XFree86, выпущенных раньше, чем XFree86 4.0. Если же вас интересует только работа с форматом TrueType, xfstt можно использовать в качестве сетевого сервера шрифтов. Инсталляционный пакет `xfstt` находится по адресу `ftp://ftp.metalab.unc.edu/pub/Linux/X11/fonts/xfstt-1.1.tar.gz` (в последующих версиях данного продукта файл `xfstt-1.1.tar.gz` может быть переименован). Принимая решение об использовании `xfstt`, следует помнить, что этот сервер предоставляет клиентам шрифты в формате, который зависит от порядка следования байтов, принятого в компьютере. Если в сети присутствуют компьютеры с различными сочетаниями байтов (например, x86 и PowerPC), `xfstt` не может выполнять функции сетевого сервера шрифтов.

- **xfsft**. Данный сервер представляет собой модифицированный вариант стандартного пакета xfs, входящего в состав XFree86 3.3.x. Сервер **xfsft** включает поддержку TrueType средствами FreeType (`http://freetype.sourceforge.net/index2.html`). Результатом данной модификации стал сервер, поддерживающий TrueType, Type 1, BDF и другие форматы шрифтов. Все возможности **xfsft** обеспечивает также стандартная программа **xfs**, входящая в состав XFree86 4.0; ее вы можете использовать даже при работе с ранними версиями XFree86. Если же вы по каким-либо причинам предпочтете работать с сервером xfsft, вы можете получить его, обратившись по адресу `http://www.dcs.ed.ac.uk/home/jec/programs/xfsft/`.

Описанные выше пакеты обрабатывают шрифты TrueType по-разному. Используемые в этих серверах алгоритмы обработки в свою очередь отличаются от алгоритмов, реализованных в системах Windows и MacOS. Применение разных принципов обработки приводит к тому, что символы одинакового размера, выведенные на одно и то же устройство с использованием разных серверов шрифтов, будут несколько различаться между собой. И xfstt, и **xfsft** обеспечивают достаточно хорошее качество воспроизведения символов. Если же при работе с каким-либо шрифтом возникнут проблемы или если внешний вид отображаемых символов не будет удовлетворять вас, вам придется рассмотреть вопрос об использовании другого сервера.

В системах Windows и MacOS реализована возможность сглаживания границ символов (anti-aliasing). Чтобы границы символов **выглядели** более ровно, вместо черного или белого цвета некоторые пиксели закрашиваются оттенками серого цвета. Если пользователю не понравится внешний вид обработанных подобным образом символов, он имеет возможность отключить средства сглаживания. В X Window до появления версии 4.0.2 сглаживание не поддерживалось. Чтобы включить сглаживание, необходимо выполнить дополнительные действия по настройке, которые описаны в документе `http://sdb.suse.de/en/sdb/html/chofman_ttf_72.html`.

При настройке различных серверов шрифтов, предназначенных для работы в системе Linux, выполняются практически одинаковые действия. Шрифты располагаются в специально предназначенных для них каталогах, и создаются файлы, которые описывают находящиеся в них шрифты. Затем сервер шрифтов конфигурируется для просмотра каталогов и предоставления необходимых шрифтов клиентам. Последующие разделы посвящены настройке **xfs** и **xfsft**. Конфигурация xfstt лишь незначительно отличается от этих серверов.

Конфигурация серверов шрифтов, установленная по умолчанию

После инсталляции Linux и XFree86 система создает конфигурационный файл XFree86 с именем **XF86Config** и помещает его в каталог /etc или **/etc/X11**. Как было сказано ранее, в этом файле содержатся записи Font Path, которые указывают на каталоги в файловой системе компьютера или на серверы шрифтов. Примеры записей в файле **XF86Config** приведены ниже.

```
FontPath "/usr/X11R6/lib/fonts/Type1/"
```

```
FontPath "unix/:7100"
FontPath "tcp/zapf:7100"
```

В конфигурационном файле, созданном по умолчанию, вы никогда не встретите такой набор записей. Приведенные выше строки лишь иллюстрируют три основных типа записей FontPath.

Первая строка определяет локальные шрифты, которые используются без участия сервера шрифтов. В большинстве версий Linux шрифты размещаются в нескольких каталогах, поэтому в файле XF86Config присутствует несколько строк, определяющих локальные каталоги со шрифтами (по одной строке на каждый каталог). Когда система получает команду найти шрифт с определенным именем, она просматривает каждый каталог по очереди до тех пор, пока шрифт не будет найдет или пока записи FontPath не будут исчерпаны.

Вторая строка иллюстрирует использование сервера шрифтов, расположенного на локальном компьютере. Ключевое слово `unix` указывает на то, что к серверу можно обращаться через сетевое соединение, используя гнезда UNIX. Число в конце записи (7100) определяет порт, по которому сервер принимает обращения. Если в вашем конфигурационном файле присутствует подобная запись, строки, непосредственно указывающие на каталоги со шрифтами, скорее всего будут отсутствовать. Вопросы настройки для использования дополнительных шрифтов рассматриваются далее в этой главе.

Третья строка определяет сервер шрифтов, доступный по сети. Ключевое слово tcp указывает на то, что к серверу можно обращаться с помощью стандартных средств TCP/IP. Имя после косой черты (в данном случае `zapf`) — это имя компьютера, на котором выполняется сервер шрифтов. (При необходимости вы можете задать полное доменное имя узла, например `zapf.threeroomco.com`.) Число, следующее за именем, определяет порт, по которому сервер принимает обращения.

Как локальный сервер шрифтов, так и сервер, доступный по сети TCPЛP, традиционно используют для приема обращений от клиентов порт 7100. (Иногда для обработки обращений от локальных программ применяется порт –1.) В некоторых случаях данное соглашение приводит к возникновению конфликтов. Это может случиться, если в системе выполняется программа, которая запускает сервер шрифтов с расширенными возможностями. В подобной ситуации вам следует использовать другой порт, например 7101 или 7102.

Порядок выполнения сервера шрифтов определяется содержимым конфигурационного файла. В большинстве случаев роль конфигурационного файла выполняет файл `/etc/X11/fs/conf`, но в некоторых системах вместо conf используется файл с именем `config`. В этом файле указывается расположение файлов шрифтов и определяются особенности работы сервера. Для запуска сервера шрифтов обычно применяются сценарии SysV, но если вы включаете сервер в систему, в котором по умолчанию его выполнение не предусмотрено, вы можете воспользоваться локальным сценарием запуска. В некоторых системах, например в Red Hat, сценарий SysV проверяет каталоги со шрифтами и определяет, должен ли быть обновлен список шрифтов. При необходимости список обновляется автоматически. Это существенно упрощает включение новых шрифтов, так как вам достаточно записать новые файлы в соответствующий каталог и перезагрузить сервер шрифтов. Если же утилита, автоматически генерирующая конфигурационный файл,

некорректно работает с каким-либо из шрифтов, вы можете запретить автоконфигурацию для одного или нескольких каталогов и создавать конфигурационный файл вручную.

Настройка сервера шрифтов для работы в сети

Если в дистрибутивном пакете по умолчанию предусмотрено выполнение сервера шрифтов, то в системе, как правило, принимаются меры для того, чтобы этот сервер не был доступен с остальных компьютеров. Доступ по сети блокируется исходя из соображений безопасности. Если же вы хотите, чтобы другие компьютеры могли обращаться к вашему серверу шрифтов, вам необходимо выполнить одно из следующих действий.

- Запустить второй экземпляр сервера шрифтов и обеспечить доступ к нему. Если вы будете использовать данный подход, вам придется модифицировать сценарий запуска или обеспечить выполнение второго сервера другим способом. Чтобы два экземпляра сервера использовали различные конфигурационные файлы, можно при вызове xfs указать опцию `-config` *`/путь_к_конфигурационному_файлу`*.

- Модифицировать конфигурационный файл сервера шрифтов и разрешить в нем доступ с других компьютеров. Такой способ более эффективен, но он непригоден в тех случаях, когда вам необходимо предоставлять локальным и удаленным клиентам различные наборы шрифтов.

Сервер шрифтов можно запустить даже на компьютере, на котором отсутствует система X Window. Чтобы это стало возможным, вам придется установить все программы, которые нужны для работы сервера. Несмотря на то что эти программы составляют основную часть X Window, X-сервер на этом компьютере запускать не обязательно.

Существуют два способа, позволяющие ограничить доступ к серверу шрифтов.

- Запрет установления TCP-соединения. В системе Red Hat 7.2 в файле `/etc/X11/fs/config` содержится строка `no-listen = tcp`, которая запрещает серверу принимать запросы на установление TCP-соединения. Если эта строка отсутствует, сервер принимает обращения от клиентов через порт 7100. Таким образом, чтобы обеспечить доступ к серверу, надо закомментировать данную строку, завершить работу сервера и запустить его снова. В системе Red Hat для остановки и запуска сервера может быть использован сценарий SysV с именем xfs.

- Использование порта `-1`. В системе Mandrake 8.1 сервер шрифтов по умолчанию настраивается для приема обращений через порт с номером –1. Такая настройка запрещает установление сетевых соединений с другими компьютерами. Для того чтобы изменить конфигурацию сервера, надо отредактировать сценарий запуска (обычно он содержится в файле `/etc/rc.d/init.d/xfs`) и изменить в нем номер порта. Найдите строку, которая начинается с `daemon xfs -port -1`, и замените число –1 на 7100 или на другой номер порта, который вы собираетесь использовать. Вам также надо отредактировать файл `/etc/XF86Config`, содержащийся в каталоге `/etc/X11` (в зависимости от используемого X-сервера этот файл может также называться `XF86Config` или `XF86Config-4`), и указать в нем, что обращение к серверу шрифтов должно осуществляться с использованием другого номера

порта. Найдите запись `FontPath`, ссылающуюся на `unix/:-1`, и замените -1 на 7100 или другой номер порта, который вы указали в сценарии запуска xfs. После этого вам придется завершить работу `xfs` и снова запустить его, а также перезапустить X-сервер (для этого можно использовать кнопку Restart X Server в окне регистрации Mandrake).

ВНИМАНИЕ Изменение конфигурации работающего сервера шрифтов представляет собой достаточно сложную задачу. Если сервер станет недоступным, прикладные программы, использующие шрифты, могут зависнуть. Поэтому лучше всего переконфигурировать сервер шрифтов, зарегистрировавшись в текстовом режиме. Если передать сценарию SysV, используемому для запуска `xfs` в системах Red Hat и Mandrake, опцию `restart`, изменения конфигурации не будут учтены. Поэтому вам необходимо остановить работу сервера, указав опцию `stop`, а затем снова запустить его с помощью опции `start`.

После того как вы настроите сервер шрифтов для работы в сети, вам следует изменить конфигурацию X-серверов и указать им на то, что за получением шрифтов они должны обращаться к установленному вами серверу шрифтов. Чтобы сделать это, надо включить в файл XF86Config на каждом из компьютеров новую запись FontPath. Примером такой записи может служить рассмотренная ранее запись FontPath, содержащая ключевое слово `tcp`. Очевидно, что в ней должны быть указаны имя компьютера, на котором выполняется сервер шрифтов, и номер порта, используемый этим сервером. В зависимости от набора шрифтов, установленных на каждом компьютере, вам, возможно, удастся удалить некоторые записи FontPath на клиентских системах, но при этом возрастет загрузка сервера шрифтов. Чтобы уменьшить число запросов к серверу шрифтов, запись FontPath, ссылающуюся на этот сервер, надо указывать после остальных записей данного типа. Это приведет к тому, что клиенты по возможности будут использовать файлы шрифтов, содержащиеся на локальных дисках.

ВНИМАНИЕ Если вы сконфигурируете систему так, что она сможет использовать только шрифты, предоставляемые внешним сервером, система станет неработоспособной при отсутствии доступа к серверу шрифтов. Поэтому желательно оставить записи FontPath, установленные при инсталляции системы, без изменений и добавить к ним запись, указывающую на сервер шрифтов. (В составе некоторых дистрибутивных пакетов поставляется огромное количество шрифтов. Возможно, вы захотите перенести некоторые из них на сервер. Однако основные шрифты, используемые системой X Window, лучше оставить на месте.)

Используя сервер шрифтов, нельзя упускать из виду вопросы защиты. Если сервер шрифтов работает в локальной сети, не подключенной к Internet, о безопасности системы можно не слишком беспокоиться, особенно если вы полностью контролируете все компьютеры в сети. Если же компьютер, на котором выполняется данный сервер, доступен из Internet, нельзя полностью исключать возможность незаконного проникновения в систему посредством сервера шрифтов. Сервер шрифтов — не очень сложная программа, и при обращении к нему пароль не указывается. Как и в любом сервере, в сервере шрифтов могут быть обнаружены ошибки, допускающие возможность взлома системы. Поэтому сеть, в которой работает сервер шрифтов, рекомендуется изолировать от остальной части Internet с помощью выделенного брандмауэра. Брандмауэр следует настроить так, что-

бы через порт, используемый сервером, могли обращаться только локальные машины. Вопросы конфигурирования iptables будут рассматриваться в главе 25.

Обеспечение доступа к шрифтам

Серверы шрифтов в основном предназначены для того, чтобы системный администратор мог, не затрачивая больших усилий, обеспечить доступ к шрифтам узлов сети, состоящей из компьютеров под управлением Linux и UNIX. (X-серверы в Windows, MacOS и других системах также иногда используют удаленные серверы шрифтов.) Для того чтобы обеспечить работу программ с использованием сервера шрифтов, необходимо выполнить два основных действия: задать пути к шрифтам для сервера шрифтов (эти пути отличаются от путей к шрифтам, задаваемых для X-сервера) и включить шрифты в соответствующие каталоги.

Указание путей к шрифтам

Пути к шрифтам задаются в конфигурационном файле сервера шрифтов (обычно это файл /etc/X11/fs/config или /etc/X11/fs/conf). Вместо ключевого слова FontPath, присутствующего в файле XF86Config, для указания шрифтов в файле config или conf используется ключевое слово catalogue. Пример записи, с помощью которой задаются пути к шрифтам, приведен ниже.

```
catalogue = /usr/X11R6/lib/X11/fonts/75dpi:unscaled,
            /usr/X11R6/lib/X11/fonts/Type1,
            /usr/X11R6/lib/X11/fonts/TrueType,
            /usr/X11R6/lib/X11/fonts/75dpi
```

Запись catalogue может занимать несколько строк. Каталоги в списке разделяются запятыми. После последнего каталога запятая не указывается, что является признаком окончания списка. Если имя каталога сопровождается выражением :unsealed, это означает, что растровые шрифты, находящиеся в этом каталоге, могут использоваться только в том случае, если их размеры совпадают с требуемыми размерами шрифтов. При отсутствии ключевого слова unscaled обработка растровых шрифтов производится следующим образом: если имя шрифта совпадает с именем, указанным в запросе, а файл, содержащий битовые карты нужного размера, отсутствует, шрифт приводится к требуемому размеру путем масштабирования (при этом качество шрифта обычно получается невысоким). Подобные соглашения действуют также при описании путей к шрифтам в файле XF86Config. В приведенном выше примере сервер шрифтов использует растровый шрифт из каталога 75dpi только в том случае, если его размер строго соответствует размеру, указанному в запросе. Если найти битовые карты подходящего размера не удается, система продолжает поиск шрифта в каталогах Type1 и TrueType, а затем возвращается к каталогу 75dpi и масштабирует один из находящихся там растровых шрифтов.

Изменяя список, задаваемый с помощью ключевого слова catalogue, вы можете добавлять или удалять каталоги со шрифтами. Так, например, если вы хотите добавить набор шрифтов, скопированный из Internet или полученный на компакт-диске, то должны поместить файлы в каталог и включить его в запись. При инсталляции некоторых серверов в конфигурационном файле указываются каталоги, которые содержат шрифты, поставляемые вместе с сервером. Если эти шрифты вам не нужны, удалите путь к ката-

логу из списка. (При этом необходимо следить за тем, чтобы после последнего каталога в списке не было запятой.)

Включение шрифтов в каталог

Настройка сервера шрифтов предполагает создание файлов описания каталогов. Файл описания каталогов имеет имя `fonts.dir`, а его содержимое представляется в следующем формате:

```
число
имя_файла_шрифта1 XLFD1
имя_файла_шрифта2 XLFD1
...
```

Первая строка содержит число, которое указывает, сколько шрифтов описано в данном файле. Каждая последующая строка описывает один шрифт. Все строки, кроме первой, начинаются с имени шрифта (например, `goodfont.ttf` или `t1f32.pfb`). Файл шрифта с указанным именем должен присутствовать в каталоге.

Каждый шрифт Type 1 реализуется в виде нескольких файлов. Файл PFB (Printer Font Binary — двоичный шрифт печати) содержит основную информацию о шрифте; имя этого файла обычно указывается в `fonts.dir`. Вместо PFB-файла вы можете задать в составе `fonts.dir` файл PFA (Printer Font ASCII — ASCII шрифт печати), в котором находятся те же данные, представленные в другом формате. Прочие файлы, определяющие шрифт, имеют расширения `.pfm`, `.afb` и `.afm`. Эти файлы необходимы для того, чтобы сервер шрифтов мог предоставлять клиентам шрифты Type 1.

Остальная часть строки представляет собой логический дескриптор шрифта (XLFD — X Logical Font Descriptor). Ниже приведен пример подобного дескриптора.

```
-bitstream-charter-medium-r-normal--0-0-0-0-p-0-iso8859-1
```

Логический дескриптор шрифта состоит из нескольких полей, разделенных дефисами (-). В полях дескриптора содержатся информация об изготовителе шрифта (bitstream); название семейства шрифтов (charter); "вес" шрифта (medium); сведения о том, представлены ли символы шрифта курсивом (r); ширина символов (normal); дополнительное имя стиля (в данном примере не используется); обобщенные данные о размере (строка, состоящая из нулевых значений, означает, что шрифт допускает масштабирование); сведения о том, является ли шрифт моноширинным или пропорциональным (p); средняя ширина (0 для масштабируемого шрифта) и кодировка (iso8859-1).

При составлении XLFD легко допустить ошибку, а в случае ошибки сервер не сможет предоставить шрифт клиенту. Поэтому для создания XLFD и даже для формирования всего файла `fonts.dir` предусмотрены специальные утилиты.

Для поддержки семейства шрифтов серверу требуется несколько файлов. Предположим, что в текстовом процессоре используется шрифт Times и возникает необходимость выделять фрагменты текста полужирным шрифтом или курсивом. Разновидности шрифта Times по сути являются отдельными шрифтами, для их представления используются отдельные файлы шрифтов, а в файле `fonts.dir` создаются XLFD. Многие текстовые процессоры и подобные им программы могут имитировать курсив и полужирный текст, но гораздо лучшие результаты получаются при использовании специальных шрифтов, в особенности это относится к символам, представленным курсивом.

Утилита, позволяющая создавать файл `fonts.dir` на основании файлов шрифтов Type 1, содержащихся в каталоге, называется `typelinst`. Эта утилита поставляется в составе многих дистрибутивных пакетов Linux, но по умолчанию она не инсталлируется. После установки данной программы надо сделать текущим каталог со шрифтами Type 1 и ввести следующую команду:

```
# typelinst
```

Программа `typelinst` просматривает файлы шрифтов, извлекает имена шрифтов и другую XLFD-информацию и на основании полученных данных создает файл `fonts.dir`. Данная программа также оповещает пользователя о ходе обработки шрифтов, например, она может сообщить, что на данный момент создана 21 запись в файле `fonts.dir`, одна из них описывает шрифт, изготовителя которого не удалось определить. Файл `fonts.dir`, созданный программой `typelinst`, можно отредактировать вручную и удалить несоответствия, например, выявить файлы шрифтов, принадлежащие одному семейству, но созданные различными производителями. X-программы используют информацию, содержащуюся в файле `fonts.dir`, и игнорируют данные в составе шрифтов. Поэтому изменение некоторых деталей файлов шрифтов не влияет на работу этих программ. Несоответствия, о которых шла речь выше, могут привести к возникновению проблем, в частности, если информация о производителе не совпадает, то, запросив шрифт, клиент может получить ту или иную его разновидность.

Программа аналогичного назначения создана и для работы с шрифтами TrueType. Эта программа называется `ttmkf`dir и входит в состав библиотеки FreeType, используемой `xfsf`t и XFree86 4.0. Программа `ttmkf`dir работает подобно программе `typelinst`, но позволяет задавать имя выходного файла посредством опции -o. Данная программа не включает в выходной файл сведения о шрифтах, в которых отсутствуют некоторые символы. Для того чтобы сведения об этих шрифтах были включены в выходной файл, необходимо задать опцию -c. Чтобы учесть изменения, внесенные в каталог со шрифтами, надо задать следующую команду:

```
# ttmkfdir -c -o fonts.dir
```

Если вы обнаружите, что при работе с некоторыми шрифтами возникают проблемы, вызовите эту же команду, но без опции -c. Список шрифтов станет короче, но оставшиеся в нем шрифты скорее всего будут работоспособны.

ВНИМАНИЕ Как было сказано выше, программы `typelinst` и `ttmkfdir` создают новый файл `fonts.dir` взамен существующего. Если вы добавляете шрифты в каталог, желательно создать резервную копию файла `fonts.dir`. Как вы уже знаете, автоматически созданный файл `fonts.dir` можно редактировать вручную. Копия файла поможет вам вспомнить, какие изменения уже были внесены в него.

После изменения файла `fonts.dir` необходимо остановить и снова запустить сервер шрифтов. Кроме того, надо либо перезапустить X-серверы, использующие сервер шрифтов, либо указать им на то, что список доступных шрифтов должен быть обновлен. Сделать это можно с помощью следующей команды:

```
# xset fp rehash
```

Если вы не сделаете этого, новые шрифты будут не доступны X-серверам. Если вы удалили шрифты и не оповестили об этом X-сервер, то при попытке получить отсутствующий шрифт, работа X-сервера будет приостановлена.

Сервер шрифтов с расширенными возможностями

Возможности, предоставляемые традиционным сервером шрифтов, не соответствуют требованиям, предъявляемым современными операционными системами и выполняющимися в них приложениями. Процедура включения новых шрифтов слишком трудоемкая, а качество текста часто оставляет желать лучшего, в особенности в тех случаях, когда путь к шрифту установлен некорректно или когда используются шрифты, заданные по умолчанию. Немаловажен и тот факт, что система поддержки шрифтов в X Window не предназначена для интеграции экранных шрифтов со шрифтами для печати. Эти и другие недостатки приводят к тому, что разработка приложений, для которых требуется высококачественный пользовательский интерфейс (например, текстовых процессоров или издательских систем), существенно затрудняется. Для разрешения этих проблем были созданы серверы шрифтов с расширенными возможностями. Многие из них встроены в другие приложения, но некоторые могут использоваться как независимые программы.

Среди серверов шрифтов с расширенными возможностями, предназначенных для использования в системе Linux, наибольшей популярностью пользуется FontTastic (`http://www.bitstream.com/categories/developer/fonttastic/`). Этот сервер распространяется на коммерческой основе, функционирует как сервер шрифтов X Window и предоставляет дополнительные возможности клиентам, ориентированным на взаимодействие с ним. FontTastic снабжает клиента дополнительной информацией, например, может при необходимости предоставить ему шрифты в контурном виде, необработанные данные шрифтов и сведения о кернинге. Эту информацию нельзя получить ни с помощью традиционного сервера шрифтов, ни в результате непосредственной обработки шрифтов X-сервером. Получив подобную информацию, приложение может осуществлять действия, которые невозможно выполнить из-за ограничений, накладываемых традиционными средствами обработки шрифтов X Window. Рассмотрим, например, работу текстового процессора. В обычных условиях пользователь указывает стандартный шрифт для документа. Когда приходит время вывести сформированный документ на печать, текстовый процессор обязан скопировать нужный шрифт на принтер либо каким-то образом сообщить принтеру о том, что тот должен использовать один из встроенных шрифтов. Если пользователь выбрал нестандартный шрифт, текстовый процессор может передать принтеру только битовые карты символов, так как именно в таком виде он сам получает эту информацию. Текстовый процессор должен каким-либо способом узнать разрешающую способность принтера, запросить символы с соответствующим разрешением у сервера шрифтов и вывести требуемые данные. При этом качество часто бывает невысоким.

Работая совместно с FontTastic, текстовый процессор может запросить у него необработанные данные шрифта и включить их в документ. Такой подход обеспечивает гораздо более высокое качество отображения и требует для этого меньших усилий. (При использовании шрифтов TrueType текстовый процессор, работающий с PostScript-принтером, может преобразовать их в формат Type 42. Кроме того, имея необработанные данные шрифта, он может эффективно восстановить шрифт. При этом достигаются го-

раздо лучшие результаты по сравнению с использованием битовых карт.) Взаимодействуя с FontTastic, текстовый процессор также получает дополнительные сведения (например, размер символов, кернинг и т. д.), что также позволяет улучшить внешний вид выводимых данных.

Поскольку FontTastic распространяется на коммерческой основе, этот сервер не стал стандартным инструментом для Linux. Однако на работу с ним ориентированы по крайней мере два широко используемых приложения: Corel WordPerfect Office 2000 (которое в в настоящее время уже не поддерживается разработчиком) и VistaSource ApplixWare Office (`http://www.vistasource.com/products/axware/`). Если вы запустите любую из этих программ, произойдет автоматическая инсталляция FontTastic и данный сервер будет выполняться в течение времени работы приложения. При необходимости FontTastic можно сконфигурировать так, что программы на других компьютерах смогут работать с ним как с обычным сервером шрифтов.

Использование FontTastic — не единственный способ разрешения проблемы несоответствия между выводом на экран и на принтер. Некоторые инструменты, например TeX, ориентированы на работу с принтером, а обеспечению качества при выводе на экран уделяется гораздо меньше внимания. TeX — это язык описания страниц. В некоторых случаях пользователь даже не видит на экране, как будет выглядеть составленный им документ, а лишь вводит требуемые директивы с помощью текстового редактора. Версии WYSIWYG TeX-редакторов появились сравнительно недавно.

Для того чтобы обеспечить согласование шрифтов на экране и на принтере, некоторые программы отказываются от взаимодействия со средствами поддержки шрифтов X Window. В качестве примера такой программы можно привести WordPerfect 8. Данный продукт непосредственно работает со шрифтами. Он самостоятельно растрирует контурные шрифты, отображает их на экране, а также создает битовые карты символов для вывода на принтер.

Еще один подход заключается в том, чтобы сообщить программе расположение файлов шрифтов. Имея такую информацию, программа может передать файлы шрифтов на принтер, обеспечивая таким образом согласование вывода данных на экран и на печать. Такой подход очень прост, но отсутствие в X Window некоторых средств специальной обработки шрифтов приводит к тому, что качество отображения текста на экране становится недопустимо низким. Если же приложение не в полной мере использует информацию, содержащуюся в файлах шрифтов, страдает также качество вывода на принтер. Ни один из описанных выше подходов не ориентирован на использование сетевых средств, поэтому в данной книге они не рассматриваются. Я упоминаю их лишь как альтернативу серверу FontTastic.

Резюме

Серверы шрифтов представляют собой достаточно простые программы. Они не требуют аутентификации и предоставляют пользователю лишь ограниченный доступ к ресурсам компьютера. Получая запрос от клиентской программы, сервер преобразует исходный файл шрифтов в битовую карту, посредством которой осуществляется вывод символов определенного размера. В большинстве случаев в роли клиента выступает X-сервер, расположенный на другом компьютере. Сервер шрифтов, выполняющийся на локальном компьютере, позволяет работать с форматами файлов шрифтов, которые не

поддерживаются X-сервером; кроме того, при этом появляется возможность использования сервера шрифтов с расширенными возможностями, предоставляющего приложению дополнительную информацию. Сервер шрифтов, поддерживающий обращение по сети, позволяет осуществлять централизованное управление шрифтами. Шрифт, установленный на таком сервере, становится доступен многим компьютерам в сети. Процедура установки шрифтов достаточно трудоемкая и требует внимания, но при использовании сервера шрифтов она выполняется лишь один раз.

Глава 16

Удаленное администрирование системы

Средства удаленной регистрации, которые рассматривались в главах 13 и 14, позволяют пользователям запускать программы с удаленного компьютера. Эти инструменты можно использовать для регистрации в системе и управления ею. Существуют также специализированные инструменты, предназначенные для решения задач удаленного администрирования. Некоторые из них предоставляют дружественный интерфейс, упрощая тем самым работу начинающих администраторов. Даже если интерфейс сложен для восприятия, работать с этими инструментами все же проще, чем вручную редактировать конфигурационные файлы. Справочная информация, предоставляемая инструментами удаленного администрирования, помогает в работе даже опытным специалистам. В данной главе рассматриваются два универсальных инструмента (Linuxconf и Webmin), а также утилита, ориентированная на работу с единственным сервером (Samba Web Administration Tool, или SWAT). Кроме того, в конце главы обсуждаются вопросы безопасности при использовании инструментов удаленного администрирования.

Использование средств удаленного администрирования

В некоторых случаях возникает необходимость выполнять администрирование системы с удаленного компьютера. Справиться с этой задачей помогают специализированные инструменты. Эти средства можно использовать и локально. Все программы удаленного администрирования предоставляют Web-интерфейс, с помощью которого можно работать, используя в качестве клиентской программы обычный Web-броузер. Linuxconf, помимо Web-интерфейса, поддерживает также альтернативные средства взаимодействия с пользователем. Учитывая, что для удаленного администрирования подходит любой из серверов удаленной регистрации, рассмотренных в главах 13 и 14, становится ясно, что основное преимущество специализированных инструментов удаленного администрирования — это интерфейс, упрощающий работу администратора.

Многие начинающие администраторы испытывают большие трудности при установке конфигурации посредством редактирования текстовых файлов, поэтому они с радостью применяют рассматриваемые здесь инструменты. А поскольку вряд ли отыщется специалист, который детально представляет себе структуру всех конфигурационных файлов, используемых в системе, то инструменты администрирования могут помочь в работе даже квалифицированному администратору. В идеале данные средства призваны исключить ошибки, допускаемые при формировании конфигурационных файлов, и выявлять некорректные сочетания параметров. Но на практике дело обстоит несколько хуже. Неправильно сконфигурировать систему можно даже с помощью специализированного инструмента, поэтому необходимо знать функционирование системы и структуру конфигурационных файлов. Никакие инструментальные средства не заменят специальные знания и опыт администратора.

Программы Linuxconf и Webmin, рассматриваемые в данной главе, представляют собой универсальные средства, ориентированные на работу с различными подсистемами. Они хорошо подходят для определения основных параметров системы, но не поддерживают специфические установки сложных подсистем. Для некоторых из подсистем разработаны специальные программы администрирования. Один из таких инструментов, SWAT, предназначенный для работы с Samba, будет обсуждаться ниже. SWAT поддерживает практически все параметры Samba, однако для работы с другими компонентами Linux этот инструмент непригоден.

Все рассматриваемые в данной главе средства удаленного администрирования используют протокол HTTP, поэтому взаимодействие с ними может осуществляться посредством Web-броузера. Порты, по которым данные инструменты принимают обращение клиентской программы, отличаются от стандартных портов, используемых Web-серверами. При выполнении администрирования с удаленного узла необходимо знать требуемый номер порта и указывать его в составе URL.

Использование средств удаленного администрирования для настройки различных версий Linux

Все инструментальные средства, рассматриваемые в этой главе, могут работать с различными версиями Linux. Основная трудность, возникающая при этом, состоит в том, что в программе администрирования должны быть учтены характерные особенности каждой версии системы. Так, например, в разных дистрибутивных пакетах запуск одного и того же сервера осуществляется различными способами, в качестве суперсервера в одних версиях Linux используется `inetd`, а в других — `xinetd`, нумерация сценариев запуска SysV различается в зависимости от версии операционной системы и т. д. Для инструмента, ориентированного на работу с конкретной реализацией Linux (например, для YaST, используемого в SuSE), эти различия не имеют значения, а для универсальных инструментов, таких как Linuxconf и Webmin, они чрезвычайно важны.

Для обеспечения работы Linuxconf и Webmin с разными версиями Linux используются конфигурационные модули. Каждый из этих инструментов сам по себе является не более чем базовой программой, которая осуществляет взаимодействие с клиентом и позволяет пользователям выбирать опции, вводить строки символов и выполнять другие подобные

действия. Для выполнения конкретных действий с конфигурационными файлами используются модули, в которых содержится подробная информация о системе: расположение файлов, сведения об их структуре и другие данные, имеющие отношение к настройке системы. Существуют модули, предназначенные для установки общей конфигурации системы (например, для конфигурирования сценариев запуска SysV и формирования файла `/etc/inittab`) и для настройки конкретных серверов (например, Apache, sendmail и Samba). Если в состав системы входит инструмент Linuxconf или Webmin, дистрибутивный пакет содержит модули для серверов, поставляемых вместе с системой. При установке нового сервера необходимые модули чаще всего поставляются в инсталляционном пакете. Если вы копируете программу удаленного администрирования с Web-узла, вы найдете на том же узле набор модулей для вашей версии системы.

Конфигурационные модули обеспечивают большую гибкость таких программ, как Linuxconf и Webmin, но они же могут стать источником проблем. Если модули не отражают изменения, внесенные в систему, инструмент администрирования будет работать ненадежно. Не исключено, что конфигурационные файлы не станут обновляться или при их обновлении появятся ошибки. Использование устаревших модулей может даже привести к тому, что сама программа администрирования не будет выполняться. Эти проблемы особенно заметны при работе с Linuxconf в системе Red Hat. Чтобы исключить их, в состав Red Hat начиная с версии 7.1 включается соответствующим образом настроенная программа Linuxconf.

Инструменты администрирования, ориентированные на конкретный сервер (такие как SWAT), имеют более простую структуру. В частности, SWAT оперирует лишь с одним конфигурационным файлом (`smb.conf`). Несмотря на то что расположение этого файла может изменяться, SWAT обычно компилируется вместе с Samba, поэтому сведения о местонахождении конфигурационного файла учитываются автоматически. Поэтому в инструментах, подобных SWAT, конфигурационные модули не используются.

Выполнение Linuxconf на удаленном компьютере

Традиционно программа Linuxconf поставляется с Red Hat и Mandrake, однако она разрабатывалась не только для них. Данный инструмент может также использоваться в других системах, например в Caldera, Debian, Slackware и SuSE. Обратившись по адресу `http://www.solucorp.qc.ca/linuxconf/`, вы можете скопировать архив, содержащий код Linuxconf.

Linuxconf включает модули, обеспечивающие выполнение как на локальном, так и на удаленном компьютере. При работе на локальной машине Linuxconf предоставляет текстовый или графический интерфейс, а для взаимодействия с данной программой при выполнении ее на удаленном компьютере используется Web-интерфейс. Web-интерфейс предоставляет те же возможности, что и текстовый или графический интерфейс, используемый при работе на локальной машине. По умолчанию поддержка Web-интерфейса обычно запрещена.

Настройка Linuxconf для выполнения на удаленном компьютере

По умолчанию Linuxconf настраивается для работы в текстовом и в одном из двух графических режимов (в большинстве дистрибутивных пакетов поддерживается GNOME-Linuxconf, но Solucorp использует другой режим). Разрабатывается также Java-интерфейс. Поскольку поддержка Web-интерфейса по умолчанию запрещена, вам надо разрешить ее самостоятельно. Для этого необходимо обеспечить выполнение Linuxconf в режиме сервера и сконфигурировать его для приема обращений по сети.

Запуск сервера Linuxconf

Для запуска Linuxconf в режиме сервера чаще всего используется суперсервер. В зависимости от версии Linux это может быть `inetd` или `xinetd`. Как вы уже знаете, для запуска программы посредством суперсервера необходимо включить в файл `/etc/services` описание порта. Для Linuxconf такое описание имеет следующий вид:

```
linuxconf 98/tcp
```

Кроме того, необходимо сконфигурировать суперсервер так, чтобы он вызывал программу `linuxconf` с указанием опции `-http`. Соответствующая запись в файле `/etc/inetd.conf` имеет следующий вид:

```
linuxconf stream tcp wait root /bin/linuxconf linuxconf --http
```

Если в вашей системе используется `xinetd` и Linuxconf входит в комплект поставки, в каталоге `/etc/xinetd.d` скорее всего содержится файл `linuxconf-web` или `linuxconf`, предназначенный для запуска данного сервера. Чтобы удаленный доступ к Linuxconf был возможен, следует убедиться, что в этот файл не включена строка `disable = yes`, если же она присутствует в файле, вам надо заменить значение yes на no.

Авторизация удаленного доступа

Сконфигурировав суперсервер для запуска Linuxconf, вы можете обращаться к данной программе, выполняющейся в режиме сервера, с помощью Web-броузера. Для этого надо ввести имя узла и указать порт 98. Например, для того, чтобы начать администрирование компьютера `remote.threeroomco.com`, вы должны указать URL `http://remote.threeroomco.com:98`. Web-броузер может выполняться на любой платформе, поэтому выполнять администрирование системы Linux можно, работая на компьютере под управлением Linux, UNIX, Windows, MacOS или любой другой системы.

Несмотря на то что сервер Linuxconf отвечает на запрос Web-броузера, по умолчанию никаких полезных действий он не выполняет. Конфигурация, установленная изначально, позволяет Linuxconf передать клиент-программе Web-страницу с кратким описанием данной программы, но если вы щелкнете на кнопке Enter, вы получите страницу с сообщением об ошибке. Такая мера предосторожности предпринята для повышения уровня защиты системы. Чтобы получить возможность выполнять реальные операции по администрированию системы, надо разрешить программе обрабатывать обращения по сети. Возможно, вы решите ограничить круг компьютеров, которым разрешен доступ к серверу Linuxconf, теми узлами сети, с которых вы собираетесь осуществлять администрирование системы. Необходимые изменения в конфигурацию программы проще всего внести, запустив Linuxconf локально. Для этого выполните следующие действия.

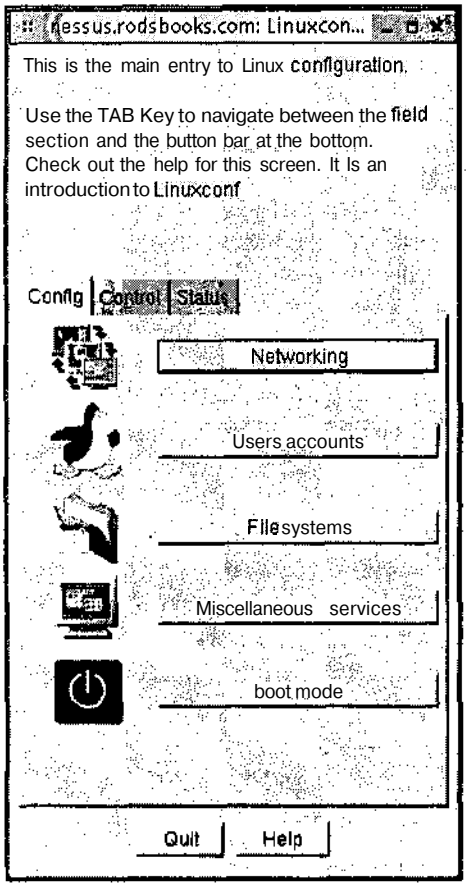


Рис. 16.1. Окно Linuxconf может выглядеть по-разному, но оно предоставляет один и тот же набор возможностей

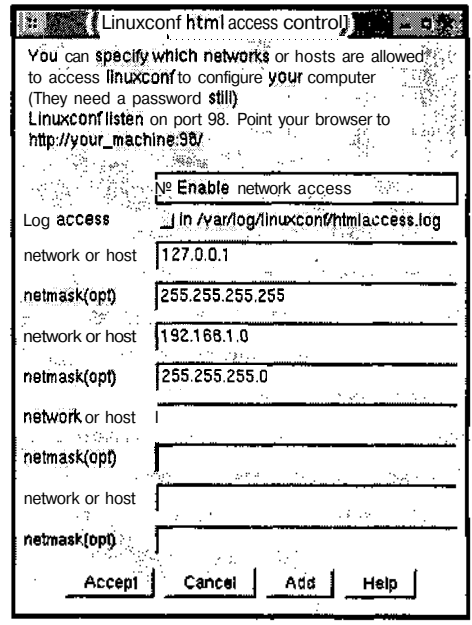


Рис. 16.2. В этом окне можно разрешить доступ к Linuxconf по сети и указать, какие системы могут обращаться к серверу

1. Запустите программу `linuxconf` от имени пользователя `root`. Для этого введите команду `linuxconf`. Если вы работаете в текстовом режиме или если средства поддержки графического интерфейса Linuxconf отсутствуют, на экране отобразится текстовое меню. Если же вы работаете в системе X Window и в ней поддерживается графический интерфейс Linuxconf, вы увидите окно Linuxconf. Окно Linuxconf, отображаемое в системе Mandrake, показано на рис. 16.1. В других версиях Linux это окно может выглядеть несколько по-другому, но оно позволяет выполнить те же действия.

2. Выберите пункт меню Config⇨Networking⇨Misc⇨Linuxconf Network Access Options. В результате вам станут доступны опции, показанные в окне на рис. 16.2, но поля будут пустыми. (Разные системы отображают данную информацию различными способами.)

3. Установите флажок опции Enable Network Access. В результате Linuxconf станет обрабатывать обращения по сети.

4. В первом поле network or host введите адрес 127.0.0.1, а в первом поле netmask(opt) — значение маски 255.255.255.255. Этим вы укажете Linuxconf на необходимость обрабатывать обращения с локального компьютера.

5. Во втором поле network or host введите адрес компьютера, с которого вы собираетесь выполнять администрирование Linux. В следующем за ним поле netmask(opt) задайте маску подсети. Например, значение, показанное на рис. 16.2, сообщает о том, что администрирование системы может осуществляться с любого компьютера, принадлежащего сети 192.168.1.0/24.

6. Повторите действия, выполняемые на предыдущем шаге, для всех компьютеров (или сетей), с которых вы собираетесь обращаться к Linuxconf.

7. Переходя от окна к окну и активизируя кнопки Accept, Dismiss и Quit, сохраните внесенные изменения и завершите работу с Linuxconf. Возможно, вы получите сообщение о том, что система не синхронизирована с текущей конфигурацией. Чтобы изменения были учтены, щелкните на Do It.

После выполнения указанных действий программа Linuxconf будет настроена для обработки обращений по сети с локального компьютера и со всех компьютеров, которые вы указали на этапах 5 и 6. Пользователи, работающие на других узлах сети, увидят лишь первую страницу, на которой кратко описывается программа Linuxconf. (Если вы применяете брандмауэр, внешние пользователи не увидят даже этой информации.)

Обращение к Linuxconf с помощью Web-броузера

Чтобы воспользоваться Web-интерфейсом, предоставляемым Linuxconf, вам надо задать в поле, предназначенном для ввода URL, значение `http://имя_узла:98`. Вместо имени узла можно указать его IP-адрес. В ответ вы получите описание Linuxconf. На этой же странице присутствует кнопка Enter, после щелчка на которой появится диалоговое окно, предназначенное для ввода пользовательского имени и пароля. Вам необходимо зарегистрироваться как root или указать имя другого пользователя, обладающего полномочиями на выполнение действий по администрированию Linux. Основное меню Linuxconf показано на рис. 16.3.

Пункты меню, показанного на рис. 16.3, активизируются так же, как и обычные гипертекстовые ссылки, расположенные на любой Web-странице. После первых одного-двух щелчка отобразятся новые меню, но в конце концов вы увидите страницу, позволяющую вводить текст в полях редактирования, устанавливать и сбрасывать флажки опций и выполнять другие подобные действия. Например, если в окне, показанном на рис. 16.3, вы щелкнете на Networking в области Config, а на странице, которая отобразится после этого щелчка, вы активизируете ссылку Linuxconf Network Access в области Misc, то увидите Web-страницу, изображенную на рис. 16.4. На этой странице можно выполнять такие же действия, как и диалоговом окне, изображенном на рис. 16.2. Вы можете запретить доступ по сети или изменить набор компьютеров, которые имеют право обращаться к Linuxconf.

Web-интерфейс Linuxconf отличается от графического интерфейса, рассмотренного выше, тем, что, при работе с Web-броузером от вас требуется меньше действий для под-

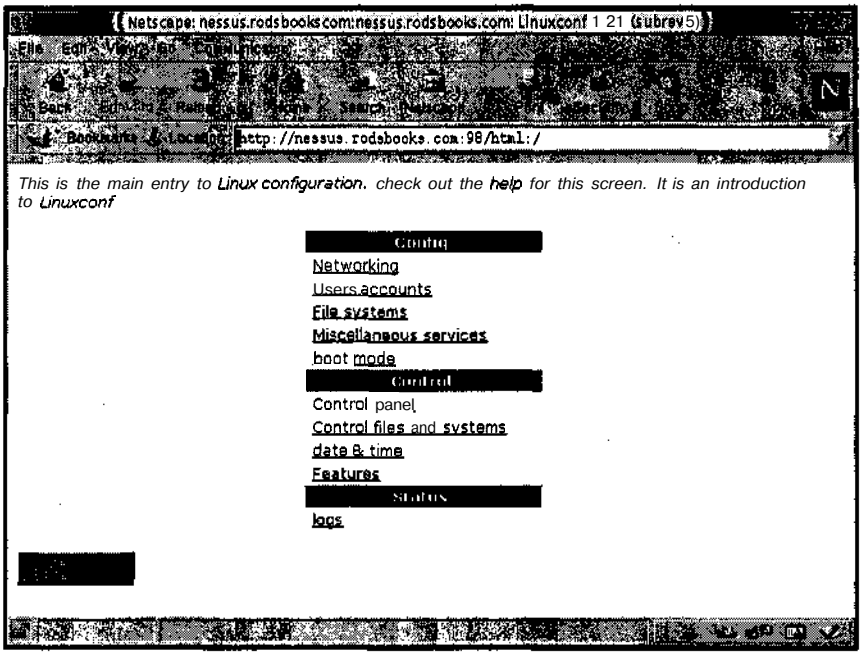


Рис. 16.3. Главное меню Linuxconf, отображаемое средствами Web, содержит тот же набор пунктов, что и меню, которое выводится при работе на локальном компьютере

тверждения внесенных изменений. После щелчка на кнопке Accept (эта кнопка не показана на рис. 16.4, но вы увидите ее, если воспользуетесь полосой прокрутки) сделанные вами установки будут приняты программой. Чтобы сделать то же самое, работая с графическим интерфейсом Linuxconf, надо щелкнуть на нескольких кнопках, расположенных в различных окнах.

ВНИМАНИЕ Закончив установку параметров посредством Web-страницы, предоставляемой сервером Linuxconf, необходимо щелкнуть на кнопке Accept, в противном случае внесенные изменения не будут учтены. Если вы хотите, чтобы сервер Linuxconf проигнорировал выполненные вами действия, завершите работу с Web-страницей щелчком на кнопке Back.

Как видно на рис. 16.1 и 16.3, модули Linuxconf организованы в виде иерархической структуры. Самый простой способ ознакомиться с набором возможностей Linuxconf — просмотреть Web-страницы, отображающиеся при активизации гипертекстовых ссылок. Если вы не хотите изменить конфигурацию системы, не активизируйте кнопку Accept, а заканчивайте работу с очередной страницей щелчком на кнопке Back. Не исключено, что некоторые модули будут расположены не там, где вы ожидаете их увидеть. Возможно, вам не удастся установить с помощью Linuxconf все необходимые конфигурационные параметры. Это может произойти из-за отсутствия требуемого модуля либо потому, что в Linuxconf не учтены некоторые особенности вашей системы. Например, если Linuxconf ожидает, что конфигурационный файл находится в каталоге /etc, а на самом деле

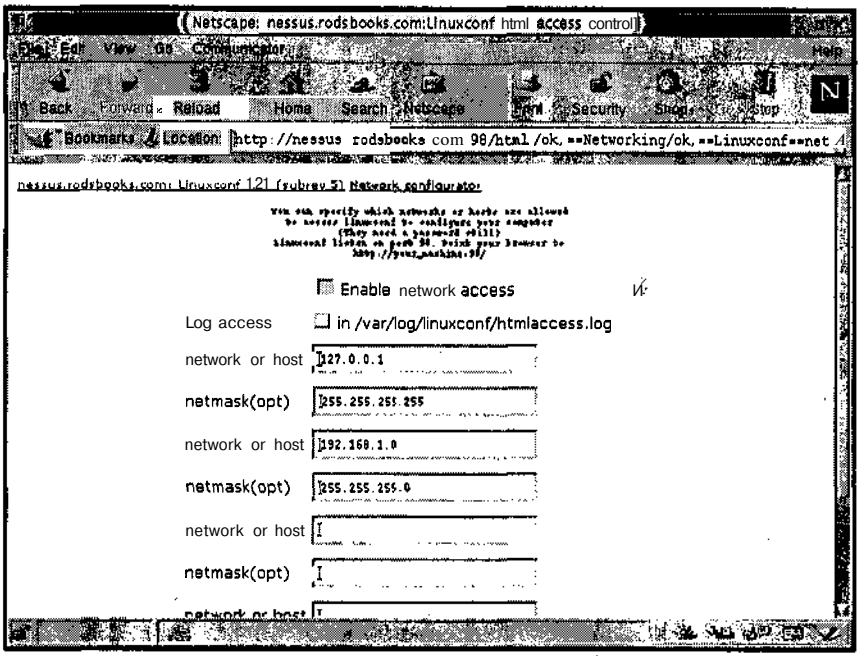


Рис. 16.4. Модули Linuxconf предоставляют поля редактирования, флажки опций, списки и другие интерфейсные элементы, предназначенные для установки конфигурации

этот файл расположен в `/usr/local/etc`, действия по настройке не будут выполнены. Источником проблем при работе с Linuxconf может стать несоответствие версий системы. Возможно, что Linuxconf не сможет интерпретировать некоторые новые опции либо предпримет попытку установить параметры, которые уже не поддерживаются в системе.

После окончания работы с Linuxconf желательно завершить выполнение Web-броузера. Со временем Linuxconf прекратит сеанс взаимодействия по тайм-ауту, но до тех пор любой пользователь, случайно получивший доступ к вашему компьютеру, сможет изменить конфигурацию системы. Не исключено, что вы и сами непреднамеренно измените некоторые параметры.

Удаленное администрирование с помощью Webmin

Инструмент Webmin (`http://www.webmin.com/webmin/`) позволяет решить те же задачи, что и Linuxconf. Он упрощает действия администратора по конфигурированию системы и предназначен для настройки различных версий Linux. Webmin обеспечивает работу не только с системой Linux, но и с некоторыми версиями UNIX (например, Solaris и FreeBSD), а также с MacOS. (Полный список поддерживаемых систем находится по адресу `http://www.webmin.com/webmin/support.html`.) Настройка системы с помощью Webmin во многом напоминает работу с Linuxconf. Поскольку Web-

min изначально создавался как сетевой инструмент, конфигурирование этой программы для обработки обращений с удаленного компьютера осуществляется несколько проще по сравнению с Linuxconf.

Настройка Webmin

Из всех версий Linux, которые обсуждались в данной книге, только Mandrake поставляется с Webmin (планируется включение данного инструмента в комплект Debian 3.0). При работе с другими версиями системы вам придется скопировать Webmin с Web-узла. Пакет Webmin доступен как в формате RPM, так и в виде tar-архива. Для установки Webmin с помощью RPM приходится затрачивать меньше усилий, так как при этом автоматически выполняется сценарий, который определяет версию системы и автоматически настраивает сервер. Если вы используете tar-архив, вам потребуется вручную запустить содержащийся в нем сценарий и ответить на ряд вопросов по системе. Процедура установки Webmin с помощью tar-архива описана ниже.

1. Зарегистрировавшись в системе как root, сделайте текущим каталог, в котором должен находиться подкаталог Webmin. В документации на данный продукт рекомендуется устанавливать его в каталоге /usr/local, но при желании вы можете разместить его в другой позиции файловой системы, например в каталоге /opt.

2. Распакуйте архив Webmin, вызвав для этого команду tar xvfz /путь/webmin-версия.tar.gz. В результате выполнения этой команды будет создан подкаталог webmin-версия, в котором разместятся файлы Webmin.

3. Перейдите в созданный каталог Webmin по команде cd webmin-версия.

4. Запустите сценарий инсталляции по команде ./install.sh. Этот сценарий задаст вам ряд вопросов о системе, например, вам придется сообщить путь к интерпретатору Perl. Очень важно правильно ответить на вопрос о версии системы. Необходимо также указать имя пользователя, имеющего право выполнять администрирование системы, и пароль (эти сведения будут впоследствии использоваться при обращении к серверу Webmin). По окончании выполнения сценарий запустит Webmin, и вы сразу же сможете приступить к работе с данным инструментом.

Инструмент Webmin написан на Perl, поэтому компилировать программу не приходится. Один и тот же пакет можно использовать в различных системах, независимо от архитектуры процессора. Чтобы программа Webmin работала, в системе должен присутствовать интерпретатор Perl, однако это требование по умолчанию выполняется во всех версиях Linux.

Конфигурация самой программы Webmin определяется содержимым файлов, находящихся в каталоге /etc/webmin (если вы используете для инсталляции Webmin tar-архив, то можете указать другое расположение конфигурационных файлов). Вероятнее всего, вам не понадобится модифицировать эти файлы, но если вы захотите изменить конфигурацию программы, вам скорее всего придется отредактировать файлы config и miniserv.conf. В этих файлах находятся такие сведения, как номер порта, через который Webmin принимает обращения, и тип системы. Кроме того, в файле miniserv.users содержатся также пользовательское имя администратора и пароль.

(Если вы инсталлируете Webmin с помощью RPM, программа использует в качестве имени администратора root и читает пароль из файла /etc/passwd или /etc/shadow. Если установка Webmin производится посредством tar-архива, имя пользователя и пароль надо ввести вручную.) В подкаталогах каталога /etc/webmin содержится информация о серверах и подсистемах, поддерживаемых Webmin.

В большинстве случаев запуск сервера Webmin осуществляется посредством сценария SysV. Этот сценарий, в свою очередь, использует для запуска Perl-кода Webmin сценарий /etc/webmin/start.

Использование Webmin

Для того чтобы обратиться к Webmin, надо выполнить те же действия, что и при обращении к Linuxconf. Если вы зададите URL Web-сервера (указав при этом номер порта 10000), вам будет предложено ввести имя пользователя и пароль, после чего в окне Web-броузера отобразится Web-страница, показанная на рис. 16.5. Подобно Linuxconf, компоненты Webmin организованы в виде иерархии категорий, но глубина вложенности подкатегорий меньше, чем в Linuxconf. Большинство средств, которые могут потребоваться вам при работе, расположены на вкладках System и Servers. Вкладка Webmin используется для настройки самой программы Webmin. Вкладка Hardware предназначена для согласования конфигурации программы с конфигурацией аппаратных средств (например, для указания информации о разделах), а на вкладке Others расположены элементы различного назначения.

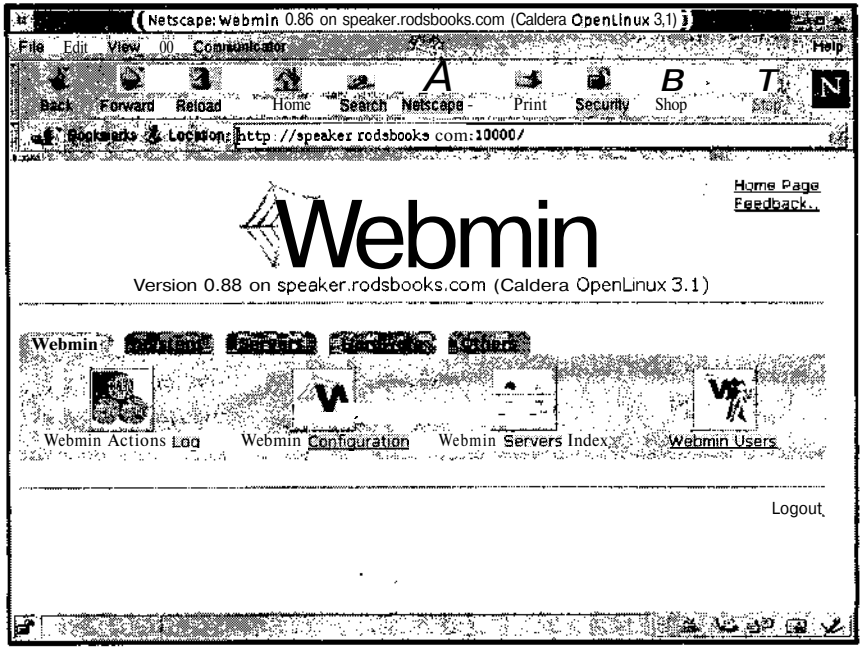


Рис. 16.5. Основная страница Webmin позволяет выбрать общую категорию, а в ней указать подсистему для настройки

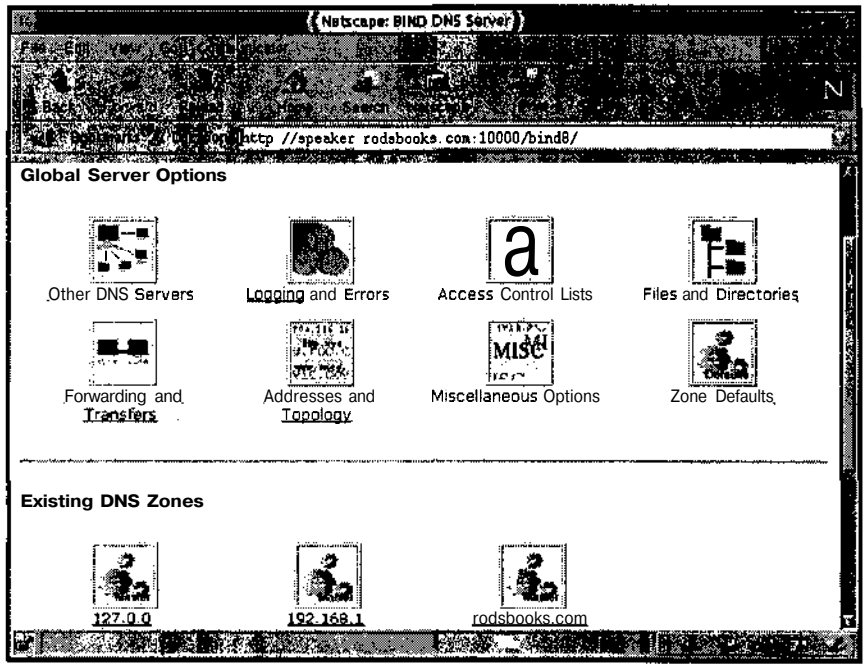


Рис. 16.6. Многие серверы и подсистемы предоставляют собственный набор ссылок. Такой подход позволяет ограничить размер страниц, используемых для настройки компонентов системы

После щелчка на пиктограмме, соответствующей серверу или подсистеме, Webmin предоставит Web-страницу, содержащую список компонентов, либо страницу, позволяющую непосредственно выполнять действия по настройке. Например, страница, соответствующая серверу DNS, содержит ссылки, указывающие на Web-страницы, которые можно использовать для протоколирования, управления файлами и выполнения других действий. Кроме того, как видно на рис. 16.6, для каждой зоны, обслуживаемой сервером DNS, создается отдельная ссылка. Переходя по ссылкам, вы получите страницу, содержащую поля редактирования, флажки опций, списки и другие элементы. Пример такой страницы приведен на рис. 16.7. Выполнив необходимые действия по настройке, щелкните на кнопке Save, чтобы сохранить внесенные изменения. Многие конфигурационные модули предоставляют кнопку Apply Changes, при активизации которой сервер учитывает изменения конфигурации. Другие модули, в зависимости от того, выполняется ли связанный с ними сервер, отображают на странице кнопку Stop или Start. Чтобы ваши установки были приняты, вам надо щелкнуть на кнопке Stop, а затем на кнопке Start.

Список модулей Webmin может включать серверы, которые не выполняются и даже не установлены в системе. Если вы активизируете ссылку, соответствующую такому серверу, вы получите сообщение, что Webmin не может найти конфигурационный файл. В этом же сообщении высказывается предположение о том, что модуль сконфигурирован неверно или сервер не установлен. Если вы инсталлировали сервер с помощью нестандартного пакета, щелкните на ссылке Module Configuration, расположенной в окне с сообщением, в результате чего вы увидите Web-страницу, предназначенную для настройки конкретно-

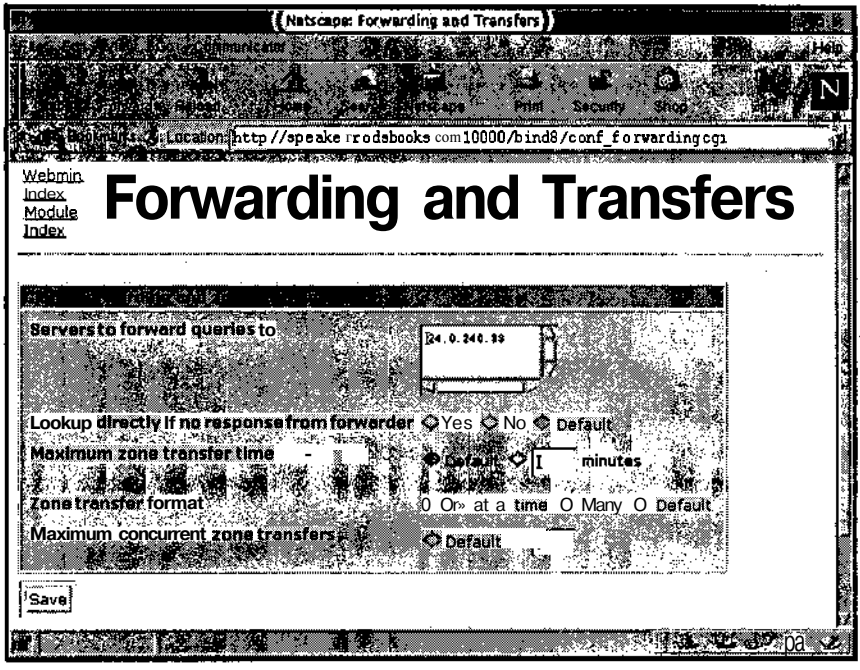


Рис. 16.7. Webmin предоставляет приблизительно такие же средства настройки, как и Linuxconf

го модуля. Если вы знаете расположение конфигурационных файлов, можете настроить Webmin для администрирования вашего сервера.

Обычно с Webmin поставляется более полный набор модулей, чем с Linuxconf. В некоторых дистрибутивных пакетах Webmin работает лучше Linuxconf, в других Linuxconf предпочтительнее. Желательно опробовать оба инструмента и выбрать тот из них, который больше подходит для вашей системы.

Окончив работу с Webmin, щелкните на ссылке Logout, расположенной на основной странице (рис. 16.5). В результате сеанс работы с Webmin будет завершен, и вам не придется закрывать Web-броузер, как это рекомендуется делать при использовании Linuxconf.

Настройка сервера Samba с помощью SWAT

SWAT (Samba Web Administration Tool), в отличие от Linuxconf и Webmin, является специализированным инструментом. Как следует из названия, SWAT предназначен для администрирования лишь сервера Samba. В результате многие проблемы, связанные с инсталляцией и настройкой для работы с конкретной версией операционной системы, типичные для продуктов Webmin и Linuxconf, не возникают, а сама программа SWAT достаточно полно охватывает набор конфигурационных параметров Samba. SWAT удобно использовать для администрирования выделенных серверов Samba, в особенности этот продукт полезен тем администраторам, которые не имеют достаточного опыта работы и чувствуют себя неуверенно, редактируя текстовые конфигурационные файлы. SWAT

иногда применяют в работе даже квалифицированные администраторы, так как этот инструмент избавляет их от необходимости помнить синтаксис записей в составе конфигурационных файлов. Активизируя ссылки Help, расположенные на Web-странице рядом с интерфейсными элементами, предназначенными для редактирования параметров, вы получите данные из справочной системы, которые описывают соответствующие записи в файле `smb.conf`. Недостатком SWAT является тот факт, что этот продукт удаляет комментарии из файла `smb.conf` и не поддерживает параметр `include`, включающий дополнительные конфигурационные файлы. Поэтому, когда необходимо устанавливать сложную конфигурацию Samba, опытные администраторы предпочитают обходиться без помощи SWAT.

Запуск SWAT

Функции сервера SWAT реализованы в программе `swat`. Для ее запуска может быть использован любой из способов, описанных в главе 4, но чаще всего `swat` запускается посредством суперсервера. Соответствующая запись в файле `/etc/inetd.conf` имеет следующий вид:

```
swat stream tcp nowait.400 root /usr/sbin/tcpd /usr/sbin/swat
```

Если в операционной системе используется суперсервер `xinetd`, для SWAT создается файл `/etc/xinetd.d/swat`. Чтобы обеспечить работу SWAT, необходимо убедиться в том, что в данном файле отсутствует запись `disable = yes`. Если такая строка содержится в файле, ее надо удалить либо заменить значение yes на по. Независимо от того, используется ли в системе `inetd` или `xinetd`, для того, чтобы SWAT стал доступен, вам надо перезапустить суперсервер.

Иногда SWAT включается в состав пакетов Samba (samba, `samba-common`, `samba-server` и т. д.), в других случаях поставляется в отдельном пакете (обычно он называется `swat` либо `samba-swat`). В системах Mandrake, Slackware, SuSE и TurboLinux SWAT интегрируется в состав Samba, а в системах Caldera, Debian и Red Hat SWAT применяется как независимый пакет.

По умолчанию SWAT использует порт 901. При работе как с `inetd`, так и с `xinetd` в файле `/etc/services` должна присутствовать следующая запись:

```
swat 901/tcp
```

В большинстве случаев данная запись включается в этот файл по умолчанию.

Использование SWAT

После установки SWAT в систему обращаться к этому серверу следует так же, как и к серверам Webmin и Linuxconf, но в составе URL надо указывать порт 901. Например, чтобы использовать SWAT для администрирования сервера Samba, расположенного на узле сети `samba.threeroomco.com`, вам надо задать в поле ввода URL броузера строку `http://samba.threeroomco.com:901`. Как и при работе с другими серверами, вы можете использовать Web-броузер, выполняющийся на любой платформе.

Samba обрабатывает запросы с указанием имен NetBIOS, поэтому SMB/CIFS-**клиенты** могут пользоваться соответствующим механизмом преобразования имен. SWAT не содержит модуля подобного назначения, но если клиентские компьютеры поддерживают имена NetBIOS, сервер SWAT будет доступен не только по доменному имени, но и по имени NetBIOS. Для этого сервер Samba должен выполняться в системе. Как правило, клиенты Windows настроены для поддержки имен NetBIOS, а клиенты, работающие в системе Linux, могут использовать только доменные имена.

Подобно другим серверам удаленного администрирования, рассматриваемым в данной главе, при первом обращении к серверу SWAT он запросит пользовательское имя и пароль. Для того чтобы получить максимальные привилегии, надо указать имя `root`. (Работая с инструментом SWAT, необходимо указывать пароль, используемый в системе Linux, а не пароль, применяемый для регистрации на сервере Samba.) Если вы зарегистрируетесь как обычный пользователь, SWAT позволит лишь просматривать содержимое конфигурационного файла и вносить те изменения, которые разрешены для этого пользователя. Так, например, каждый пользователь имеет право изменять свой пароль. Независимо от того, зарегистрировались ли вы под именем `root` или как обычный пользователь, SWAT вернет броузеру исходную страницу, показанную на рис. 16.8. После щелчка на пиктограмме или пояснительном тексте к ней вы можете вызвать другие страницы: Globals, Shares, Printers, Status, View и Password. Первые три из них предназначены соответственно для редактирования глобальных опций в файле `smb.conf` и определения разделяемых объ-

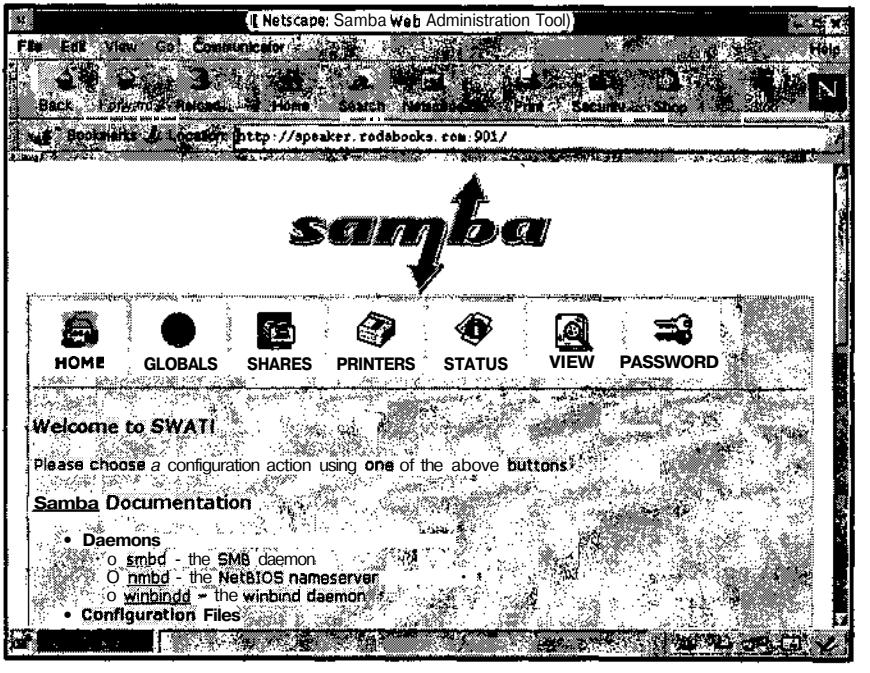


Рис. 16.8. Исходная страница SWAT позволяет выбрать категорию для настройки, а также содержит ссылки на документы, описывающие Samba

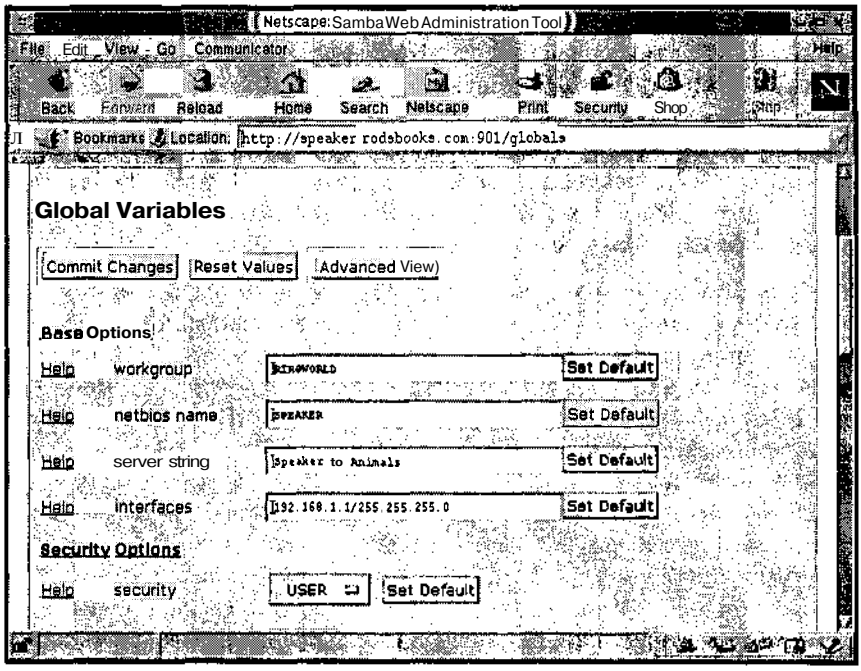


Рис. 16.9. Глобальные опции воздействуют на разделяемые объекты файлов и принтеров и определяют особенности выполнения основных операций Samba

ектов, описывающих каталоги и принтеры. Страница Status предоставляет информацию об использовании разделяемых объектов. Страница View отображает содержимое файла `smb.conf`, а страница Password позволят изменять пароль. Если вы укажете имя, отличное от `root`, страницы Globals, Shares и Printers будут не доступны, а на страницах Status и Password будет представлен ограниченный набор опций. Помимо ссылок на описанные выше страницы, исходная Web-страница содержит ссылки на документацию по Samba.

Основные действия по настройке Samba выполняются с помощью страниц Globals, Shares и Printers. Страница Globals, показанная на рис. 16.9, позволяет задавать опции в разделе `[globals]` файла `smb.conf`. С ее помощью можно указать имя NetBIOS, имя рабочей группы, кодирование пароля и другие опции.

По умолчанию на страницах Shares и Printers не отображается информация о разделяемых объектах. Чтобы создать или отредактировать объект, вам необходимо выполнить одно из следующих действий.

- Для того чтобы отредактировать существующий разделяемый объект, надо выбрать имя объекта в одном из списков, расположенных возле кнопок Choose Share и Choose Printer, а затем щелкнуть на кнопке Choose Share или Choose Printer. Информация о разделяемом объекте отобразится в окне броузера, и вы сможете изменить необходимые установки.

- Для того чтобы удалить существующий разделяемый объект, выберите его имя, как описано выше, и щелкните на кнопке Delete Share или Delete Printer.

- Чтобы создать новый разделяемый модуль, введите в поле редактирования имя, которое вы хотите присвоить ему, и щелкните на кнопке Create Share или Create Printer. Имя объекта не надо помещать в квадратные скобки, как вы делаете это, редактируя файл `smb.conf`; SWAT добавит скобки самостоятельно. После создания объекта вы можете отредактировать любые опции так же, как и при работе с существующим объектом.

Как было сказано в главе 7, разделяемый объект `[homes]` имеет специальное назначение, но вы можете создавать, удалять и редактировать его, как любой другой объект. Принтер, помеченный в списке символом *, является принтером по умолчанию, созданным посредством объекта `[printers]`. Лучше всего непосредственно изменять опции в объекте `[printers]`, но если вы хотите создать разделяемый объект для конкретного принтера, переопределяющий установки в `[printers]`, отредактируйте объект, помеченный звездочкой.

На страницах Globals, Shares и Printers присутствует кнопка Advanced View. (На страницах Globals и Shares она появляется лишь после создания или выбора разделяемого объекта.) По умолчанию администратору доступны лишь наиболее часто используемые опции Samba. После щелчка на Advanced View в окне броузера будут представлены все опции, соответствующие некоторой категории, а кнопка Advanced View будет заменена на Basic View. При активизации кнопки Basic View SWAT снова переходит к отображению ограниченного набора опций. В большинстве случаев разделяемые объекты Samba можно отредактировать с помощью опций, отображаемых по умолчанию, а кнопку Advanced View приходится использовать лишь в отдельных случаях. При переходе к ограниченному набору опций изменения, внесенные в режиме полного набора опций, не теряются.

По окончании работы со страницами Globals, Shares или Printers щелкните на кнопке Commit Changes, в результате чего внесенные вами изменения будут записаны в файл `smb.conf`. Чтобы эти изменения были учтены при работе сервера Samba, вам надо перезапустить его. Сделать это можно с помощью интерфейсных элементов, расположенных на странице Status. Щелкните на кнопке Restart `smbd`, а затем на кнопке Restart `nmbd`. (Некоторые изменения требуют перезапуска лишь одного из двух серверов, но если вы не знаете, какой сервер затрагивают выполненные вами действия, лучше перезагрузить оба.) Закончив работу с инструментом SWAT, надо завершить выполнение Web-броузера, в противном случае вы можете создать угрозу безопасности системы.

Вопросы безопасности при удаленном администрировании

Удаленное администрирование само по себе создает опасность для системы, независимо от того, как оно выполняется: путем удаленной регистрации (в текстовом режиме или с использованием графического интерфейса) либо с помощью специализированных инструментов администрирования. При удаленном администрировании возникает угроза безопасности двух типов.

- **Похищение пароля.** Если пароль администратора станет известен пользователю, не имеющему права выполнять администрирование, этот пользователь может изменить конфигурацию системы в своих целях.

- **Наличие недостатков в системе защиты.** Ошибки, которые можно использовать для незаконного доступа, периодически выявляются в различных серверах. Не исключено, что подобные ошибки присутствуют и в серверах, применяемых для удаленного администрирования. В этом случае злоумышленник сможет проникнуть в систему, даже не зная пароль.

Если помимо специализированного инструмента удаленного администрирования вы также используете сервер удаленной регистрации, опасность незаконного доступа в систему посторонних лиц возрастает. Как было сказано в главе 13, при работе с некоторыми серверами удаленной регистрации пароль передается в незакодированном виде, а другие средства, например SSH, обеспечивают шифрование не только пароля, но и всех передаваемых данных. Таким образом, если вместо регистрации на удаленном компьютере посредством SSH использовать для администрирования системы Linuxconf или SWAT, уровень защиты резко снизится. Сервер Webmin может осуществлять взаимодействие с клиентом через SSL (`http://www.openssl.org`), в результате чего обеспечивается кодирование пароля и других данных, но это возможно только в том случае, если на компьютерах установлены и настроены средства поддержки SSL. Из сказанного выше следует, что инструменты Linuxconf и SWAT (а также Webmin при отсутствии кодирования данных) желательно использовать лишь в локальных сетях, полностью контролируемых системными администраторами.

Инструменты удаленного администрирования, рассмотренные в данной главе, предоставляют пользователю доступ к удаленному компьютеру в том случае, если он знает имя и пароль. Средства удаленной регистрации, о которых шла речь в главе 13, могут быть настроены так, что регистрация под именем `root` будет запрещена. Чтобы приступить к администрированию системы, пользователю приходится сначала зарегистрироваться под своим именем, а затем получить дополнительные привилегии с помощью команды `su`. В этом случае необходимо знать два пароля: свой и пользователя `root`. Среди инструментов удаленного администрирования только Webmin обеспечивает кодирование данных, но даже он предоставляет низкий уровень защиты в случае, если злоумышленнику станет известен один из паролей. (Как вы уже знаете, существуют различные способы "подслушивания" паролей, передаваемых по сети.)

Снизить риск неавторизованного доступа к данным можно, ограничив набор компьютеров, которым разрешено обращение к инструментам удаленного администрирования. Как вы уже знаете из материала данной главы, Linuxconf позволяет указать IP-адреса компьютеров и сетей, имеющих право обращаться к нему. Если в системе используется `xinetd`, любая программа удаленного администрирования, запускаемая с помощью данного суперсервера, может быть защищена с помощью TCP Wrappers. Запретить доступ к серверу из внешней сети можно также с помощью брандмауэра. Все эти меры не исключают возможности подслушивания пароля, а IP-адрес, как показывает опыт, может быть фальсифицирован. Поэтому действия по обеспечению защиты лишь снижают вероятность незаконного доступа к системе, но не исключают полностью такой возможности.

Чем меньше серверов выполняется в системе, тем меньше вероятность того, что злоумышленнику удастся воспользоваться недостатком в защите какого-либо из серверов. Если в системе обязательно должен присутствовать сервер удаленной регистрации, желательно использовать для этой цели SSH. Удаленное администрирование также лучше выполнять посредством SSH-соединения; в этом случае риск для системы будет минимальным. Применение специализированных инструментов удаленного администрирова-

ния оправдано в тех случаях, когда сеть надежно защищена от доступа извне, а на компьютере, подлежащем администрированию, отсутствует сервер удаленной регистрации.

На первый взгляд может показаться, что применение сервера удаленного администрирования с ограниченной сферой действия, например SWAT, создает меньшую угрозу безопасности системы, чем использование универсальных серверов, таких как Linuxconf и Webmin. На самом деле это не так. Если злоумышленник проникнет в систему посредством SWAT, он может создать новый разделяемый объект, позволяющий ему читать и записывать данные в файлы, содержащиеся в каталоге /etc. С помощью этого объекта можно изменить конфигурацию системы, обеспечив для себя доступ посредством одного из серверов удаленной регистрации, например Telnet. Таким образом, применение инструментов с ограниченной областью действия может в лучшем случае замедлить процесс незаконного проникновения в систему.

Здесь приведены лишь самые общие рассуждения, имеющие отношение к обеспечению безопасности системы. Вопросам защиты полностью посвящена часть IV.

Резюме

Средства удаленного администрирования не являются неотъемлемой частью Linux, даже в том случае, если вам необходимо выполнять удаленное администрирование системы. Они очень удобны в работе, в особенности тогда, когда администратор лишь в общих чертах представляет себе функционирование сервера, который ему предстоит настраивать. Использовать инструменты удаленного администрирования можно как с локального, так и с удаленного компьютера. Примерами универсальных инструментов администрирования являются Linuxconf и Webmin. Теоретически каждый из них может поддерживать все подсистемы Linux, однако на практике этого добиться не удается, так как эти инструменты никогда не поставляются с полным набором конфигурационных модулей. Примером инструмента, который предназначен для администрирования одного сервера, является SWAT. При использовании программ удаленного администрирования, рассмотренных в данной главе, необходимо уделять большое внимание защите системы.

Глава 17

Резервное копирование

Создание резервных копий вряд ли можно считать увлекательным занятием, но если вы хотите поддерживать работоспособность вашей системы в течение длительного времени, выполнять эту рутинную работу необходимо. Некоторые сети насчитывают один или два сервера и несколько клиентов, причем конфигурация клиентских компьютеров чрезвычайно проста, и они не содержат важной информации. Клиентские машины по сути выполняют функции X-терминалов. Создавая резервные копии данных в такой сети, достаточно скопировать содержимое сервера. Что же касается клиентов, для них можно создать одну копию, отражающую стандартную конфигурацию системы. Если к сети подключено несколько серверов, необходимо организовать копирование данных с каждого из них. Если же на каждом из клиентских компьютеров будет содержаться важная информация, процедура резервного копирования будет выглядеть по-другому. В начале данной главы обсуждаются общие вопросы, связанные с созданием резервных копий, после чего будут рассмотрены три способа решения задачи резервного копирования: использование tar для архивирования информации, содержащейся на компьютерах под управлением Linux или UNIX, применение Samba для копирования данных в системе Windows и координация действий по резервному копированию в сети с помощью инструмента AMANDA.

Резервное копирование данных в сети — достаточно сложная задача, которая усложняется еще больше с увеличением размеров сети. Дополнительную информацию по этой теме вы найдете в документации, поставляемой в комплекте с инструментами резервного копирования. Тем, кому необходимо подробно разобраться в данном вопросе, можно порекомендовать книгу Престона (Preston) Unix Backup & Recovery (O'Reilly, 1999).

Использование серверов резервного копирования

Резервные копии проще всего создавать на локальной машине. Если вы установите накопитель на магнитных лентах на компьютер, работающий под управлением Linux, вы можете пользоваться такими утилитами, как tar, cpio или dump, причем для этого вам не потребуется специально настраивать вашу систему. Резервное копирование в сетевом окружении представляет собой гораздо более сложную задачу, так как компью-

теры, участвующие в сетевом взаимодействии, должны быть соответствующим образом сконфигурированы, а программы, применяемые для создания резервных копий, должны обеспечивать работу в сети. (Отсутствие сетевой поддержки в программах резервного копирования в некоторых случаях можно компенсировать за счет применения различных утилит.) Организовать восстановление данных в сети — еще более сложная задача, так как при этом приходится осуществлять взаимодействие с компьютером, на котором отсутствуют многие из сетевых инструментов. По этой причине в небольших сетях рекомендуется осуществлять резервное копирование с использованием локальных устройств. Для обеспечения подобного копирования необходимо затратить значительные средства, поскольку с ростом сети увеличивается число устройств, которые должны быть установлены на компьютерах. Стоимость накопителя, который может быть установлен на рабочей станции, составляет от 100 до 1000 долларов. К этой сумме надо добавить стоимость носителей. Устройства, позволяющие осуществлять резервное копирование в сети, стоят значительно больше, но расходы в пересчете на один компьютер оказываются намного меньше, чем при использовании локальных устройств. Таким образом, одним из аргументов в пользу сетевого резервного копирования является экономия средств.

Аппаратные средства, предназначенные для создания резервных копий

Чаще всего для хранения резервных копий данных используются магнитные ленты. Как правило, чем меньше цена накопителя, тем больше стоят ленты для него, таким образом, по мере приобретения магнитных носителей разница в цене между низкоуровневым и высокоуровневым устройством будет становиться все меньше и меньше.

В небольших сетях для резервного копирования чаще всего применяются накопители среднего и высокого уровня стоимостью около 10QO долларов. Такие устройства обычно обеспечивают запись на носитель от 5 до 20 Гбайт информации. Как правило, подобные устройства используют формат DAT (Digital Audio Tape — цифровая аудиолента) или DLT (Digital Linear Tape — лента с цифровой линейной записью). Для резервного копирования в больших сетях применяются более дорогие устройства, использующие DLT либо другие "экзотические" форматы. Объем копируемой информации может быть настолько велик, что станет оправданным применение накопителей с автоматической сменой лент. Логически устройства с автоматической сменой лент могут рассматриваться как накопители с носителем сверхбольшого объема.

Магнитная лента — не единственный носитель, пригодный для создания резервных копий. В качестве альтернативы могут рассматриваться оптические носители, такие как компакт-диски (в том числе перезаписываемые) и DVD. Обычные компакт-диски имеют небольшой объем (630 Мбайт) и не всегда подходят для резервного копирования, однако на них можно хранить информацию, не изменяющуюся во времени, например, использовать их для инсталляции операционной системы на клиентском компьютере.

Объем перезаписываемых DVD исчисляется гигабайтами, и на них помещается операционная система в полном объеме, но для сетевого копирования он может оказаться недостаточным. Срок хранения информации на оптических носителях очень велик (для компакт-дисков он составляет от 10 до 100 лет), поэтому они хорошо подходят для создания архивов. В системе Linux подобные устройства гораздо менее удобны, чем накопители на лентах, так как для записи данных на них нужны специальные программы, например `cdrecord`. Однако процесс восстановления информации с таких носителей очень прост, поскольку они читаются стандартными устройствами, входящими в состав практически каждого компьютера.

В дальнейшем при изложении материала данной главы предполагается, что при создании резервных копий вы используете в качестве носителя магнитную ленту. Возможно, что вы захотите дополнить данные, хранящиеся на лентах, копиями на оптических носителях. В этом случае восстановление данных несколько упрощается. Вы можете скопировать исходную конфигурацию с компакт-диска, а затем прочитать требуемые данные с ленты.

Создание резервных копий в сети несколько упрощает работу системного администратора по сравнению с локальным резервным копированием. Централизованно контролируя процесс создания резервных копий, администратор избавляет пользователей от необходимости решать данную задачу самостоятельно. Пользователь может забыть скопировать данные, администраторы же в большинстве случаев автоматизируют операцию копирования так, что она выполняется незаметно для пользователей.

Способы резервного копирования

Существуют два основных способа сетевого резервного копирования. Первый способ состоит в том, что операция копирования инициируется компьютером, данные которого должны быть сохранены; при этом используется накопитель на лентах, подключенный к другому компьютеру. Такой способ называется *резервным копированием, инициируемым клиентом*. Второй способ предполагает копирование по инициативе компьютера, к которому подключен накопитель; этот компьютер обращается к узлу сети, содержащему данные, предназначенные для сохранения. В этом случае говорят о *резервном копировании, инициируемом сервером*. Оба способа будут рассмотрены в данной главе. Каждый из них имеет свои преимущества и недостатки.

Компьютер, к которому подключено устройство резервного копирования, называется *сервером резервного копирования,* а компьютер, который содержит данные, предназначенные для сохранения, называется *клиентом резервного копирования*. Несмотря на это определение, очевидно, что в случае копирования, инициируемого сервером, клиент резервного копирования должен содержать программное обеспечение, выполняющее функции сервера. По отношению к данному серверу сервер резервного копирования выполняет функции клиента. Использование терминов *сервер резервного копирования* и *клиент резервного копирования* упрощает обсуждение способов создания резервных копий.

Резервное копирование, инициируемое клиентом

При выполнении копирования, инициируемого клиентом, на компьютере, выполняющем функции сервера резервного копирования, должна находиться программа-сервер, которая обеспечивает доступ клиента к устройству, предназначенному для создания резервных копий. Эта программа может не только участвовать в резервном копировании, но и выполнять другие функции. Так, например, в качестве программ-серверов могут быть использованы `rshd` и Samba. Копирование, инициируемое клиентом, имеет следующие особенности.

- Планирование. Если копирование данных на резервный носитель должно осуществляться периодически, инициировать процесс резервного копирования можно

с помощью cron или другого инструмента, осуществляющего вызов программ по расписанию. При этом необходимо предусмотреть средства идентификации, которые позволяли бы различать между собой копии, созданные разными клиентами.

- **Контроль пользователя за процессом копирования.** Создавая резервные копии содержимого рабочих станций, удобнее осуществлять копирование, инициируемое клиентом, так как в этом случае пользователь может выбирать время начала копирования и даже файлы, предназначенные для сохранения на резервном носителе. Возможно, пользователь захочет выполнить внеочередное копирование после окончания определенного этапа работы над проектом или перед началом отпуска.

- **Безопасность системы.** На компьютере, выполняющем роль клиента резервного копирования, не должна присутствовать программа-сервер; в результате уровень безопасности системы повышается. Если данные копируются с машины, работающей под управлением Linux или UNIX, резервное копирование может осуществляться от имени пользователя root, что обеспечивает полный доступ ко всем файлам.

Учитывая эти особенности, становится очевидно, что резервное копирование, инициируемое клиентом, удобно использовать в небольших сетях. Конфликты, возникающие при планировании операций копирования, легко разрешимы. Недостатком такого способа является тот факт, что копированием по сути управляют пользователи, которые могут забыть вовремя создать резервную копию своих данных. Используя cron или другой подобный инструмент, можно автоматизировать процесс резервного копирования, но при этом надо уделять большое внимание составлению расписания, чтобы исключить возможность конфликта.

Резервное копирование, инициируемое сервером

При выполнении копирования, инициируемого сервером, на компьютере, к которому подключен накопитель (сервере резервного копирования), выполняется клиентская программа, поддерживающая сетевое соединение с клиентом резервного копирования. Сервер резервного копирования получает данные с клиента резервного копирования и сохраняет их на ленте или на другом носителе. Такой подход имеет ряд особенностей.

- **Планирование.** Поскольку управление резервным копированием осуществляется централизованно, вы можете без труда предотвратить конфликты и даже составить расписание так, чтобы копирование осуществлялось в те часы, когда нагрузка на сеть минимальна (например, ночью).

- **Контроль пользователя за процессом копирования.** Пользователь, работающий на рабочей станции, не контролирует процесс резервного копирования. Если необходимо обеспечить создание резервных копий по требованию пользователя, можно организовать дополнительное копирование важных данных, осуществляя запись на альтернативные носители, например на диск Zip.

- **Безопасность системы.** На каждом клиенте резервного копирования должна выполняться некоторая серверная программа. Обычно эта программа поддерживает протокол разделения файлов, например NFS или SMB/CIFS, либо протокол, подобный FTP, и обеспечивает полный доступ к данным на компьютере. Если злоумышленник использует метод фальсификации IP-адресов и обратится к компьютеру от

имени сервера резервного копирования, он может похитить важные данные, например файл /etc/shadow. Сервер резервного копирования в этом случае менее подвержен атакам, так как на нем выполняется только клиентская программа.

Поскольку при резервном копировании, осуществляемом по инициативе сервера, легко составить график копирования и обеспечить его выполнение, данный способ пригоден для работы в больших сетях. Чтобы ограничить круг узлов сети, которые имеют возможность обращаться к файловой системе клиента резервного копирования, можно использовать брандмауэр или другой механизм управления доступом. В специализированных пакетах, например AMANDA, для основных исполняемых программ устанавливается признак SUID. В результате право запускать эти программы получают обычные пользователи.

ВНИМАНИЕ

Независимо от того, выполняется ли резервное копирование по инициативе клиента или по инициативе сервера, при создании резервных копий в сети необходимо уделять большое внимание вопросам безопасности. Если используемый протокол предполагает ввод пароля и этот пароль пересылается в незашифрованном виде, он может быть перехвачен. То же справедливо для данных, передаваемых в процессе копирования. Однако даже кодирование информации при передаче не обеспечивает полной безопасности. Необходимо также следить за сохранностью носителей. Если лента с данными хотя бы на короткое время попадет в руки злоумышленника, он может скопировать с нее важную информацию, несмотря на то, что на компьютере доступ к ней запрещен для всех, кроме пользователя root.

Использование tar

Утилита tar — одна из самых популярных программ, используемых для резервного копирования в системах Linux и UNIX. Она объединяет несколько файлов в один файл архива, что упрощает передачу информации по сети и сохранение ее на резервном носителе. Название программы представляет собой сокращение слов tape archive (архив на ленте). Утилиту tar можно использовать для организации резервного копирования как по инициативе клиента, так и по инициативе сервера. Вместо tar в системе Linux могут применяться и другие подобные программы, например cpio или dump. Особенности работы с ними описаны в документации на программы и в справочной системе Linux. В данной главе обсуждается лишь программа tar; ей уделено особое внимание потому, что она наиболее популярна среди пользователей, а также потому, что она используется другими инструментальными средствами, например smbtar и AMANDA.

Возможности tar

Утилита tar — чрезвычайно мощный инструмент; она поддерживает большое количество опций. Опции программы tar делятся на две категории: команды и модификаторы. Команды указывают утилите tar, какие действия она должна выполнить, например, создать архив, вывести содержимое существующего архива, извлечь файлы и т. д. Модификаторы уточняют действия программы. С их помощью можно определить устройство, на которое следует записать архив, указать файлы, которые необходимо включить в архив,

или задать сжатие архива посредством gzip или bzip2 и т. д. Утилита tar вызывается следующим образом:

tar *команда [модификаторы] имена_файлов*

В качестве имен файлов в большинстве случаев задаются имена каталогов. Если при вызове программы задано имя каталога, tar включает в состав архива все файлы и все подкаталоги этого каталога. Чтобы создать архив всей файловой системы, надо указать корневой каталог (/).

В табл. 17.1 и 17.2 перечислены наиболее часто используемые команды и модификаторы утилиты tar. На самом деле набор допустимых опций гораздо более обширный и включает большое количество команд и модификаторов. Дополнительную информацию о них можно получить на страницах справочной системы, посвященных утилите tar.

В качестве примера использования приведенных выше опций рассмотрим следующую ситуацию. Предположим, что к компьютеру через интерфейс SCSI подключен накопитель на магнитных лентах. Для доступа к этому устройству используется имя /dev/stO или /dev/nstO. Для создания резервной копии содержимого каталога /home с сохранением прав доступа и с выводом имен архивируемых файлов надо задать следующую команду:

```
# tar --create --verbose --file /dev/stO /home
```

Если указать сокращенные обозначения опций, приведенные в табл. 17.1 и 17.2, то данная команда примет вид

```
# tar cvf /dev/stO /home
```

Некоторые опции программы tar (а именно --one-file-system, --same-permissions, --listed-incremental и --verify) заслуживают более подробного обсуждения. В состав файловой системы Linux могут входить виртуальные файловые системы (например, /proc) и сменные носители. Кроме того, не исключено, что вы захотите запретить резервное копирование файловых систем, находящихся на некоторых устройствах. При использовании опции --one-file-systemкопироваться будут только те разделы, которые вы непосредственно укажете. Вместо --one-file-system можно задать опцию --exclude или --exclude-from, которая позволяет непосред-

Таблица 17.1. Часто употребляемые команды утилиты tar

Команда	Сокращенный вариант	Описание
--create	c	Создает архив
--concatenate	A	Добавляет tar-файл к существующему архиву
--append	r	Добавляет обычные файлы к существующему архиву
--update	u	Добавляет файлы, которые имеют более позднюю дату создания, чем файлы с соответствующими именами, присутствующие в составе архива
--diff или --compare	d	Сравнивает файлы в архиве с файлами на диске
--list	t	Выводит содержимое архива
--extract или --get	x	Извлекает файлы из архива

Таблица 17.2. Часто употребляемые модификаторы утилиты tar

Модификатор	Сокращенный вариант	Описание
`--absolute-paths`	`P`	Сохраняет символ / в начале пути к файлу
`--bzip2`	`I`	Задает обработку архива с помощью `bzip2`. (В старых версиях tar не поддерживается)
`--directory` *каталог*	`C`	Перед обработкой данных делает указанный каталог текущим
`--exclude` *файл*	(отсутствует)	Запрещает включать файл в архив
`--exclude-from` *файл*	X	Запрещает включать в архив файлы, указанные в данном файле
`--file [`*узел*`:]`*файл*	`f`	Выполняет архивирование, используя в качестве архива указанный файл на указанном узле. (Узел сети указывается при выполнении резервного копирования, инициируемого клиентом.)
`--gzip` или `--ungzip`	z	Задает обработку архива программой `gzip` или `ungzip`
`--listed-incremental=`*файл*	g	Создает или использует файл, содержащий результаты инкрементного копирования
`--multi-volume`	M	Задает обработку архива на нескольких лентах
`--one-file-system`	1	Сохраняет или восстанавливает только одну файловую систему
`--same-permissions` или `--preserve-permissions`	p	Сохраняет информацию о пользователях и о правах доступа
`--tape-length` *N*	`L`	Определяет длину ленты в килобайтах; используется совместно с `--multi-volume`
`--verbose`	v	Выводит информацию об обработанных файлах
`--verify`	`W`	Сразу после записи сравнивает исходный файл с файлом, записанным в архив

ственно исключать из процесса резервного копирования некоторые каталоги, например `/proc`.

Опция `--same-permissions` важна при работе с системными файлами, поскольку в ряде случаев утилита tar теряет некоторые данные о правах доступа. Чаще всего это проявляется, когда конкретные права не соответствуют значению `umask`. Опция `--same-permissions` бывает необходима при восстановлении сохраненных файлов.

Если при вызове tar указана опция `--listed-incremental`, программа создает новый файл либо использует имеющийся с информацией о файлах, включенных в ар-

хив. При пером запуске tar с этой опцией создается файл для хранения сведений об архиве, и все указанные файлы помещаются в архив. При последующих вызовах утилиты tar с опцией `--listed-incremental` обрабатываются только те файлы, которые были созданы или модифицированы с момента последней операции резервного копирования. Другими словами, данная опция позволяет вместо полного резервного копирования осуществлять *частичное,* или *инкрементное,* копирование. Многие администраторы раз в неделю или раз в месяц выполняют полное копирование и ежедневно — частичное. Такой подход позволяет обеспечить сохранность данных минимальными усилиями. (Восстанавливая информацию с резервной копии, созданной посредством инкрементного копирования, вы, возможно, обнаружите, что недавно удаленные файлы снова появились на диске. Дело в том, что при инкрементном копировании файлы, которые были удалены с момента последнего полного копирования, не помечаются как отсутствующие.) Выполняя резервное копирование в сетевом окружении, целесообразно в разные дни недели осуществлять полное копирование информации на разных компьютерах. Например, вы можете составить расписание и указать в нем, что в понедельник должно выполняться копирование содержимого machinel, во вторник — machine2 и т. д.

Опция `--verify`предназначена для проверки того, насколько корректно выполнено копирование данных. В результате такой проверки время копирования существенно возрастает, но подобное замедление работы часто бывает оправдано, особенно в тех случаях, когда в накопителе отсутствует встроенная функция проверки. (В большинстве устройств среднего и высокого уровня поверка осуществляется на аппаратном уровне.) Если вы указываете при вызове tar опцию `--verify`или `--diff`, возможны ложные сообщения об ошибках. Дело в том, что в то время, когда выполняется резервное копирование, пользователи продолжают работать в системе, поэтому с момента копирования на резервный носитель до момента проверки файл может быть изменен. Наиболее часто модифицируются файлы протоколов, содержимое очереди на печать, файлы в каталоге `/tmp` и другие подобные данные. Если в результате проверки выявлено несоответствие содержимого часто изменяемых файлов, повода для беспокойства нет. Если же при проверке не совпали статические данные, например файлы в каталоге `/usr`, это может означать, что копирование выполняется некорректно.

Многие современные накопители на магнитных лентах поддерживают встроенные функции сжатия, поэтому в применении опции `--bzip2` или `--gzip` обычно нет необходимости. Если же вы указываете данные опции при вызове tar, помните, что их использование может представлять угрозу для целостности резервной копии. Опции `--bzip2` и `--gzip` осуществляют сжатие не отдельных файлов, а всего архива, поэтому если в процессе сжатия возникнет ошибка, данные, содержащиеся в архиве, будут утеряны. Сжатие, реализованное на аппаратном уровне, обеспечивает определенную устойчивость к ошибкам подобного рода. В случае сбоя искажаются один-два файла, а остальные данные в архиве можно использовать. Некоторые программы резервного копирования сжимают данные таким же способом, повышая тем самым надежность всей системы. Например, в коммерческом продукте BRU (`http://www.tolisgroup.`com) используется пофайловое сжатие информации.

Тестирование средств резервного копирования на локальном компьютере

Устанавливая сервер резервного копирования, целесообразно проверить, насколько хорошо он справляется с задачей копирования локальной информации, и лишь затем использовать его для работы в сети. Копировать данные, расположенные на локальном компьютере, намного проще, чем выполнять те же действия при взаимодействии по сети, поэтому, для того, чтобы разрешить проблемы, возникающие при резервном копировании в сети, желательно убедиться, что на локальной машине те же самые операции выполняются нормально. Нельзя также забывать о том, что содержимое сервера резервного копирования, как и любого другого компьютера, также необходимо записывать на резервный носитель. Если вы не сделаете этого, то при выходе сервера из строя вам не удастся быстро восстановить его работоспособность, в результате чего вы не сможете выполнять резервное копирование данных, содержащихся на других компьютерах.

В качестве простейшей проверки можно скопировать данные с помощью одной из команд, рассмотренных в предыдущем разделе. Чтобы сделать это, необходимо выяснить, к какому устройству следует обращаться. Для накопителей среднего и высокого уровня чаще всего используются следующие четыре имени: `/dev/st0`, /dev/nstO, `/dev/ht0` и `/dev/nht0`. Первые два имени применяются для обозначения устройств SCSI, а остальные два соответствуют устройствам EIDE/ATAPI. Имена файлов, имена которых начинаются с буквы п, представляют *устройства без перемотки* (nonrewinding device). По окончании действий с таким устройством лента остается в той позиции, в которой были записаны последние данные, в результате вы можете размещать несколько копий на одной ленте. Если в начале имени файла буква п отсутствует, это устройство считается *устройством с перемоткой* (rewinding device). В этом случае по окончании операции записи лента автоматически перематывается. Заметьте, что наличие или отсутствие перемотки является характеристикой не устройства, а представляющего его файла. Каждый накопитель, в зависимости от используемого драйвера, может рассматриваться либо как устройство без перемотки, либо как устройство с перемоткой. Если к компьютеру подключено несколько накопителей на ленте, имя файла, представляющего второе устройство, будет заканчиваться 1, следующему устройству будет соответствовать 2 и т. д.

Существуют накопители, которым соответствуют файлы устройств с другими именами. Например, раньше использовались накопители, подключаемые через порт, предназначенный для работы с гибкими дисками. Работа с ними осуществлялась посредством файлов `/dev/qft0` и `/dev/nqft0`. Эти накопители отличаются малой емкостью носителей и низким быстродействием и не отвечают современным требованиям, предъявляемым к устройствам резервного копирования. Иногда устройства подключаются через специализированный интерфейс. Драйверы для работы с ними должны быть установлены при настройке Linux.

Если при выполнении копирования данных с локального компьютера у вас возникают проблемы, проверьте используемые аппаратные средства и драйверы устройств. Для работы с устройством SCSI необходимы как базовые средства поддержки SCSI, так и средства для взаимодействия с накопителем SCSI. Соответственно при использовании устройств EIDE/ATAPI необходимо обеспечить поддержку как EIDE, так и накопителя EIDE/ATAPI. При тестировании следует проверить не только создание резервных копий, но и восстановление данных. Целесообразно сначала испробовать сервер резервного копирования на небольшом каталоге, а потом переходить к работе с данными значительного объема.

Чтобы убедиться, что информация восстановлена корректно, надо использовать режим верификации.

Если необходимо хранить несколько архивов на одной ленте, вам может пригодиться в работе утилита mt. Эта программа позволяет управлять накопителем на магнитных лентах, задавать различные режимы работы, например включать встроенные средства сжатия, и выполнять другие функции.

В справочных руководствах по mt и tar резервные копии называются файлами. Магнитную ленту можно сравнить с жестким диском без файловой системы. Архивы, созданные с помощью программы tar, располагаются на ленте последовательно один за другим.

Используя tar и mt, можно разместить на одной ленте несколько резервных копий. Утилита mt вызывается следующим образом:

mt [-f *устройство] операция [счетчик]* [*параметры*]

Под операцией подразумевается одна из следующих команд: `fsf` (forward space files — переход к следующему файлу), bsf (backward space files — переход к предыдущему файлу), rewind (перемотка ленты) и `datcompression` (установка режима сжатия; параметр 0 запрещает, а параметр 1 разрешает сжатие). Например, в результате приведенной ниже последовательности команд создаются две резервные копии и осуществляется их верификация.

```
# tar cvplf /dev/nstO testdir-1/
# tar cvplf /dev/nstO testdir-2/
# mt -f /dev/nstO rewind
# tar df /dev/nstO testdir-1/
# mt -f /dev/nstO fsf 1
# tar df /dev/nstO testdir-2/
```

Большинство из этих команд предполагает выполнение некоторых действий с лентой. Первые два вызова утилиты tar сопровождаются отображением имен копируемых файлов, а в результате последующих двух вызовов выводятся имена файлов, в которых обнаружено расхождение между исходными данными и данными, записанными на ленту. Второй вызов утилиты mt нужен при чтении архива. При создании резервных копий в этой команде нет необходимости.

Резервное копирование, инициируемое клиентом

Для того чтобы резервное копирование с использованием tar, осуществляемое по инициативе клиента, стало возможным, на клиентской машине должна присутствовать программа tar, а на сервере резервного копирования должна выполняться программа-сервер, предоставляющая утилите tar на клиентской машине доступ к накопителю. Действия на стороне клиента немногим отличаются от рассмотренных ранее, необходимо лишь изменить порядок вызова утилиты tar. Что же касается сервера резервного копирования, то при работе с большинством дистрибутивных пакетов вам необходимо изменить его конфигурацию.

Настройка сервера для резервного копирования по инициативе клиента

Опция `--file`, описанная в табл. 17.2, позволяет указать программе tar файл архива. Это может быть обычный файл на диске, файл устройства, представляющий накопитель на магнитных лентах, и путь к сетевому ресурсу. В последнем Случае на компьютере, выступающем в роли сервера резервного копирования, должен выполняться демон rshd (часто он называется `in.rshd`). Этот демон позволяет удаленной системе выполнять команды на локальной машине. Благодаря наличию программы rshd утилита tar получает возможность передать созданный ею файл на устройство, подключенное к серверу резервного копирования. Сервер rshd поставляется с большинством версией системы Linux и обычно запускается посредством суперсервера. Соответствующая запись в файле `/etc/inetd.conf` имеет следующий вид:

```
shell stream tcp nowait root /usr/sbin/tcpd \
/usr/sbin/in.rshd -h
```

Если в вашей системе используется сервер xinetd, вам необходимо создать запись аналогичного назначения в файле `/etc/xinetd.conf` или создать отдельный файл и включить его в каталог `/etc/xinetd.d`. При выполнении резервного копирования чрезвычайно важны меры по ограничению доступа, предпринимаемые посредством TCP Wrappers или непосредственно предусмотренные в xinetd. Решение о предоставлении доступа через rshd принимается на основе анализа IP-адреса. Хотя TCP Wrappers и xinetd используют тот же механизм контроля за обращениями клиентов, избыточные средства обеспечения безопасности пригодятся на тот случай, если в программе rshd будут обнаружены ошибки, позволяющие обойти механизмы защиты.

Несмотря на то что основные меры защиты rshd базируются на использовании информации об IP-адресе, данная программа также проверяет имена пользователей. Это делается для того, чтобы предотвратить попытки запуска программ, которые могут нанести вред системе. В обычных условиях rshd не обрабатывает команды, полученные от пользователя root, независимо от того, на каком компьютере он работает. Это правило можно отменить путем указания опции -h, как это было сделано в рассмотренном выше примере записи в файле `inetd.conf`. Данная опция чрезвычайно важна, так как для выполнения резервного копирования приходится иметь дело с системными файлами, а для работы с ними необходимы максимальные привилегии. Если вы не укажете опцию -h, то обычные пользователи смогут выполнять резервное копирование только в том случае, если права доступа к файлу устройства на сервере позволят им сделать это. (В большинстве дистрибутивных пакетов обычным пользователям запрещен доступ к накопителям на магнитных лентах.)

ВНИМАНИЕ В некоторых системах опция -h программы rshd не обрабатывается. В этом случае приходится создавать резервные копии по-другому. Необходимо запустить на сервере резервного копирования сервер SSH и на стороне клиента связать ssh с именем rsh. При этом утилита tar для передачи данных по сети будет обращаться к программе ssh. Такой подход обеспечивает дополнительную степень защиты, и ему имеет смысл следовать даже в тех случаях, когда опция -h обрабатывается так, как указано в документации. Сервер SSH необходимо сконфигурировать таким образом, чтобы он при выполнении аутентификации не запрашивал пароль.

Учитывая вопросы защиты данных, желательно выделить для сервера резервного копирования отдельную машину. Для этой цели подойдет компьютер небольшой мощности. Единственное требование, предъявляемое к нему, — это наличие накопителя на магнитных лентах и высокопроизводительного сетевого соединения. Используя брандмауэр, можно запретить доступ к серверу резервного копирования из Internet, но, несмотря на это, нежелательно размещать на нем важные данные.

Выполнение резервного копирования

Установив сервер резервного копирования, его можно использовать для создания копий данных. Для этого вам надо установить магнитную ленту и задать на компьютере, выполняющем функции клиента резервного копирования, команду наподобие следующей:

```
# tar cvlpf buserver:/dev/st0 /home /var /
```

В результате выполнения данной команды содержимое каталогов /home, /var и / локального компьютера будет передано на устройство записи на магнитную ленту, расположенное на компьютере `buserver`. Если в подкаталогах указанных каталогов смонтированы другие файловые системы, они исключаются из процесса копирования. Если содержимое компьютера исчерпывается тремя указанными выше каталогами, в результате выполнения данной команды осуществляется полное резервное копирование.

Для управления накопителем, подключенным к серверу резервного копирования, может использоваться утилита mt. Например, по команде mt `-fbuserver:`/dev/nstO rewind осуществляется перемотка ленты.

Процедура резервного копирования, инициированная клиентом, во многом напоминает создание резервной копии на локальном компьютере. Отличие лишь в том, что, задавая устройство, следует указать сервер резервного копирования. Однако, для того, чтобы подобные действия стали возможны, необходимо специальным образом настроить сервер резервного копирования.

Резервное копирование, инициируемое сервером

Как было сказано ранее, резервное копирование, инициируемое сервером, дает возможность составить график создания резервных копий и соблюдать его. При этом основные действия по настройке производятся на компьютере, выступающем в роли клиента резервного копирования; в частности, на этом компьютере необходимо организовать работу сервера, обеспечивающего взаимодействие по сети. В данном разделе описывается создание резервных копий с использованием сервера NFS. После настройки клиента процедура резервного копирования немногим отличается от создания резервных копий на локальном компьютере, необходимо лишь смонтировать каталоги, экспортируемые клиентом резервного копирования.

Для выполнения копирования можно использовать протокол разделения файлов, отличный от NFS. Далее в этой главе будет рассматриваться копирование данных с компьютера под управлением Windows с использованием `smbmount`. Для работы с данными в системе Linux необходим протокол, который обеспечивал бы сохранение информации о владельцах файлов и правах доступа к ним. Таким протоколом является NFS.

Установка конфигурации сети для резервного копирования, инициируемого сервером

Вопросы настройки компьютера под управлением Linux для экспортирования файловых систем рассматривались в главе 8. Чтобы скопировать на резервный носитель всю информацию, содержащуюся на компьютере, надо сконфигурировать систему так, чтобы на сервере резервного копирования можно было смонтировать все файловые системы. При необходимости можно исключить из процесса резервного копирования `/proc`, сменные носители и некоторые разделы файловой системы. Как правило, клиент резервного копирования настраивается так, чтобы сервер имел доступ ко всем разделам жестких дисков.

Для создания резервной копии достаточно смонтировать каталоги лишь для чтения; сервер резервного копирования не должен записывать в них новую информацию. Если же вам надо восстановить сохраненные данные, конфигурацию необходимо несколько изменить, предоставив серверу право записи в соответствующие каталоги. При необходимости можно использовать более сложные способы восстановления данных. Например, вы можете сначала сформировать требуемые каталоги на сервере резервного копирования и записать в них информацию, а затем скопировать все дерево каталогов на клиентскую машину. Даже в том случае, когда резервная копия создавалась по инициативе сервера, можно организовать восстановление информации по инициативе клиента.

При выполнении резервного копирования по инициативе сервера возникает угроза безопасности системы. Она состоит в том, что пользователь, зарегистрированный на сервере резервного копирования под именем `root`, получает неограниченный доступ к данным на клиентской машине. Происходит это потому, что при экспортировании каталогов указывается опция `no_root_squash`. Без этой опции сервер резервного копирования не сможет прочитать системные файлы и файлы, принадлежащие различным пользователям. Такая конфигурация позволяет злоумышленнику, имеющему доступ к компьютерам локальном сети, прочитать все файлы с клиента резервного копирования, а если при экспортировании каталогов была разрешена запись, то он сможет даже изменить содержащиеся в них данные. Подобные действия имеют возможность выполнять и пользователи внешней сети, указав в запросе фальсифицированный IP-адрес. Для того чтобы уменьшить вероятность несанкционированного доступа, необходимо защитить сервер и все клиенты резервного копирования с помощью брандмауэра, а систематически просматривая файлы протоколов, вы сможете вовремя выявлять попытки взлома системы.

В качестве примера выбора конфигурации системы рассмотрим клиент, который содержит три каталога, предназначенных для копирования на резервный носитель: /home, /var и / (корневой каталог). Для экспортирования соответствующих файловых систем необходимо создать записи в файле `/etc/exports`. Если сервер резервного копирования имеет имя `buserver`, эти записи будут иметь следующий вид:

```
/home   buserver(ro,no_root_squash)
/var    buserver(ro,no_root_squash)
/       buserver(ro,no_root_squash)
```

Для восстановления файлов следует вместо `ro` указать rw и перезагрузить сервер NFS. При восстановлении данных необходимо также обеспечить сохранность информации о принадлежности файлов владельцам. Если некоторый файл принадлежит, например, пользователю `j`brown, но соответствие имени и числового идентификатора установить не удается, то данные о владельце восстановленного файла могут быть искажены. Чтобы

исключить подобный эффект, надо обеспечить, чтобы на сервере и на клиенте резервного копирования содержались учетные записи одних и тех же пользователей и чтобы совпадали их числовые идентификаторы.

Выполнение резервного копирования

Команды, по которым выполняется резервное копирование, напоминают команды, рассмотренные ранее, но, для того, чтобы они выполнялись успешно, необходимо смонтировать каталоги, экспортируемые клиентом, на сервере резервного копирования. Предположим, что клиент резервного копирования имеет имя budient, а на сервере существует каталог /mnt/client, выполняющий роль точки монтирования. Для монтирования каталогов и создания резервной копии может быть использована приведенная ниже последовательность команд.

```
# mount -t nfs -o soft buclient:/ /mnt/client
# mount -t nfs -o soft buclient:/var /mnt/client/var
# mount -t nfs -o soft buclient:/home /mnt/client/home
# cd /mnt/client
# tar cvlf /dev/stO home var ./
```

При составлении приведенного выше набора команд предполагалось, что сервер NFS на компьютере, выполняющем роль клиента резервного копирования, не обеспечивает автоматическое монтирование экспортируемых подкаталогов. Если автоматическое монтирование подкаталогов поддерживается, достаточно первой из перечисленных команд mount.

Обратите внимание на то, что среди представленных выше команд присутствует команда cd. С ее помощью точка монтирования экспортируемых каталогов объявляется как текущий каталог. При этом дерево подкаталогов выглядит так, как будто вы зарегистрировались на клиенте резервного копирования. Программа tar сохраняет информацию о каталогах, лишь начиная с точки монтирования; полный путь к файлу не записывается. В результате в архиве на ленте никак не отображена информация о том, что экспортируемые каталоги были смонтированы в точке /mnt/client и при восстановлении данных требуемые каталоги можно смонтировать в любой другой позиции файловой системы. Для создания резервной копии можно также использовать следующую команду:

```
# tar cvlf /dev/stO /mnt/client/home /mnt/client/var /mnt/client
```

В данной команде явно указан каталог /mnt/client. (Если вы не зададите модификатор --absolute-paths, при создании архива первый символ / будет удален и сохранится лишь имя mnt/client.) В этом случае при восстановлении файлов каталоги необходимо монтировать либо в той же точке, в которой они монтировались при создании резервной копии, либо в каталоге, путь к которому оканчивается на mnt/client. В противном случае будет создан новый каталог, причем он разместится не на клиентской машине, а на сервере резервного копирования.

ВНИМАНИЕ! Недостаток описанного способа резервного копирования состоит в том, что если клиент прекратит обмен по сети, процесс сохранения данных остановится на неопределенное время. В приведенном выше примере при монтировании указывалась опция -o soft. Она позволяет клиенту NFS на сервере резервного копирования передавать программе tar сведения об ошибках, предотвращая тем самым "зависание" процедуры копирования данных.

Использование SMB/CIFS

В предыдущем разделе рассматривалось использование утилиты tar совместно с программой rshd, выполняющейся на сервере резервного копирования, либо с сервером NFS, который выполняется на компьютере, выступающем в роли клиента резервного копирования. Для обеспечения взаимодействия компьютеров можно также использовать и другие серверы. В данном разделе рассматриваются вопросы выполнения резервного копирования с применением сервера SMB/CIFS. Такой способ создания резервных копий удобно использовать при работе в сетях, содержащих большое количество компьютеров под управлением Windows. В главе 7 описывался продукт Samba, с помощью которого можно организовать взаимодействие систем Windows и Linux по протоколу SMB/CIFS. Этот протокол может использоваться для создания резервных копий данных, содержащихся не только в системе Windows, но и в Linux. Если вы плохо представляете себе особенности конфигурации Samba, желательно перед прочтением данного раздела еще раз просмотреть главу 7.

Создание резервной копии клиента Windows с помощью сервера Linux

Резервное копирование, инициализируемое сервером, с применением Samba осуществляется почти так же, как и копирование по инициативе сервера с использованием NFS, но работа с продуктом Samba и системой Windows имеет ряд особенностей. Необходимо обратить внимание на возможности, предоставляемые программой smbtar, и специфику поддержки длинных имен Windows.

Объявление файлов для совместного использования

Как известно, для создания резервных копий по инициативе сервера необходимо, чтобы на клиенте резервного копирования выполнялась программа-сервер, обеспечивающая доступ к файлам. В большинстве случаев для взаимодействия с клиентами резервного копирования под управлением Windows удобно использовать протокол SMB/CIFS. Вы также можете запустить Samba на компьютере Linux, обеспечив тем самым доступ к файловой системе Linux и возможность резервного копирования содержащихся в ней данных. Однако при этом теряется информация о владельцах файлов и правах доступа (если говорить точно, эта информация может стать некорректной).

В составе системы Windows поставляются программы поддержки сервера SMB/CIFS, однако по умолчанию эти программные средства не установлены. Для включения необходимых программных компонентов необходимо использовать элемент в составе Control Panel, который в зависимости от версии системы называется Network или Network and Dial-Up Connections. Работая с системой Windows 9x/Me, следует дважды щелкнуть на пиктограмме Network, в результате чего отобразится диалоговое окно Network. В системе Windows NT или 2000 необходимо щелкнуть правой кнопкой мыши на соответствующем объекте в Network and Dial-Up Connections и выбрать пункт Properties. Необходимый вам компонент называется File and Printer Sharing for Microsoft Networks. Если данный компонент отсутствует, щелкните на Add или Install. Перечень сетевых служб, среди которых присутствует File and Printer Sharing for Microsoft Networks, показан на рис. 17.1. Возможно, вам потребуется задать принадлежность системы к рабочей группе. Сделать

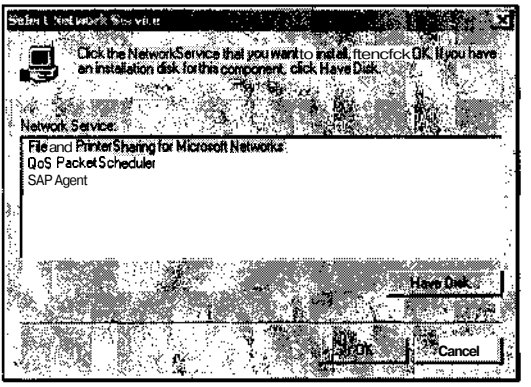


Рис. 17.1. Чтобы система Windows функционировала как сервер SMB/CIFS, в ней должен быть установлен компонент File and Printer Sharing for Microsoft Networks

это можно с помощью вкладки Identification диалогового окна Network (Windows 9x/Me) или объекта System в составе Control Panel (Windows 2000).

Инсталлировав сервер SMB/CIFS, вам надо организовать разделение дисков, содержимое которых вы хотите записывать на резервный носитель. Для этого выполните следующие действия.

1. В окне My Computer щелкните правой кнопкой мыши на устройстве, доступ к которому вы хотите разрешить, и в появившемся контекстном меню выберите пункт Sharing. (Если этот пункт отсутствует, то, вероятнее всего, программное обеспечением сервера SMB/CIFS не установлено.) В результате вы увидите диалоговое окно Properties, подобное изображенному на рис. 17.2.

2. Чтобы разрешить доступ к устройству, щелкните на опции Shared As или Share This Folder. При этом вам потребуется ввести имя разделяемого объекта, которое вы будете использовать при монтировании на сервере резервного копирования. В данном случае роль сервера резервного копирования выполняет компьютер под управлением Linux. (В системе Windows 2000 для ввода имени разделяемого объекта надо щелкнуть на New Share.)

3. При работе с Windows 9x/Me необходимо с помощью опции Access Type разрешить чтение и запись или только чтение и ввести пароль. Для создания резервных копий достаточно, чтобы данные были доступны только для чтения, но для восстановления данных необходимо также разрешить запись информации (в Windows 9x/Me, чтобы предоставить право чтения и записи, надо установить значение Full опции Access Type). В Windows 2000 с помощью вкладки Security можно определить, кто имеет право доступа к разделяемому объекту.

4. Щелкните на кнопке OK, разрешив тем самым совместное использование устройства.

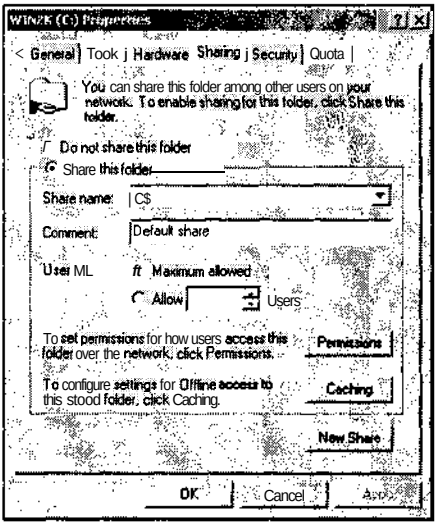


Рис. 17.2. Диалоговое окно Sharing системы Windows 2000. Аналогичное окно системы Windows 9x/Me содержит другой набор опций

5. Повторите пп. 1–4 для каждого устройства, содержимое которого необходимо записать на резервный носитель.

После выполнения указанных действий устройство становится доступным для удаленных компьютеров. Чтобы убедиться в наличии доступа, можно использовать броузер Network Neighborhood либо попробовать обратиться к файлу с другого узла сети. На компьютере под управлением Linux для этой цели можно использовать инструмент `smbclient`.

Использование smbtar

В составе пакета Samba поставляется программа smbtar. Как нетрудно догадаться, этот инструмент сочетает в себе возможности утилиты tar и клиента SMB/CIFS. На самом деле smbtar представляет собой сценарий оболочки, который вызывает программы tar и `smbclient`, используя предоставляемые ими возможности для создания резервных копий данных, которые содержатся на компьютерах под управлением Windows. Инструмент smbtar можно использовать как для создания резервной копии всего разделяемого объекта, так и для копирования отдельных файлов. Сценарий `smbtar` вызывается с помощью следующего выражения:

```
smbtar -s клиент_резервного_копирования \
[-x имя_разделяемого_объекта] [-u имя_пользователя] \
[-p пароль] [-d каталог] [-t устройство] [-r] [-v]
```

Обратившись к справочной системе, вы получите подробную информацию об использовании `smbtar`. Ниже описано назначение основных опций.

- s *клиент_резервного_копирования*. Эта единственная обязательная опция задает имя клиента резервного копирования. В качестве ее значения указывается NetBIOS-имя компьютера. В зависимости от значения опции name resolve order в файле smb.conf, система также может обрабатывать DNS-имена узлов сети.

- x *имя_разделяемого_объекта*. Данная опция позволяет задать имя разделяемого объекта (это имя вводится на этапе 2 описанной выше процедуры). По умолчанию принимается имя backup.

- и *имя_пользователя*. Если вы хотите установить соединение под именем, отличающемся от имени пользователя, под которым вы выполняете резервное копирование, вам необходимо указать данную опцию. Заметьте, что в Windows 9x/Me пользовательское имя не применяется, за исключением тех случаев, когда система входит в состав домена.

- p *пароль*. Если для работы с разделяемым объектом необходим пароль, вы можете задать его с помощью данной опции. При этом возникает серьезная угроза безопасности системы, так как значение пароля будет сохранено в списке предыстории, поддерживаемом оболочкой (в случае, если вы вводите команду smbtar вручную), кроме того, пароль отображается в перечне выполняемых процессов (соответствующие данные доступны посредством утилиты ps). Если же вы запускаете smbtar из сценария, необходимо проследить за тем, чтобы код сценария мог просматривать только пользователь root.

- d *каталог*. Если вы хотите работать лишь с одним каталогом, вы можете указать его имя с помощью данной опции. В случае, когда необходимо сохранить на резервном носителе весь разделяемый объект, опцию -d указывать не следует.

- t *устройство*. Эта опция позволяет указать файл устройства, соответствующий накопителю на магнитной ленте, или задать имя файла, в котором будет сохранена резервная копия. По умолчанию в качестве значения данной опции используется значение переменной окружения $TAPE, если же данная переменная не указана, принимается имя tar.out.

- r. По умолчанию smbtar используется для создания резервной копии. Если же указана опция -r, данная программа будет работать в режиме восстановления данных.

- v. Данная опция включает режим вывода дополнительной информации. Если опция -v задана, smbtar отображает имена копируемых файлов.

В качестве примера рассмотрим команду, которая создает резервную копию объекта CDRIVE на компьютере WORK. Эта команда имеет следующий вид:

```
# smbtar -s WORK -p password -x CDRIVE -t /dev/stO -v
```

При выполнении данной команды сначала выводится информация о состоянии системы, затем список файлов, а после этого — сведения о числе файлов и объеме сохраненных данных в байтах. Форматы файлов, созданных с помощью smbtar и tar, совпадают, поэтому при необходимости вы можете просмотреть содержимое архива посредством утилиты tar.

Использование smbmount

Вместо того чтобы работать с инструментом smbtar, вы можете воспользоваться предоставляемой Linux возможностью монтировать разделяемые объекты SMB/CIFS. Для монтирования подобных объектов можно применять утилиту mount или smbmount. При использовании программы mount надо указать тип файловой системы smbfs, задать NetBIOS-имя компьютера под управлением Windows, имя разделяемого объекта и имя пользователя. Сформированная таким образом команда имеет следующий вид:

```
# mount -t smbfs //WORK/CDRIVE /mnt/backup -o \
username=fred,password=password
```

Эквивалентная ей команда smbmount выглядит так:

```
# smbmount //WORK/CDRIVE /mnt/backup -o \
username=fred,password=password
```

Реализации утилиты smbmount в пакетах 2.0.x Samba существенно отличаются одна от другой. В ранних версиях данной программы использовался другой синтаксис. Приведенный выше вызов корректен для программ smbmount, поставляемых в составе версий 2.0.5a–2.2.2 Samba.

Если вы не укажете пароль, то программы mount и smbmount запросят его. Поэтому, если вы хотите вызывать программу резервного копирования из командной строки, эти утилиты предпочтительнее smbtar. Кроме того, используя mount или smbmount, вы можете смонтировать несколько устройств и скопировать их содержимое с помощью одного вызова утилиты tar. Такой подход упрощает создание резервной копии, но замедляет восстановление содержимого одной системы, так как в этом случае необходимо сначала прочитать информацию, соответствующую другим системам.

После окончания копирования содержимого компьютера под управлением Windows необходимо разорвать соединение с клиентом резервного копирования с помощью команды umount или smbumount. Пример вызова umount приведен ниже.

```
# umount /mnt/backup
```

Особенности обработки имен файлов Windows

Для создания резервных копий содержимого системы Windows часто используются компьютеры под управлением Linux. Однако при этом необходимо учитывать особенности обработки имен файлов. Дело в том, что программы mount и smbmount интерпретируют имена файлов иначе, чем это происходит в системе Windows. Для того чтобы понять эти различия, необходимо рассмотреть правила хранения файлов в Windows. Файловая система FAT (File Allocation Table — таблица размещения файлов), используемая в Windows 9x/Me и поддерживаемая в системах Windows NT, 2000 и XP, ориентирована на работу с файлами, имена которых содержат восемь символов, а расширение — три символа. Такие имена файлов называются *именами 8.3*. Для хранения длинных имен файлов в каталогах Windows предусмотрены дополнительные записи. Длинными считаются имена файлов, содержащие больше восьми символов имени и больше трех символов расширения, либо имена, составленные из символов, регистр которых должен быть сохранен. (Имена 8.3, в зависимости от используемых программ, могут отображаться символами верхнего регистра либо представляться как имя, начинающееся с прописной буквы, например File.txt. Существует также возможность указать, что имя 8.3 долж-

но представляться символами только верхнего или только нижнего регистра.) Проблема с обработкой имен файлов в Linux возникает из-за того, что в данной системе не поддерживаются имена 8.3, а используются только длинные имена файлов.

В отличие от Windows, Linux интерпретирует имена 8.3, которые присутствуют в каталогах, смонтированных с помощью mount или `smbmount`, как имена, состоящие только из символов нижнего регистра (например, `file.`txt). Такие файлы восстанавливаются корректно, но если при создании имени 8.3 было указано, что оно должно включать только символы нижнего регистра, процедура восстановления данных может представить его как состоящее из символов верхнего регистра. Такая особенность обработки имен редко приводит к возникновению серьезных проблем, поскольку при интерпретации имен файлов Windows не учитывает регистр.

Программа `smbtar` интерпретирует имена 8.3 как полностью состоящие из символов верхнего регистра, поэтому при ее использовании может возникнуть следующая проблема. Предположим, что в системе Windows был создан файл, имя которого полностью состоит из символов верхнего регистра, считается длинным именем, но содержит не больше восьми, а в составе расширения — не больше трех знаков. Программа smbtar, выполняемая в среде Linux, может интерпретировать такое имя как имя 8.3, в результате чего при восстановлении оно будет обработано некорректно, т. е. не будет указано, что имя является длинным и состоит из символов верхнего регистра. Данная проблема также не является `серьезной`, но может создавать неудобства для пользователей.

Существенные трудности возникают при создании имен 8.3, которые должны соответствовать длинным именам. В системе Windows эта задача решается автоматически; если в окне DOS, отображаемом в среде Windows, вы зададите команду DIR, вы увидите как короткие, так и длинные имена файлов. Поскольку система Linux не имеет информации о том, какие имена являются короткими, при восстановлении данных она полагается на соответствующие средства Windows. Как правило, регистр символов и другие особенности коротких имен не имеют значения для системы, но в некоторых случаях в результате несоответствия имен могут возникать нежелательные последствия. Например, если в конфигурационном файле указано короткое имя файла и если в результате восстановления данных это имя подверглось изменениям, программа не найдет требуемый файл. Кроме того, в системном реестре Windows некоторые имена файлов хранятся в формате 8.3, поэтому в результате восстановления файлов часть записей может оказаться некорректной. Это приведет к ошибкам в работе и даже к разрушению системы. Чтобы уменьшить вероятность возникновения подобных проблем, следует придерживаться приведенных ниже правил.

- **Используйте короткие имена каталогов.** Применяя короткие имена каталогов вместо длинных, вы устраните ряд проблем. Например, многие программы в системе Windows размещаются в каталоге `Program Files`. Если вы будете использовать вместо этого каталог с именем APPS, уменьшится вероятность того, что при восстановлении данных имя будет восстановлено некорректно. Аналогичным образом следует выбирать подкаталоги для установки программ. Имена файлов, содержащих данные, не обязательно должны быть короткими, так как информация о таких файлах практически никогда не помещается в системный реестр.

- **Используйте длинные имена в составе конфигурационных файлов.** Если вам необходимо включить в конфигурационный файл имя каталога, задавайте длинное имя. Например, если вы хотите задать каталог в качестве значения переменной

PATH в файле AUTOEXEC.BAT, используйте его полное имя. Этим вы достигнете того, что при внесении изменений в имена файлов 8.3 функционирование системы не изменится.

- **Создавайте длинные имена файлов, различающиеся первыми шестью символами.** При создании имени 8.3 Windows оставляет первые шесть символов низменными, затем присоединяет к ним символ ~ и порядковый номер, начинающийся с единицы. Например, имя longfilename.txt может быть преобразовано в имя 8.3 LONGFI~1.TXT. Если все длинные имена файлов в каталоге будут различаться первыми шестью символами, при преобразовании в формат 8.3 все они будут оканчиваться последовательностью ~1, в результате вероятность некорректного восстановления имени файла уменьшится.

Оканчивая разговор об обработке имен файлов, следует заметить следующее. Вероятность возникновения проблем при создании резервных копий Windows-файлов с использованием компьютера под управлением Linux невелика. Если же вы примете рекомендуемые меры предосторожности, она станет еще меньше. Однако полностью игнорировать ее нельзя, поэтому желательно создать в системе Windows вариант резервной копии, который можно было бы использовать для восстановления базовой конфигурации системы.

Файловая система NTFS (New Technology Filesystem — новая технология файловой системы), используемая в Windows NT, 2000 и XP, также поддерживает и имена 8.3, и длинные имена файлов. Однако вероятность возникновения проблем с преобразованием имен гораздо меньше, чем в тех системах, в которых используется FAT.

Разделяемые объекты резервного копирования

Один из способов создания резервных копий с применением Samba состоит в использовании разделяемых объектов резервного копирования. Разделяемый объект резервного копирования предоставляет собой объект Samba, связанный с устройством резервного копирования. Существует несколько подходов к работе с данными объектами, и все они могут рассматриваться как резервное копирование, инициируемое клиентом.

Что такое сервер резервного копирования

Существуют два основных подхода к созданию разделяемых объектов резервного копирования.

- **Непосредственное копирование.** Вы можете создать разделяемый объект, связанный с точкой монтирования устройства со сменными носителями (например, Zip или Jaz). После монтирования носителя клиент резервного копирования может обращаться к нему как к любому другому разделяемому объекту. Так, например, вы можете записывать на этот носитель данные, запуская программу создания архивов и даже перетаскивая необходимые файлы с помощью мыши.

- **Опосредованное копирование.** Для копирования на резервный носитель файлов, передаваемых клиентом, можно создать разделяемый объект, использующий сценарии. Сценарий может непосредственно записывать полученные данные либо выполнять их обработку. Пример использования сценария для записи данных на компакт-диск рассматривался в главе 7.

Возможности непосредственного копирования ограничены объемом сменных носителей. Лишь на жесткий диск можно записать всю информацию, содержащуюся на клиентской машине. Опосредованное копирование обеспечивает большую степень гибкости, однако полученные данные временно записываются на жесткий диск. Поэтому если вам необходимо сохранить большой объем информации, на диске сервера резервного копирования должно быть свободное пространство, достаточное для временного хранения данных, передаваемых клиентом.

Создание разделяемых объектов резервного копирования

Разделяемые объекты резервного копирования создаются так же, как и другие типы разделяемых объектов Samba. Различие состоит лишь в том, что вы, вероятнее всего, захотите использовать сценарии для автоматического монтирования устройства при первом обращении к нему и размонтирования его по завершении процедуры создания резервной копии. Кроме того, в определении объекта резервного копирования обычно присутствует параметр max connections, ограничивающий число пользователей, которые имеют возможность одновременно работать с данным объектом. Ниже приведено определение объекта резервного копирования, который позволяет удаленным пользователям записывать резервные копии данных на устройство Zip, смонтированное в позиции файловой системы /mnt/zip.

```
[zip]
    comment = Zip Backups
    path = /mnt/zip
    read only = No
    max connections = 1
    preexec = /bin/mount /mnt/zip
    postexec = /bin/umount /mnt/zip
```

Поскольку многие клиенты SMB/CIFS, в том числе средства, реализованные в системе Windows, не размонтируют разделяемый объект, вам, возможно, придется использовать глобальный параметр deadtime, который указывает на то, что после определенного периода бездействия соединение должно быть разорвано. Для устройства резервного копирования желательно задать параметр deadtime = 5. Значение этого параметра сообщает, что соединение будет разорвано через пять минут после прекращения активности.

Разделяемые объекты резервного копирования, подобные приведенному выше, применяются для копирования ограниченного объема данных. Такие объекты удобно использовать в небольших сетях для предоставления пользователям доступа к Zip и другим подобным устройствам.

На сменных носителях может использоваться любая файловая система, поддерживаемая Linux. Однако, если эти носители должны читаться в других системах, вам, возможно, придется отказаться от применения некоторых форматов. Например, если пользователям необходимо читать данные, записанные на сменном диске в системе Windows, очевидно, что на нем должна быть сформирована файловая система FAT. Если же диски будут устанавливаться только на устройствах, подключенных к компьютеру под управлением Linux, то на нем может присутствовать FAT, ext2fs или любая другая файловая система, с которой может работать Linux.

Совместное использование некоторых носителей обеспечить достаточно сложно. Для работы с ними необходимо создать разделяемый объект, который предпринимал бы специальные меры по записи данных на носитель. В главе 7 были приведены два примера разделяемых объектов, которые записывали информацию на компакт-диск. Модифицированный вариант разделяемого объекта, описанного в главе 7, представлен ниже.

```
[backup]
    path = /var/spool/samba
    printable = Yes
    print command = /usr/local/bin/samba-backup %H %s %U \
                    /var/spool/samba; rm %s
```

Данный объект определяет псевдопринтер, который получает от клиента резервного копирования zip-файлы. Для извлечения содержимого zip-файла и копирования его на ленту с помощью tar этот объект использует сценарий /usr/local/bin/samba-backup, код которого приведен в листинге 17.1. В результате получается копия данных на ленте, аналогичная той, которая создается с помощью программы smbtar. Модифицировав сценарий или изменив параметр print command, вы можете организовать непосредственную запись zip-файла на ленту. Это позволит предотвратить потерю признаков скрытых и системных файлов при извлечении содержимого zip-файла в системе Linux.

Листинг 17.1. Сценарий, поддерживающий работу разделяемого объекта резервного копирования, реализованного в виде псевдопринтера

```
#!/bin/sh
# $1 = Рабочий каталог пользователя, который передал
#      задание на обработку
# $2 = Имя zip-файла
# $3 = Имя пользователя, который передал задание на обработку
# $4 = Путь к zip-файлу
mkdir -p $1/backup/samba
cd $1/backup/samba
unzip $4/$2
tar cvpf /dev/st0 ./ > $1/tar.out
mail -s "Backup finished" $3 < $1/tar.out
rm $1/tar.out
rm -r $1/backup/samba
```

ВНИМАНИЕ Описанный здесь подход предполагает, что файл устройства резервного копирования доступен всем пользователям. Существуют, однако, способы обойти это требование. Например, вы можете использовать для всех соединений учетную запись root. Также можно включить в определение разделяемого объекта в файле smb.conf параметр force user, который устанавливал бы идентификатор пользователя, по инициативе которого выполняется команда "печати". Вы можете создать специальную группу, членам которой накопитель на магнитных лентах был бы доступен для чтения и записи; в этом случае вам потребуется параметр force group.

Данный подход может быть использован для обработки файлов, отличных от zip-фалов. Например, если система получает tar-файлы, то их можно непосредственно скопировать на ленту. В этом случае отпадает необходимость в извлечении файлов (чему посвящена основная часть сценария в листинге 17.1). Такой подход обеспечивает более высокую скорость обработки данных, но менее удобен при работе с клиентами Windows, поскольку формат tar в основном используется в системе Linux, а в Windows он применяется крайне редко.

Применение разделяемых объектов резервного копирования

Как было описано в главе 7, применение псевдопринтера предполагает создание файла архива на локальном компьютере и передачу его на сервер резервного копирования без использования драйвера принтера. При желании вы можете создать в системе Windows bat-файл, который после щелчка на пиктограмме выполнял бы все действия, связанные с созданием резервной копии. По завершении процедуры копирования сценарий, код которого представлен в листинге 17.1, посылает по почте отчет, содержащий список скопированных файлов.

Поскольку использование псевдопринтера предполагает передачу файла архива, вы можете использовать этот способ для создания резервной копии данных Linux, при этом сведения о файлах сохраняются в той мере, в которой формат архива обеспечивает их поддержку. Если вы настроите псевдопринтер для приема tar-файла, вы можете использовать данный метод для создания резервной копии информации, содержащейся на клиентах под управлением Linux. Рассматриваемый подход обеспечивает более высокую степень защиты по сравнению с использованием сервера `rshd`. Работая с Samba, вы можете ограничивать доступ на основании IP-адреса компьютера, в то время как при работе с `rshd` необходимо также указывать пароль. Недостатком данного способа является тот факт, что для организации резервного копирования нужно очень много свободного пространства на диске. Процедура создания резервной копии занимает много времени, кроме того, если два пользователя одновременно передадут задания на создание резервной копии, возможен конфликт.

Использование AMANDA

В предыдущих разделах данной главы рассматривались основные способы создания резервных копий и конфигурация компьютеров, необходимая для реализации этих способов. Если в вашей сети содержится относительно небольшое число компьютеров, вы, возможно, станете создавать резервные копии вручную либо напишете сценарий для автоматизации выполнения данной задачи. Если же число узлов в сети велико, вам понадобятся более сложные инструменты. Одним из таких инструментов является AMANDA (Advanced Maryland Automatic Network Disk Archiver). Данный пакет объединяет различные инструменты, предназначенные для выполнения резервного копирования в сети, и ориентирован в основном на работу в сетях небольшого и среднего размера, но может применяться и в больших сетях. Программное обеспечение AMANDA поставляется в составе версий Linux Debian, Red Hat, Mandrake и SuSE. Если же требуемые программы отсутствуют, вы можете скопировать их с Web-страницы AMANDA (`http://www.amanda.org`).

ВНИМАНИЕ В исполняемых файлах AMANDA закодированы некоторые значения, используемые при работе данного инструмента, поэтому если вы попытаетесь объединить компоненты AMANDA, предназначенные для выполнения в различных системах, то они, вероятнее всего, работать не будут. Если в вашей операционной среде установлено большое количество различных пакетов, рекомендуется самостоятельно построить создать AMANDA из исходных кодов. Убедитесь, что во всех системах посредством опций `--with-user` и `--with-group` заданы один и тот же пользователь и группа. Возможно, вы захотите создать учетную запись и группу, специально предназначенные для выполнения различных операций с помощью AMANDA.

Выполнение AMANDA

Пакет AMANDA включает программное обеспечение как сервера, так и клиента. Серверные программы следует установить на компьютере, выполняющем функции сервера резервного копирования, а клиентские программы — на клиенте резервного копирования. Для взаимодействия между собой эти программы применяют собственные протоколы. NFS, `rshd` и другие подобные протоколы при работе AMANDA не используются. (При выполнении резервного копирования данных, расположенных на компьютерах под управлением Windows, AMANDA использует программу `smbclient` и стандартный сервер SMB/CIFS.)

AMANDA не только поддерживает сетевые соединения, но и выступает в роли инструмента планирования. Эта возможность помогает осуществлять резервное копирование, а для сети большого размера планирование является неотъемлемой частью работ по администрированию систем. Так, например, вы вряд ли собираетесь при каждой операции резервного копирования создавать копию каждого файла на каждом компьютере. Подобная политика создала бы огромную нагрузку на сеть, и процесс копирования длился бы несколько суток. Для того чтобы минимизировать влияние процедуры резервного копирования на работу сети, при запуске простого инструмента наподобие tar задается опция `--listed-incremental` (или эквивалентная ей). AMANDA позволяет указать, какой компьютер должен участвовать в резервном копировании в конкретный момент времени и должно ли выполняться полное или инкрементное копирование данных.

Подобно серверу Samba, используемому для создания резервных копий, AMANDA в обычных условиях сначала копирует данные с клиентской машины на жесткий диск сервера резервного копирования, а затем записывает эти файлы на ленту. (При необходимости вы можете настроить AMANDA так, что данные будут непосредственно записываться на ленту, но такая конфигурация снизит производительность программ.) Сервер AMANDA лучше всего работает при наличии жесткого диска большого объема. На этом диске должны помещаться системные программы и резервные копии данных, созданные по крайней мере в течение одного дня. Если же объем диска будет вдвое увеличен, вы сможете записывать имеющиеся данные на ленту и одновременно копировать информацию из сети. В принципе AMANDA может работать и при нехватке свободного места на диске, но в этом случае резервное копирование будет осуществляться по частям. Например, если объем свободного дискового пространства составляет 1 Гбайт, AMANDA скопирует 1 Гбайт информации с клиентской машины на диск, запишет эти данные на ленту, снова скопирует 1 Гбайт данных и т. д.

Настройка клиентских машин для использования AMANDA

AMANDA осуществляет резервное копирование, инициируемое сервером, поэтому на компьютере, выступающем в роли клиента, должна выполняться программа-сервер. Данная программа, предназначенная для работы в системах Linux и UNIX, поставляется в составе пакета AMANDA и называется **amandad**. Эта программа-сервер обычно запускается с помощью суперсервера. Соответствующая запись в конфигурационном файле **/etc/inetd.conf** имеет следующий вид:

```
amanda dgram udp wait amanda amandad amandad
```

Данная запись запускает сервер amandad от имени пользователя amanda. Учетная запись, соответствующая этому пользователю, должна присутствовать в системе. При необходимости вы можете изменить данную запись в соответствии с особенностями системы, например, вам, возможно, придется указать полный путь к исполняемому файлу amandad. Если в вашей системе используется суперсервер **xinetd**, вы должны будете создать запись в его конфигурационном файле. Формат конфигурационного файла **xinetd** рассматривался в главе 4.

В документации на пакет AMANDA описывается использование учетной записи и группы, специально предназначенных для выполнения резервного копирования, но такой подход часто не дает результатов. Чтобы сервер amandad работал, его следует запустить от имени пользователя **root**.

Для запуска сервера AMANDA с помощью суперсервера в файл **/etc/services** необходимо включить специальную запись. Она может выглядеть следующим образом:

```
amanda 10080/udp
```

После того как вы создали записи в конфигурационном файле суперсервера и в файле **/etc/services**, вам следует перезагрузить суперсервер. В результате клиент резервного копирования AMANDA станет доступен для сервера резервного копирования. Остальные действия по настройке выполняются на компьютере, выступающем в роли сервера резервного копирования.

При выполнении резервного копирования данных, содержащихся на узлах сети, необходимо также создать резервную копию сервера резервного копирования. Для этого на сервере обычно устанавливается программное обеспечение клиента резервного копирования.

Для работы клиента резервного копирования необходим файл авторизации с именем **.amandahosts**, находящийся в рабочем каталоге пользователя, который запускает AMANDA. В этом файле должны быть указаны полностью определенное доменное имя сервера резервного копирования и имя пользователя, разделенные пробелом или знаком табуляции. Например, приведенная ниже запись позволяет пользователю amanda на сервере **buserver.threeroomco.com** создавать резервные копии данных.

```
buserver.threeroomco.com amanda
```

Как было замечено ранее, для создания резервных копий данных, расположенных на компьютере под управлением Windows, AMANDA использует сервер SMB/CIFS. На-

стройка клиентов Windows резервного копирования для взаимодействия по протоколу SMB/CIFS рассматривалась выше в данной главе.

Настройка сервера резервного копирования AMANDA

Поскольку с точки зрения сетевого взаимодействия сервер резервного копирования действует как клиент, на этом компьютере не нужно программное обеспечение сервера. Тем не менее следует заметить, что AMANDA поддерживает восстановление данных по инициативе клиента, поэтому в пакете AMANDA содержатся две серверные программы, предназначенные для выполнения на сервере резервного копирования. В результате пользователь, работающий на компьютере, выполняющем функции клиента резервного копирования, может просматривать имеющиеся данные и инициировать процесс восстановления информации. Серверные программы на сервере резервного копирования обычно запускаются посредством суперсервера. Соответствующие записи в файле `/etc/inetd.conf` имеют следующий вид:

```
amandaidx stream tcp nowait amanda amindexd amindexd
amidxtape stream tcp nowait amanda amidxtaped amidxtaped
```

Как и для программ, выполняющихся на клиенте резервного копирования, вам, возможно, придется указать полный путь к исполняемым файлам. Если же в вашей системе используется `xinetd`, вам надо создать в конфигурационном файле записи по соглашениям, описанным в главе 4. Чтобы указанные программы могли запускаться посредством суперсервера, надо включить в файл `/etc/services` следующие строки:

```
amandaidx 10082/tcp
amidxtape 10083/tcp
```

Пользователь, запускающий программы AMANDA, должен иметь право читать и записывать информацию на магнитную ленту. В противном случае AMANDA не будет иметь доступа к устройству и не сможет создавать резервные копии данных.

Формирование конфигурационного файла AMANDA

Особенности работы пакета AMANDA определяются содержимым конфигурационных файлов `amanda.conf`, которые обычно находятся в подкаталогах каталога /etc или `/usr/local/etc`. Обычно конфигурационные файлы AMANDA размещаются на двух уровнях подкаталогов. Эти подкаталоги доступны для чтения только пользователю, который должен работать с пакетом AMANDA (в данном примере это пользователь amanda). Подкаталог верхнего уровня обычно называется amanda, а имена подкаталогов нижнего уровня выбираются в соответствии с задачами резервного копирования. Например, конфигурационный файл, описывающий правила ежедневного копирования, может размещаться в каталоге `/usr/local/etc/amanda/Daily`, а файл, определяющий создание архивов, — в каталоге `/usr/local/etc/amanda/Archive`. Если вы использовали для инсталляции AMANDA исходные коды, в вашем распоряжении имеется образец конфигурационного файла. Он находится в каталоге example инсталляционного пакета.

Установка основных опций

В состав файла `amanda.conf` входит набор строк, и каждая из них начинается с ключевого слова, за которым следует одно или несколько значений. Например, запись,

указывающая, как долго длится процесс полного копирования содержимого всей сети, выглядит следующим образом:

```
dumpcycle 4 weeks
```

Некоторые из записей могут занимать несколько строк. Они помещаются в фигурные скобки и определяют набор связанных между собой опций.

Для большинства опций можно принять значения, установленные по умолчанию. Опции, которые вы, возможно, захотите изменить, описаны ниже.

- `org`. С помощью данной опции задается название организации, которое затем указывается в отчетах, генерируемых AMANDA. Значение опции org никак не влияет на процесс создания резервной копии.

- `mailto`. О выполненных действиях AMANDA сообщает, посылая письмо по адресу, указанному в качестве значения этой опции. При необходимости вы можете задать несколько адресов, разделив их пробелами.

- `dumpuser`. Посредством данной опции указывается имя пользователя, инициирующего процедуру резервного копирования. При инсталляции AMANDA это имя задается с помощью опции `--with-user`.

- dumpcycle. В качестве значения этой опции указывается число дней, составляющих цикл резервного копирования.

- runspercycle. AMANDA может запускаться каждый день, несколько раз в день или один раз в несколько дней. Периодичность запусков задается с помощью данной опции. Предположим, что цикл резервного копирования, указанный посредством опции dumpcycle, составляет четыре недели. Установив значение runspercycle, равное 20, вы сообщаете, что данные для копирования должны выбираться исходя из того, что AMANDA будет запускаться один раз в день в рабочие дни. Значение 4 говорит о том, что AMANDA будет запускаться один раз в неделю. (Заметьте, что реальный запуск AMANDA осуществляется с помощью инструмента `cron`. Значение опции runspercycle не влияет на периодичность запуска, оно лишь позволяет AMANDA планировать, какие данные должны быть скопированы на резервный носитель.)

- tapecycle. Значение данной опции определяет число магнитных лент, используемых в цикле резервного копирования. Учитывая, что некоторые носители могут быть повреждены, значение tapecycle должно быть несколько больше, чем значение опции runspercycle.

- tapetype. Если вы сообщите о том, какой тип накопителя вы используете, AMANDA сможет определить, как долго будет длиться запись информации. В конфигурационном файле, поставляемом в составе конфигурационного пакета, указано несколько типов накопителей. Если вы не найдете тип, соответствующий вашему устройству, вам придется выяснить эти сведения самостоятельно. Сделать это можно с помощью утилиты tapetype, которая поставляется в составе пакета AMANDA, но по умолчанию не устанавливается. Перейдите в каталог `tape-src` инсталляционного пакета и задайте команду make tapetype. Затем установите в устройстве ненужную вам ленту и введите `./tapetype -f /dev/`*`устройство`*(где

под устройством понимается файл, соответствующий накопителю на ленте). В результате выполнения данной команды вы получите набор значений для вашего устройства. Для завершения данной процедуры потребуется несколько часов, и все данные на ленте будут уничтожены. Если в вашем накопителе реализована встроенная функция сжатия данных, вы можете увеличить значение длины ленты в полтора-два раза. При этом соблюдайте осторожность: если вы будете копировать данные, плохо поддающиеся сжатию, места на ленте может не хватить.

- tapedev. Данная опция задает файл устройства Linux, представляющий интерфейс *без перемотки* к накопителю на ленте. В большинстве случаев в качестве значения этой опции указывается имя /dev/nstO или /dev/nht0.

- netusage. С помощью этой опции указывается максимальная пропускная способность линий, на которую может рассчитывать AMANDA при выполнении резервного копирования.

- labelstr. Значением этой опции является регулярное выражение, которое AMANDA использует для назначения имен лентам с резервными копиями данных. Имена лент учитываются при их подготовке. Особенности подготовки лент рассматриваются в следующем разделе.

- tpchanger,changerfile и changerdev. Если в вашем накопителе предусмотрена автоматическая смена ленты, вы можете управлять устройством с помощью специальных файлов. Примеры этих файлов находятся в каталоге example инсталляционного пакета.

- infofile, logdir и indexdir. Расположение файлов протоколов AMANDA задается с помощью опций infofile и logdir. Кроме того, опция indexdir определяет индексный файл, содержащий список скопированных файлов. Значения этих опций можно оставить без изменения.

Кроме установки значений приведенных выше опций, вам также надо описать область для хранения данных. Для этого используется опция holdingdisk, значение которой занимает несколько строк и состоит из ряда подопций. К ним относятся directory (каталог для хранения файлов) и use (пространство на диске, которое может быть использовано для хранения данных). Если на диске сервера резервного копирования недостаточно места, вы можете задать отрицательное значение chunksize. При этом файл, размер которого превышает абсолютное значение chunksize, непосредственно записывается на ленту, минуя область хранения. (Положительные значения chunksize указывают на то, что большие файлы должны быть разбиты на части для размещения в области хранения. Такой подход удобно применять в тех случаях, когда файловая система на диске или особенности ядра не позволяют обрабатывать большие файлы. Например, при использовании версий ядра 2.2.x на компьютерах x86 максимальный размер файлов составляет 2 Гбайт.)

Подготовка лент

Для работы AMANDA необходимо, чтобы лента была подготовлена к использованию. Подготовка осуществляется с помощью утилиты amlabel. Эта утилита должна запускаться от имени пользователя, который выполняет резервное копирование. Вызов amlabel выглядит следующим образом:

```
$ amlabel Daily DailySet123
```

Указанный здесь параметр Daily задает подкаталог, в котором размещается конфигурационный файл amanda.conf. В результате amlabel получает доступ к необходимым опциям. Параметр DailySet123 представляет собой метку ленты. Это значение должно соответствовать регулярному выражению, заданному в качестве значения опции labelstr в файле amanda.conf, в противном случае AMANDA не сможет работать с лентой. В большинстве случаев AMANDA копирует данные из сети на несколько лент. Чтобы вы могли различать ленты, вам необходимо разработать схему их именования.

Определение типов резервных копий

В конце файла amanda.conf, поставляемого в составе инсталляционного пакета AMANDA, содержится несколько записей dumptype. Они определяют, как должно выполняться резервное копирование содержимого клиентского компьютера или отдельной файловой системы. В составе записи dumptype могут указываться следующие опции.

- compress [client | server] [best | fast | none]. Вы можете задать тип сжатия, выполняемого клиентом или сервером копирования. Тип сжатия выбирается в зависимости от используемого процессора и сетевых ресурсов. Значение best указывает на то, что необходимо выполнить наиболее эффективное сжатие, для которого требуется большое количество ресурсов процессора. Значение fast задает быстрое, но менее эффективное сжатие. Значение none запрещает сжатие данных.

- exclude [list] *"строка"*. В зависимости от того, указано ли значение list, AMANDA включает заданную строку в качестве значения опции --exclude или --exclude-from утилиты tar.

- holdingdisk *логическое_значение*. Задавая логическое значение yes или no, вы можете указать AMANDA, следует ли использовать область диска для хранения данных.

- index *логическое_значение*. Задавая логическое значение равным yes или no, вы сообщите AMANDA о том, следует ли создавать перечень файлов, скопированных на резервный носитель. Такой файл может оказать помощь в работе с резервными копиями, но он занимает место на диске, которое можно использовать для других целей.

- kencrypt *логическое_значение*. Данная опция позволяет указать, следует ли кодировать данные, передаваемые по сети, с помощью протоколов Kerberos. Значение, равное yes, можно указывать только в том случае, когда сеть сконфигурирована для работы с Kerberos. Вопросы использования Kerberos обсуждались в главе 6.

- program *"строка"*. AMANDA может использовать либо утилиту tar, либо программу dump, ориентированную на работу с конкретной операционной системой. Данная опция позволяет указать, какую из этих программ следует использовать. По умолчанию AMANDA работает с программой dump (значение DUMP данной опции). Чтобы задать использование tar, необходимо, чтобы значение опции было равно GNUTAR. (При работе с Samba по умолчанию применяется утилита tar.)

- skip-incr *логическое_значение*. Если значение данной опции равно true, то при инкрементном копировании файловая система, для которой указан этот тип резервной копии, не учитывается.

В определении типа резервной копии могут присутствовать и другие опции. Некоторые из них, например dumpcycle, обсуждались в предыдущем разделе. Информацию о других опциях вы найдете в справочной системе. Большинство определений типов резервных копий начинается с имени другого определения. Это означает, что новое определение создано на основе существующего и для него справедливы значения опций базового определения. Например, в конфигурационном файле amanda.conf, поставляемом в составе инсталляционного пакета, присутствует определение с именем global, включаемое в другие определения.

Заметьте, что при выполнении одной операции резервного копирования могут быть созданы копии различных типов. Например, вы можете скопировать важные системные файлы, не используя сжатие, но сжимать при копировании менее важную информацию. Вы также можете использовать утилиту dump для копирования разделов ext2fs, а утилиту tar — для копирования разделов ReiserFS.

Определение данных для копирования

В файле amanda.conf содержатся важные опции, управляющие процессом копирования, но отсутствуют сведения о клиентах резервного копирования или каталогах, содержимое которых необходимо записать на резервный носитель. Эта информация указывается в файле disklist, который находится в том же каталоге, что и amanda.conf. В составе инсталляционного пакета AMANDA содержится пример файла disklist. Очевидно, что рабочий вариант этого файла, созданный с учетом конфигурации вашей системы, будет существенно отличаться от образца.

Содержимое файла disklist представляет собой набор записей, каждая из которых содержится в отдельной строке и состоит из трех полей. В этих полях указывается имя компьютера, выступающего в роли клиента резервного копирования, область для копирования и тип резервной копии для этой области. В качестве области для копирования может быть указано имя устройства (например, /dev/hda2 или hda2) либо точка монтирования файловой системы (например, /home). Строки, начинающиеся с символа #, содержат комментарии. Пример простого файла disklist приведен в листинге 17.2.

Листинг 17.2. Пример содержимого файла disklist

```
# Создание резервной копии сервера резервного копирования
buserver.threeroomco.com  /                  root-tar
buserver.threeroomco.com  /var               user-tar
buserver.threeroomco.com  /hold              holding-disk
# Создание резервной копии клиента Linux или UNIX
buclient.threeroomco.com  /                  root-tar
buclient.threeroomco.com  /home              user-tar
# Создание резервной копии клиента Windows
buserver.threeroomco.com  //WINPC/DRIVEC     user-tar
```

Большинство записей в этом примере не нуждается в комментариях. Раздел /hold компьютера buserver.threeroomco.com содержит область для хранения данных. В определении соответствующего типа резервной копии указано значение по опции holdingdisk, что предотвращает использование этой области для временного хранения своего же содержимого. Если в данном разделе не содержится ничего, кроме области хранения данных, вы можете исключить его из процесса копирования. Для копирования данных клиента Windows указывается имя компьютера Linux или UNIX, на котором установлен и выполняется сервер Samba, а также NetBIOS-имя клиента Windows (WINPC) и имя разделяемого объекта (DRIVEC). В листинге 17.2 в качестве компьютера, на котором установлен продукт Samba, указан сам сервер резервного копирования, но при необходимости вы можете задать имя другого компьютера. (Заметьте, что, установив Samba на сервере резервного копирования, вы уменьшаете нагрузку на сеть.) Чтобы создать резервную копию устройства на компьютере под управлением Windows, его не нужно монтировать; AMANDA обращается к системе Samba для использования smbclient. Samba использует smbclient так же, как обычный клиент резервного копирования утилиту tar или dump. В файловой системе компьютера, на котором выполняется Samba, необходимо создать файл /etc/amandapass. В этом файле следует указать имя разделяемого объекта и пароль. В качестве пользовательского имени AMANDA передает имя SAMBA, поэтому при работе с Windows NT, 2000 или XP этот пользователь должен присутствовать в системе. Для того чтобы изменить имя пользователя, применяемое по умолчанию, вам надо при установке AMANDA задать это имя в качестве значения опции --with-samba-user.

Создание резервных копий с помощью AMANDA

Для того чтобы инициировать процесс создания резервной копии с помощью AMANDA, необходимо запустить на сервере резервного копирования программу amdump. Введите имя программы и укажите после нее данные для копирования, т. е. задайте имя каталога, в котором находятся требуемые конфигурационные файлы. Например, команда может выглядеть так: amdump Daily. Очевидно, что перед запуском amdump вам необходимо установить на устройстве ленту, подготовленную так, как было описано выше. В результате выполнения данной команды AMANDA обнаружит области, предназначенные для копирования, обработает их содержимое и запишет на магнитную ленту. Процедура резервного копирования не обязательно будет охватывать все компьютеры в сети и даже все разделы на одном компьютере. В зависимости от значения опции dumpcycle и других опций подобного назначения, указанных в конфигурационном файле, а также от типа устройства резервного копирования, AMANDA может принять решение скопировать данные лишь с нескольких компьютеров или вместо полного копирования выполнить инкрементное копирование. Если продолжительность цикла резервного копирования не слишком мала, в течение этого цикла AMANDA выполнит резервное копирование всех компьютеров в сети.

Занимаясь администрированием сети, вы вряд ли захотите вручную вызывать программу amdump. Лучше всего запускать ее с помощью инструмента cron в то время, когда нагрузка на сеть минимальна, например ночью. Следует убедиться, что в эти часы клиенты резервного копирования доступны. Многие пользователи, уходя с работы, выключают свои компьютеры. Вам придется убедить их не делать этого.

По завершении резервного копирования AMANDA посылает по почте отчет о выполненных действиях. Адрес пользователя, которому должно быть передано сообщение, указывается в качестве значения опции mailto, расположенной в файле amanda.conf. Изучив этот отчет, вы получите сведения о том, успешно ли завершилось копирование или при его выполнении возникли ошибки. Так, например, некоторые компьютеры могли оказаться не доступными, а емкость ленты — меньше ожидаемой.

Восстановление данных

До сих пор мы рассматривали вопросы создания резервных копий. Однако, для того, чтобы обеспечить надежную работу компьютеров в сети, необходимо также организовать восстановление данных. Действия по восстановлению данных можно условно разделить на две описанные ниже категории.

- Частичное восстановление. При выполнении частичного восстановления необходимо извлечь с резервного носителя лишь некоторые файлы. Например, пользователю могут понадобиться файлы, которые были по ошибке удалены на прошлой неделе, или у вас может возникнуть необходимость просмотреть старые файлы протоколов. Для этого надо выполнить действия, обратные созданию резервной копии. Например, если для создания копии вы использовали опцию **--create** утилиты tar, то для восстановления файлов вам надо задать опцию **--extract** и указать имена файлов или каталогов, которые вы собираетесь восстановить. Как было сказано ранее, если вы смонтировали некоторую файловую систему и выполняли резервное копирование по инициативе сервера, вам следует убедиться, что программа-сервер, которая выполняется на компьютере, выступающем в роли клиента резервного копирования, предоставляет серверу резервного копирования право записи данных. Это требование не выдвигалось при создании резервной копии.

- Полное восстановление. В этом случае предполагается восстановление всех файлов на диске или по крайней мере тех данных, которые нужны для загрузки компьютера. Необходимость в полном восстановлении может возникнуть при повреждении жесткого диска или содержащихся на нем программ. Так, например, полное восстановление придется производить после того, как от имени пользователя root была выполнена команда rm -r /.

ВНИМАНИЕ Полное восстановление может понадобиться для того, чтобы воспроизвести состояние системы, которое было до вмешательства злоумышленника в ее работу. Однако в этом случае необходимо действовать очень осторожно, так как в результате восстановления данных потеряется информация о характере атаки. Перед тем как восстанавливать состояние системы, вам необходимо проанализировать недостатки системы, которыми смог воспользоваться хакер для ее взлома.

Полное восстановление данных представляет собой сложную задачу, так как вам необходимо найти способ записи данных на компьютер, на котором нет работоспособных программ. Чтобы решить эту проблему, многие системные администраторы создают минимальную конфигурацию системы на дискете, компакт-диске или другом носителе. Для работы в сети этот вариант системы должен содержать средства поддержки сетевого

взаимодействия, а также программы, обеспечивающие работу клиента резервного копирования.

СОВЕТ

Даже если в вашей сети присутствует сервер резервного копирования под управлением Linux и большое число клиентов резервного копирования под управлением Windows 9x/Me, вы можете использовать для восстановления данных минимальную конфигурацию системы Linux. В этой системе должны присутствовать средства Samba, которые вы используете, чтобы запустить сервер SMB/CIFS. Этот сервер необходим для восстановления файлов с сервера резервного копирования Linux. После того как полное восстановление данных окончится, вам необходимо запустить программу FDISK системы DOS, чтобы пометить раздел как загружаемый, а затем с помощью программы SYS записать данные в загрузочный сектор раздела. При работе с Windows NT, 2000 или XP процесс восстановления данных будет более сложным, особенно, если на компьютере используется файловая система NTFS. В этом случае вам надо первоначально установить систему в небольшой загрузочный раздел. Содержимое этого раздела можно сохранять и восстанавливать посредством утилиты dd, работающей в системе Linux, или с помощью одного из коммерческих инструментов, например DriveImage.

В некоторых случаях целесообразно использовать способ полного восстановления данных, отличный от того, который применялся для создания резервной копии. Например, если резервная копия создавалась по инициативе клиента с использованием разделяемого объекта резервного копирования, восстановление проще осуществить путем непосредственного извлечения содержимого tar-архива с помощью rshd или по инициативе сервера с использованием NFS или Samba.

В некоторых случаях, в особенности тогда, когда минимальная конфигурация системы, пригодная для восстановления данных, не была создана, обеспечить работу клиента резервного копирования можно, повторно инсталлировав базовую операционную систему. Эта система используется для восстановления остальных данных, причем в этом случае можно ограничиться частичным восстановлением.

AMANDA отличается от других средств, рассмотренных в данной главе, тем, что в состав этого пакета входят инструменты восстановления данных, предназначенные для выполнения на стороне клиента резервного копирования. Наиболее мощным из этих инструментов является amrecover, который вызывает в процессе работы другие программы, например amrestore. Если вы запустите amrecover на клиенте резервного копирования от имени пользователя root, программа отобразит приглашение для ввода команды. Допустимыми командами являются setdate (определение даты резервной копии), cd (изменение текущего каталога), add (добавление файла к набору, предназначенному для восстановления) и extract (восстановление файлов). После указания команды extract программа amrecover предложит установить соответствующую ленту, содержащую резервную копию.

Независимо от того, какой метод вы используете для восстановления данных, этот метод необходимо опробовать до того, как его придется применять на практике. Желательно установить специальную тестовую систему и использовать ее для проверки процедуры создания резервной копии и восстановления данных. Если в вашей сети имеется несколько различных операционных систем, проверьте, как выполняется восстановление данных в каждой из них. Подробно опишите действия по восстановлению данных, а в случае

изменения структуры сети убедитесь в том, что данная процедура по-прежнему дает ожидаемые результаты. Если вы собираетесь использовать для полного восстановления данных минимальную конфигурацию системы, создайте несколько копий диска. Согласно закону Мерфи, именно тогда, когда вам понадобится выполнять полное восстановление данных, вы обнаружите, что диск, содержащий минимальную конфигурацию системы, испорчен.

Резюме

Существует несколько различных способов создания резервных копий данных. Вы можете выполнять резервное копирование по инициативе клиента или по инициативе сервера, применяя при этом NFS, SMB/CIFS, rshd, AMANDA и другие средства сетевого взаимодействия, а также tar, dump, cpio и другие программы создания архивов. Выбор конкретных средств зависит от особенностей сети. Представляя себе особенности различных способов создания резервных копий, вы сможете принять решение, наиболее приемлемое для вашей сети. Если число компьютеров в сети относительно невелико, для создания резервных копий подойдут простые инструменты наподобие утилиты tar. При работе в больших сетях вам понадобятся сложные инструменты, например AMANDA. Создавая резервные копии, необходимо иметь в виду, что рано или поздно они понадобятся для восстановления данных. Восстановление данных нельзя рассматривать лишь как процесс, обратный резервному копированию. Для эффективного восстановления утерянных данных часто приходится применять дополнительные инструменты.

ЧАСТЬ III

Серверы Internet

Глава 18

Администрирование домена

Для того чтобы компьютеры, подключенные к сети TCP/IP, могли обращаться друг у другу по имени, они должны иметь возможность преобразовывать доменные имена в IP-адреса, а также выполнять обратное преобразование. Существуют различные средства, позволяющие осуществлять подобные преобразования. Наиболее часто для этой цели используется *DNS-сервер* (Domain Name System — система доменных имен), который часто называют *сервером имен*. В главе 2 рассматривались вопросы настройки компьютера для взаимодействия с сервером имен. Однако, для того, чтобы такое взаимодействие было возможным, сервер DNS должен присутствовать в сети. Следует заметить, что работа всей Internet базируется на использовании иерархии серверов DNS. Установив сервер DNS в локальной сети, вы не только обеспечите преобразование имен в ее пределах, но также предоставите возможность внешним пользователям обращаться к компьютерам вашей сети по именам.

Администрирование сервера DNS предполагает создание различных конфигурационных файлов, управляющих его работой. Некоторые из таких файлов должны задавать конфигурацию доменов, поддерживаемых сервером. Возможно, что при администрировании домена вам придется заниматься вопросами, непосредственно не связанными с обеспечением работы сервера DNS, например регистрировать домен. Не исключено, что вам потребуется обеспечивать взаимодействие сервера DNS с другими серверами в сети, например с сервером DHCP.

Администрирование сервера DNS не занимает столько времени и не требует таких усилий, сколько требуется для управления сложными системами, например, Kerberos выполнять все же труднее, чем администрировать простой сервер, например Telnet. Прочитав данную главу, вы получите сведения, достаточные для управления простым доменом. Если же потребуется установить более сложную конфигурацию, вам придется обратиться к документации на сервер DNS либо к изданиям, специально посвященным данному вопросу. В качестве примеров можно привести книгу Элбица (Albitz) и Лиу (Liu) *DNS and BIND, 4th Edition* (O'Reilly, 2001) и книгу Ханга (Hunt) *Linux DNS Server Administration* (Sybex, 2000).

Использование сервера DNS

Администрирование сервера DNS — нетривиальная задача, для выполнения которой администратор должен обладать определенной квалификацией. Чтобы грамотно настроить сервер, необходимо знать принципы его работы.

Сервер DNS, доступный из внешней сети

Работа сети Internet базируется на использовании взаимодействующих между собой серверов DNS. Чтобы понять, как сервер, установленный в локальной сети, позволяет внешним пользователям обращаться к локальным компьютерам по именам, необходимо рассмотреть процесс обмена данными между различными серверами DNS. Предположим, что к серверу DNS обратился Web-броузер, пользователь которого задал URL `http://www.whitehouse.gov`. Сервер DNS должен преобразовать символьное имя `www.whitehouse.gov` в IP-адрес узла сети. Локальный сервер DNS начинает процесс преобразования с того, что обращается к корневому серверу DNS. IP-адреса компьютеров, выполняющих функции корневых серверов, указаны в конфигурационном файле каждого сервера DNS; вероятность того, что эти адреса изменятся, крайне мала. Обращаясь к корневому серверу DNS, локальный сервер передает ему имя, предназначенное для преобразования (в данном примере это имя `www.whitehouse.gov`). В большинстве случаев корневой сервер не знает IP-адреса, соответствующего указанному имени, но имеет информацию о *доменах верхнего уровня* (TLD — top-level domain). Примерами таких доменов являются `.com`, `.gov`, `.uk` и т. д. Поэтому корневой сервер возвращает локальному серверу адреса компьютеров, поддерживающих DNS-сервер . gov, после чего локальный сервер передает запрос одному из компьютеров, обеспечивающих работу домена . gov. Сервер DNS . gov также не может преобразовать имя, но он знает IP-адреса компьютеров, ответственных за поддержку домена whitehouse . gov, поэтому передает их локальному серверу DNS. Сервер `whitehouse.gov` знает IP-адрес, соответствующий имени `www.whitehouse.gov`, поэтому, получив запрос локального сервера, он возвращает ему требуемую информацию. После получения IP-адреса локальный сервер DNS передает его Web-броузеру, который использует адрес при формировании запроса к Web-серверу. Процесс преобразования адресов условно показан на рис. 18.1. Детали этого процесса скрыты от пользователя. С точки зрения прикладной программы или работающего с ней пользователя дело обстоит так, как будто локальный сервер DNS имеет информацию о соответствии символьных имен и IP-адресов всех узлов Internet.

На первый взгляд описанная здесь процедура кажется сложной. Создается впечатление, что для ее выполнения потребуется много времени, однако реально преобразование имени завершается за несколько секунд. Существуют также способы ускорить этот процесс. Один из них состоит в том, что локальные серверы DNS кэшируют результаты преобразования адресов. Поэтому если два пользователя передадут серверу DNS запросы на преобразование одного и того же имени и между этими запросами пройдет не слишком много времени, то второй запрос будет обработан мгновенно. Сервер DNS лишь вернет второму пользователю результаты, полученные для первого пользователя. В кэше также сохраняются адреса серверов тех доменов, обращения к которым осуществляются наиболее часто. Например, локальный сервер DNS почти наверняка имеет сведения о серверах домена верхнего уровня . com, поэтому он не станет обращаться к корневому домену. Информация хранится в кэше в течение ограниченного времени, поэтому, если админи-

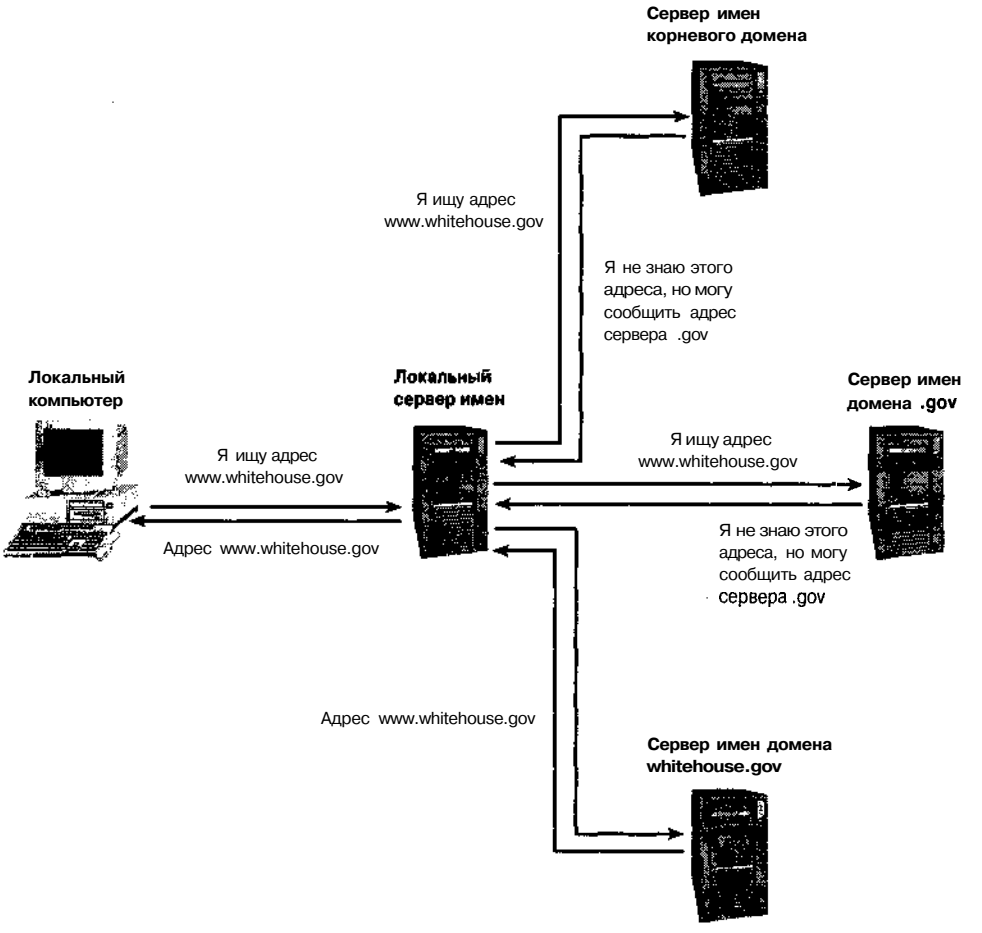


Рис. 18.1. Процесс преобразования адреса предполагает передачу запросов различным серверам DNS

стратор изменит IP-адрес одного из своих компьютерах, это не приведет к возникновению проблем при обращении к этой машине.

Если вы хотите, чтобы к компьютерам вашей сети можно было обращаться извне по именам, вам необходимо включить его в состав иерархии DNS. Другими словами, вы должны установить сервер DNS, который будет выполнять для вашего домена те же действия, которые сервер `whitehouse.gov` выполняет для своего домена. После того как ваш сервер будет настроен и запущен, а также после того как сервер DNS, поддерживающий домен более высокого уровня, узнает о существовании вашего сервера, каждый пользователь Internet сможет преобразовать имена компьютеров вашей сети в IP-адреса.

Официальные организации, осуществляющие поддержку DNS, требуют, чтобы каждый домен обслуживался как минимум двумя серверами DNS. Несмотря на то что для поддержки протокола преобразования имен достаточно одного сервера DNS, второй сервер создает избыточность, позволяющую повысить надежность системы. Если один их серверов выйдет из строя, второй сможет обрабатывать запросы. Если размеры вашей сети малы, вы можете установить в ней лишь один *(первичный)* DNS-сервер, а второй *(вторичный)* сервер разместить за пределами сети.

Вместо установки собственного сервера DNS, вы можете использовать один из серверов, предоставляющих услуги по преобразованию имен. Этим занимаются многие организации, осуществляющие поддержку DNS, и провайдеры Internet. Часто подобные услуги предоставляются бесплатно, в других случаях за них надо платить, но плата невелика (обычно она составляет несколько долларов в год). Так, например, бесплатное DNS-обслуживание предлагает организация Granite Canyon (`http://www.granitecanyon.com`). Советую вам не слишком полагаться на бесплатные услуги; помните правило: "Вы получите то, за что заплатите". В любом случае сторонние организации предоставят вам возможность разместить либо один, либо оба сервера DNS, необходимых для обслуживания вашей сети.

Существует специальное *динамическое* DNS-обслуживание, которое предоставляется в основном пользователям DSL и кабельных модемов. Организации, которые предоставляют динамическое DNS-обслуживание, поддерживают обычные серверы DNS, но они позволяют быстро обновлять записи, которые соответствуют изменяющимся IP-адресам. Многие организации предпочитают использовать статические IP-адреса и статические средства преобразования имен. Динамическое DNS-обслуживание в основном предоставляется отдельным пользователям, поддерживающим собственные серверы. Динамическое DNS-обслуживание обеспечивают многие организации. Полный их список занял бы слишком много места, поэтому здесь приводятся лишь два URL: `http://www.technopagan.org/dynamic/` и `http://www.oth.net/dyndns.html`.

Если вы не уверены в том, что сможете должным образом осуществлять поддержку сервера DNS, воспользуйтесь услугами сторонних организаций. Учитывая, что бесперебойная работа DNS важна для нормальной работы других служб, имеет смысл предоставить возможность настраивать и обслуживать сервер квалифицированным специалистам.

Даже если вы приняли решение использовать внешние серверы DNS, вам необходимо хотя бы в общих чертах представлять себе конфигурацию сервера такого типа. Дело в том, что заполняя формы для провайдеров DNS, вам придется указывать ту же информацию, которую вы включили бы в конфигурационные файлы локального сервера DNS. Вопросы конфигурации домена будут рассмотрены далее в этой главе.

Работа локального сервера DNS

Локальный сервер DNS не только предоставляет возможность внешним пользователям обращаться к компьютерам вашей локальной сети по именам. Он также выполняет преобразование символьных имен в IP-адреса для узлов локальной сети, т. е. выполняет те же функции, что и серверы, адреса которых содержатся в файле `/etc/resolv.conf`. Использование локального сервера DNS предоставляет следующие преимущества.

- Преобразование адресов по запросам компьютеров локальной сети выполняет сервер, находящийся в той же сети, что увеличивает производительность работы. Увеличение производительности особенно заметно тогда, когда сервер DNS провайдера работает медленно или ненадежно. Локальный сервер поддерживает собственный кэш, что также способствует повышению быстродействия.

- Локальный сервер DNS предоставляет локальным компьютерам информацию о других машинах, подключенных к той же сети, даже о тех, которые не доступны внешним пользователям. Так, локальный сервер DNS может использоваться для обслуживания сети, защищенной брандмауэром. При этом компьютер, на котором выполняется сервер DNS, не обязательно должен быть доступен извне.

Таким образом, локальный сервер не только позволяет внешним пользователям обращаться к вашим компьютерам по именам, но и может использоваться для обслуживания внутренней сети. Решение об установке сервера DNS полностью оправдано в том случае, когда ваша локальная сеть отделена от Internet брандмауэром и к ней подключено достаточно большое количество компьютеров.

В простых сетях вместо сервера DNS можно использовать другие средства преобразования адресов. Так, например, в системах Linux и UNIX для этой цели можно применить файл `/etc/hosts`. (Аналогичное средство доступно и в прочих системах, но соответствующий файл расположен в другом каталоге. Например, в Windows 9x/Me подобные функции выполняет файл `C:\WINDOWS\HOSTS`.) В составе файла `/etc/hosts` содержатся записи; каждая из них состоит из IP-адреса, за которыми следуют полное доменное имя и сокращенный вариант имени компьютера. Пример записи из файла `/etc/hosts` приведен ниже.

```
192.168.78.109 gingko.threeroomco.com gingko
```

При установке системы Linux в файл `/etc/hosts` помещается единственная запись, которая связывает имя `localhost` с адресом 127.0.0.1. Если в локальной сети содержится небольшое число компьютеров, этот файл несложно дополнить так, чтобы он определял все узлы сети. В небольшой сети отредактировать файлы `/etc/hosts` гораздо проще, чем настроить сервер DNS. При увеличении размеров сети для редактирования файлов `/etc/hosts` приходится прилагать все больше и больше усилий, в то время как затраты на поддержку сервера DNS увеличиваются лишь незначительно. Применение файла `/etc/hosts` теряет смысл, если в вашей сети присутствует сервер DHCP и используется динамическое распределение IP-адресов.

Получение доменного имени

Задачи запуска сервера DNS и получения доменного имени тесно связаны между собой. Без доменного имени сервер DNS не сможет обслуживать внешних пользователей, так как ссылка на него должна присутствовать на вышестоящем сервере.

Если вы собираетесь установить сервер DNS во внутренней сети, можете самостоятельно выбрать имя домена. Необходимо лишь следить за тем. чтобы оно отличалось от всех доменов, используемых в Internet. Сделать это можно, приняв несуществующее имя домена верхнего уровня, например `.unused`.

В настоящее время существуют два основных типа доменов верхнего уровня.

- Домен верхнего уровня на базе кода страны (ccTLD — country code top-level domain). Эти домены принадлежат конкретным странам. Например, домен .us принадлежит США, а .se — Швеции.

- Универсальный домен верхнего уровня (gTLD — generic top-level domain). Эти домены не отражают географическое положение узла сети. Примерами подобных доменов являются .com, .net, .org и .gov. Начиная с 2001 г. стали доступны новые gTLD; к ним относятся, например, .biz и .museum.

Процесс регистрации домена зависит от того, принадлежит ли ваш домен ccTLD или gTLD. Кроме того, процедура регистрации имеет свои особенности для разных доменов верхнего уровня внутри категории. В большинстве случаев для получения имени домена следует обращаться к одной из организаций, поддерживающей реестр доменных имен. Большинство таких организаций имеют право распределять домены в составе .com, .org, .net и ряда других TLD. Некоторые страны выделяют домены в своем ccTLD на коммерческой основе. При этом имя домена может получить организация, не имеющая никакого отношения к стране, которой принадлежит домен верхнего уровня. Списки организаций, поддерживающих реестр доменных имен, можно найти по адресам http://www.NewRegistrars.com и http://www.icann.org/registrars/accredited-list.html. Стоимость регистрации домена в составе gTLD обычно составляет от 10 до 35 долларов в год.

В некоторых доменах верхнего уровня, например в gTLD .gov и .edu, а также во многих ccTLD регистрация доменов затруднена. Список ccTLDs, включающий информацию об организациях, ответственных за распределение доменных имен, можно найти по адресу http://www.iana.org/cctld/cctld-whois.htm.

В конце 2001 г. были изменены принципы управления доменом верхнего уровня .us. С начала 2002 г. появилась возможность регистрировать домены непосредственно в TLD .us. Если вы собираетесь зарегистрировать свой домен в составе .us, вам следует обратиться по адресу http://www.nic.us.

Некоторые домены, в особенности ccTLD, содержат иерархию поддоменов. Например, в составе домена .uk созданы поддомены .gov.uk и .co.uk, предназначенные для конкретных целей. Если организация захочет зарегистрировать домен, непосредственно принадлежащий TLD .uk, ей будет отказано в этом. В различных поддоменах действуют разные правила регистрации новых доменов. Например, .gov.uk выделен для государственных учреждений Великобритании, а домен .co.uk — для коммерческих организаций (он выполняет те же функции, что и gTLD .com).

При регистрации домена вам придется представить некоторую информацию о себе, например почтовый адрес и номер телефона. Вам также необходимо сообщить IP-адреса двух серверов DNS, настроенных для поддержки домена. Если вы установили собственные серверы DNS, вы можете сами сообщить их адреса. Если вы хотите, чтобы DNS-услуги предоставляла для вас другая организация, вы попадаете в сложное положение. С одной стороны, чтобы пользоваться услугами внешнего сервера DNS, вы должны зарегистрировать домен, а с другой стороны, чтобы зарегистрировать домен, вам нужны серверы DNS. Разорвать этот замкнутый круг можно с помощью сервера DNS, принадлежащего регистрирующей организации. Многие провайдеры также готовы предоставить свои услуги на время регистрации домена.

Серверы DNS для Linux

Первое, что необходимо сделать при установке сервера DNS, — решить, какой продукт вы будете использовать в качестве сервера. Разные программы предоставляют различные возможности. Наиболее часто используемые пакеты описаны ниже.

- **BIND.** Сервер BIND (Berkeley Internet Name Domain) — самая популярная в настоящее время программа, которая может обеспечивать функции сервера DNS в системе Linux. Именно этому серверу уделяется основное внимание в данной главе. Пакет BIND поставляется в составе многих дистрибутивных пакетов, кроме того, вы можете скопировать его с узла `http://www.isc.org/products/BIND/`. В настоящее время доступна версия 9.2.0, но на момент написания данной книги, т. е. в 2002 г., многие дистрибутивные пакеты Linux еще поставлялись с версиями 8.2.x данного продукта. Заметьте, что формат конфигурационного файла старой версии 4.9.x отличается от формата, используемого в новых версиях сервера.

- djbdns. D. J. Bernstein's DNS server (сервер DNS Д. Дж. Бернстайна) представляет собой продукт, альтернативный BIND, пользующийся популярностью у некоторых пользователей. Этот сервер отличается небольшими размерами, высокой эффективностью и обеспечивает высокий уровень защиты. Он не принят в качестве стандарта и не поставляется ни с одним из дистрибутивных пакетов, рассматриваемых в данной книге. При желании вы можете заменить BIND на `djbdns`. Дополнительная информация о djbdns содержится на Web-странице `http://cr.yp.to/djbdns.html`.

- pdnsd. Данный продукт представляет собой демон, реализующий proxy-сервер DNS. Он ориентирован для использования в локальной сети в качестве посредника между локальными компьютерами и внешним сервером DNS. Он также предоставляет ограниченные средства преобразования имен, но не поддерживает все возможности BIND или `djbdns`. Дополнительную информацию о pdnsd можно найти по адресу `http://home.t-online.de/home/Moestl/`.

- dnscache. Подобно pdnsd, dnscache представляет собой proxy-сервер DNS. Он предназначен для ускорения процесса преобразования имен. В отличие от pdnsd, dnscache не подерживает локальные компьютеры, за исключением узла `localhost` (127.0.0.1). Информацию о данном продукте можно получить, обратившись по адресу `http://cr.yp.to/djbdns/dnscache.html`.

Большинство администраторов, занимающихся поддержкой компьютеров под управлением Linux, используют в качестве сервера DNS продукт BIND, поскольку он принят как стандарт и поставляется с большинством версий данной операционной системы. Администраторы, для которых вопросы безопасности системы имеют первоочередное значение, отдают предпочтение продукту djbdns. Proxy-серверы DNS в основном используются в небольших сетях для кэширования результатов запросов к внешним серверам и преобразования локальных имен. Если же вы хотите поддерживать собственный домен и выполнять преобразование имен по запросам извне, возможности подобных продуктов не позволяют решать эти задачи. Остальной материал данной главы посвящен рассмотрению BIND, но некоторые действия по администрированию этого сервера применимы также к djbdns.

Базовая конфигурация DNS

Установка конфигурации DNS предполагает решение двух задач: настройка сервера DNS (в пакете BIND функции сервера выполняет программа named) и администрирование домена. В данном разделе обсуждаются особенности выполнения первой задачи, а администрированию домена посвящен следующий раздел. (Далее в этой главе будут рассмотрены использование локального сервера DNS для кэширования результатов преобразования имен и интеграция BIND с сервером DHCP.) При настройке сервера DNS устанавливаются основные опции, указывается расположение других серверов DNS (в частности, серверов корневого домена) и задается информация о поддерживаемых зонах. Даже для небольшого домена необходим вторичный (ведомый) сервер имен. В принципе вы можете установить конфигурацию вторичного сервера, скопировав содержимое соответствующих файлов, созданных для локального сервера, но гораздо лучше настроить вторичный сервер так, чтобы от автоматически дублировал параметры первичного сервера.

Главный конфигурационный файл BIND

Основные опции BIND задаются в главном конфигурационном файле с именем `named.conf`. Этот файл обычно располагается в каталоге /etc. В некоторых дистрибутивных пакетах Linux файл с опциями, установленными по умолчанию, в каталоге /etc отсутствует. В этом случае файл-образец надо искать в каталоге, содержащем документацию BIND (обычно это каталог `/usr/share/doc/bind-`*версия*). Пример содержимого файла `named.conf` приведен в листинге 18.1.

Листинг 18.1. Пример файла `named.conf`

```
options {
    directory "/var/named/";
    auth-nxdomain yes;
    forwarders {
        10.232.7.98;
        10.232.45.1;
    };
    forward first;
};

zone "." {
    type hint;
    file "named.ca";
};
zone "threeroomco.com" {
    type master;
    file "named.threeroomco.com";
};

zone "1.168.192.in-addr.arpa"{
    type master;
```

```
    file "named.192.168.1";
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};
```

Файл named.conf состоит из нескольких разделов. В листинге 18.1 представлены раздел options и несколько разделов zone. Раздел options содержит определения глобальных опций, в частности, в нем задается каталог, в котором содержатся файлы с описанием зоны. Разделы zone описывают конкретные зоны — домены либо другие группы имен или IP-адресов. Большинство строк, содержащихся в файле named.conf, оканчиваются точкой с запятой (;). Это требование надо выполнять, в противном случае BIND может некорректно интерпретировать содержимое конфигурационного файла. В основном содержимое файла named.conf представляет собой указатели на файлы, в которых находятся дополнительные сведения о зонах. Эти файлы содержатся в каталоге /var/named либо в другом каталоге, заданном с помощью опции directory.

В последующих разделах информация, содержащаяся в файле named.conf, описывается более детально.

Расположение других серверов имен

Одна из основных задач, которые вам предстоит решить при инсталляции сервера DNS, — получить список корневых серверов. Сделать это можно несколькими способами.

- Требуемый файл может входить в поставку пакета BIND. Обычно он называется named.ca или db.cache и располагается в каталоге /var/named. Если содержимое этого файла устарело, вы можете получить новый файл одним из двух описанных ниже способов.

- Файл named.ca, содержащий список корневых серверов, можно скопировать посредством протокола FTP, обратившись по адресу ftp://ftp.rs.internic.net/domain/.

- Если в вашей системе установлена программа dig, вы можете задать команду dig @a.root-servers.net . ns > named.ca. Эта команда копирует файл, содержащий список корневых серверов, и присваивает ему имя named.ca.

Чтобы вы могли воспользоваться вторым или третьим из описанных выше способов, в вашей сети должен работать сервер DNS. Если сервер DNS в сети отсутствует, вы можете скопировать нужный файл, воспользовавшись компьютером другой сети, либо временно настроить компьютер, на котором должен быть установлен сервер DNS для преобразования посредством внешнего сервера имен (действия по настройке были описаны в главе 2).

Получив файл со списком корневых серверов, скопируйте его в каталог /var/named. Кроме того, вам следует убедиться в том, что ссылка на этот файл присутствует в конфигурационном файле /etc/named.conf. В листинге 18.1 файл, содержащий список

корневых серверов, указан с помощью опции `file`, расположенной в разделе `zone "."`. (Каждое доменное имя должно оканчиваться точкой, но имя корневой зоны состоит только из точки.)

Настройка сервера для перенаправления запросов

BIND осуществляет преобразование имен одним из трех описанных ниже способов.

1. Если пакет BIND настроен для поддержки запрошенного имени, сервер возвращает адрес, указанный в его конфигурационном файле.

2. Если запрашиваемый адрес находится в кэше, сервер возвращает его. В этом случае повышается быстродействие и снижается нагрузка на сеть.

3. Если требуемый адрес в кэше отсутствует, сервер передает запрос корневому серверу и другим серверам. Типичная процедура преобразования имен была рассмотрена выше. Для выполнения преобразования требуется лишь несколько секунд, но чтение из кэша осуществляется гораздо быстрее.

В пакете BIND реализована еще одна возможность. Он может перенаправить запрос серверу DNS, чтобы тот выполнял всю рутинную работу по преобразованию адресов. Настроенный таким образом, пакет BIND осуществляет перенаправление запроса после того, как убеждается, что в кэше необходимые данные отсутствуют, но перед тем, как приступить к стандартной процедуре преобразования. В некоторых ситуациях такое перенаправление может повысить скорость преобразования имен. Произойдет это в том случае, если сервер, установленный в небольшой локальной сети, подключенной к Internet через линию с низкой пропускной способностью, будет перенаправлять запрос другому локальному серверу, который имеет возможность передавать данные по более быстродействующим линиям. Предположим, что вы выполняете администрирование сети, подключенной к Internet по коммутируемой линии или через спутниковое соединение. В этом случае имеет смысл перенаправить запросы на преобразование имен серверу DNS провайдера. Соединение по коммутируемой линии само по себе обладает низким быстродействием, а спутниковое соединение характеризуется большой задержкой, поэтому для передачи многочисленных запросов при выполнении стандартной процедуры преобразования имен потребуется слишком много времени.

Почему же не настроить локальные компьютеры так, чтобы они непосредственно обращались к серверу DNS провайдера? Локальный сервер DNS удобно использовать для преобразования локальных имен, кроме того, кэширование результатов обработки запросов повышает быстродействие при работе с Internet. Помимо того, сервер DNS провайдера может функционировать нестабильно, поэтому наличие собственного сервера позволяет повысить надежность при установлении сетевых соединений.

Перенаправление запросов задается с помощью опций `forwarders` и `forward` (см. листинг 18.1). Опция `forwarders` позволяет задать один или несколько IP-адресов серверов DNS, к которым локальный сервер станет обращаться перед тем, как начать выполнение стандартной процедуры преобразования адресов. Опция `forward` допускает одно из двух значений: `only` или `first`. Если вы зададите `forward only`, BIND при работе будет полагаться лишь на удаленный сервер DNS, указанный с помощью опции `forwarders`, и не будет выполнять стандартную процедуру преобразования имен. Значение `first` опции `forward` указывает на то, что BIND должен сначала обратиться

к удаленному серверу DNS, а если такое обращение не дало результатов (например, если удаленный сервер не ответил на запрос), он должен осуществить преобразование имен по стандартному алгоритму. В любом случае, если к серверу поступит запрос на преобразование имен, которые принадлежат зоне, обслуживаемой данным сервером, он будет использовать описание зоны в своем конфигурационном файле.

Описание зоны

При настройке BIND вы должны указать, как следует обрабатывать запросы, в которых указаны определенные домены, поддомены и диапазоны IP-адресов. Различные группы имен и адресов называются зонами. Зоной может быть домен или поддомен (например, зоной является домен `threeroomco.com`, указанный в листинге 18.1) либо диапазон IP-адресов (такой диапазон присутствует в имени зоны, оканчивающемся символами `in-addr.arpa`). Для простого сервера DNS задается лишь несколько зон.

- **Корневая зона.** Зона, идентифицируемая посредством точки (“.”), определяет корневой узел пространства имен. В определении зоны присутствует опция `type hint`, которая сообщает о том, что список серверов содержится в файле, указанном посредством опции `file`.

- **Локальный домен.** Если ваш сервер DNS предназначен не только для кэширования, вам придется сконфигурировать BIND для поддержки зоны, соответствующей вашему домену. В листинге 18.1 примером такой зоны является `threeroomco.com`.

- **Обратная зона.** Несмотря на то что в большинстве случаев сервер DNS используется для преобразования символьных имен в IP-адреса, сервер имен также должен поддерживать обратное преобразование. Для выполнения подобного преобразования создается зона, имя которой оканчивается на `in-addr.arpa`. Перед этой последовательностью символов указывается имя зоны, т. е. часть IP-адреса, заданная в обратном порядке. Например, запись для сети 192.168.1.0/24 имеет имя `1.168.192.in-addr.arpa`.

В определении зоны указывается тип зоны и список конфигурационных файлов, предоставляющих дополнительную информацию о зоне. Типы зон описаны ниже.

- `master`. *Первичный,* или *ведущий* (master), сервер содержит описание зоны. Если вы создаете сервер DNS для небольшой сети, то, вероятнее всего, объявите локальные зоны как `master`. Пример зоны такого типа приведен в листинге 18.1.

- `slave`. *Вторичный,* или *ведомый* (slave), сервер получает информацию о зоне от другого сервера DNS. Такой сервер также может выступать в роли источника информации о зоне. В следующем разделе данный тип зоны будет рассмотрен более подробно. Простой сервер DNS может выступать для некоторых зон как ведущий, а для других зон — как ведомый.

- `stub`. Сервер такого типа похож на ведомый, но он копирует только записи NS, т. е. спецификации сервера имен. Данный тип зоны следует использовать в том случае, если вы хотите создать отдельный сервер DNS для поддомена. Предположим, что домен `threeroomco.com` содержит поддомен `sub.threeroomco.com`

и вы хотите использовать для управления им отдельный сервер DNS. Для этого вы должны включить в состав конфигурационного файла сервера BIND для threeroomco.com определение зоны с именем sub.threeroomco.com типа stub, указывающее на сервер DNS sub.threeroomco.com. Вы можете также использовать один сервер DNS для поддержки всего домена, включая поддомен sub.threeroomco.com. В этом случае вам не понадобится формировать специальную зону sub.threeroomco.com.

- forward. Подобно опции forward в разделе options, зона forward сообщает BIND, что запросы на получение информации о зоне должны передаваться другому серверу DNS. BIND формирует собственный запрос к указанному серверу, а затем использует полученные данные для построения ответа. Используя зону такого типа, вы должны включить опцию forwarders, указывающую BIND, какому из удаленных серверов DNS должны перенаправляться запросы.

- hint. Зона этого типа используется только для описания корневых серверов имен. Она сообщает системе, где следует искать список таких серверов. BIND должен самостоятельно обновлять данный список, обращаясь к корневому серверу имен.

Простой конфигурационный файл, подобный представленному в листинге 18.1, содержит одну зону hint и несколько зон master. В случае более сложной конфигурации в конфигурационном файле могут быть указаны зоны всех типов.

Настройка ведомого сервера

Если вы собираетесь зарегистрировать домен, вам необходимо настроить два сервера DNS. Обычно один сервер конфигурируют как ведущий, а другой — как ведомый. Как и ведущий, ведомый сервер хранит информацию о зоне в отдельных файлах. Различие между этими серверами состоит в том, что ведомый сервер получает информацию о зонах от ведущего сервера. Для того чтобы это стало возможным, надо задать специальным образом конфигурацию зоны в файле /etc/named.conf. Предположим, например, что вам необходимо настроить сервер так, чтобы он выполнял функции ведомого по отношению к серверу, конфигурационный файл которого приведен в листинге 18.1. На ведомом сервере зона threeroomco.com должна быть определена следующим образом:

```
zone "threeroomco.com" {
    type slave;
    file "named.threeroomco.com";
    masters { 192.168.1.50; }
};
```

Приведенная выше запись указывает на то, что ведомый сервер должен получать содержимое конфигурационного файла для threeroomco.com с сервера DNS, расположенного по адресу 192.168.1.50. Если сервер функционирует как ведомый для нескольких доменов, в его конфигурационном файле содержится несколько подобных определений. В списке masters можно указать два и более серверов DNS; их адреса отделяются друг от друга точкой с запятой. (При необходимости ведомый сервер может синхронизировать свое содержимое посредством другого ведомого сервера.) Если сервер является ведомым для нескольких зон, различные зоны могут синхронизироваться от разных ведущих серверов. Эту возможность удобно использовать в том случае, если вы администрируете

несколько доменов. Каждый домен обслуживается отдельным ведущим сервером, и для всех их роль ведомого может выполнять один сервер.

Зоны типа `slave` должны быть определены не только для прямого, но и для обратного преобразования адресов (зоны `threeroomco.com` и `1.168.192.in-addr.arpa`, приведенные в листинге 18.1). Для корневых серверов имен и для обратного преобразования `localhost` (`0.0.127.in-addr.arpa` в листинге 18.1) зоны типа `slave` создавать не надо.

Если сконфигурировав сервер как ведомый, вы запустите его на выполнение, то вскоре увидите, что сервер создал файлы описания зоны, указанные в записях zone. Если этого не произошло, необходимо просмотреть файлы протоколов как для ведущего, так и для ведомого сервера и определить причину их некорректной работы. Возможно, что ведущий сервер сконфигурирован так, что передача зоны запрещена. Такой запрет часто устанавливается по соображениям безопасности, чтобы внешние пользователи не могли получить информацию о компьютерах, принадлежащих вашему домену. Если вы хотите ограничить право передачи зоны ведущим или ведомым сервером DNS, вам надо задать опцию `allow-transfer`. Сделать это можно либо в разделе `options`, либо в определении конкретной зоны. Например, чтобы ограничить право передачи зоны компьютерами 192.168.1.0/24 и 172.19.98.23, надо создать следующую запись:

```
allow-transfer {
    192.168.1/24;
    172.19.98.23;
};
```

Управление доменом

Несмотря на то что вы создали файл `/etc/named.conf`, указали в нем глобальные опции и определили зоны, оказывается, что настройка ведущего сервера DNS не закончена и запускать его еще рано. Если в файле `/etc/named.conf` указана зона типа master, необходим также конфигурационный файл зоны. В этом файле указываются имена узлов и IP-адреса. Если сервер имен обслуживает домен, содержащий большое число узлов, создание и поддержка файла зоны составляет значительную часть работы по администрированию сети. Если же домен невелик и состояние его не изменяется, достаточно лишь один раз создать конфигурационный файл.

Пример конфигурационного файла зоны

В листинге 18.2 приведен пример простого конфигурационного файла зоны. Этот файл начинается с имени зоны (`threeroomco.com.`) и раздела, в котором определяются параметры домена по умолчанию. Эти параметры детально рассматриваются ниже. За этим разделом следует набор записей, предоставляющих информацию о соответствии имен компьютеров и IP-адресов. Некоторые из записей относятся ко всему домену. Подробно структура записей будет рассмотрена позже.

ВНИМАНИЕ! В системе DNS имена узлов должны оканчиваться точкой. Если, работая с клиентскими программами Internet (Web-броузером, клиентом FTP, программой подготовки почты), вы не укажете завершающую точку, система Linux сначала посчитает, что имя домена не указано, поэтому добавит к имени узла имя

домена, заданное в /etc/resolv.conf. Если попытка преобразования такого имени окажется неудачной, система вместо имени домена добавит к имени, указанному пользователем, точку и снова попытается выполнить преобразование. Эти действия скрыты от пользователя, поэтому он может пренебрегать точкой в конце имени без ущерба для себя. Настраивая сервер DNS, администратор не может позволить себе подобной небрежности. В конфигурационном файле можно задавать либо полностью определенное доменное имя, оканчивающееся точкой, либо имя узла, не содержащее имени домена. Если вы забудете указать точку в конце полного доменного имени, система добавит к нему имя домена. Сформированное таким образом имя будет иметь вид gingko.threeroomco.com.threeroomco.com.

Листинг 18.2. Простой конфигурационный файл зоны

```
threeroomco.com.    IN   SOA   spruce.threeroomco.com. \
                               admin.threeroomco.com. (
                    2002043004 ; serial (последовательный номер)
                    3600       ; refresh (обновление)
                    600        ; retry (повторное обращение)
                    604800     ; expire (срок действия)
                    86400      ; default_ttl (время жизни)
                    )
gingko.threeroomco.com.     IN A      192.168.1.1
birch                       IN A      192.168.1.2
spruce                      IN A      192.168.1.3
threeroomco.com.            IN A      192.168.1.4
www                         IN CNAME  gingko
kelp                        IN CNAME  jacques.pangaea.edu.
@                           IN MX     10 birch.threeroomco.com.
@                           IN MX     20 mail.pangaea.edu.
@                           IN NS     spruce.threeroomco.com.
```

Формат записи в конфигурационном файле зоны выглядит следующим образом:

имя IN *тип_записи содержимое_записи*

Здесь под именем подразумевается имя компьютера или псевдоимя, связанное с адресом и применяемое для обратного преобразования. Идентификатор IN сокращенно означает Internet и определяет класс записи. Кроме IN существуют и другие классы записей, но в данной главе они рассматриваться не будут. Тип записи — это код, определяющий запись для прямого или обратного преобразования адресов, или запись, используемая почтовым сервером. Содержимое записи может занимать одну или несколько строк и обычно представляет собой IP-адрес или имя узла. Если содержимое записи является описанием зоны, оно располагается в нескольких строках, в остальных случаях вся запись помещается в одной строке. Признаком комментариев является точка с запятой, текст, следующий за ней, не обрабатывается сервером.

ВНИМАНИЕ В различных конфигурационных файлах BIND точка с запятой интерпретируется по-разному: в файле `named.conf` ею заканчивается выражение, а в файле зоны она определяет комментарии. Это следует учитывать при редактировании конфигурационных файлов BIND, в противном случае сервер будет работать некорректно.

Конфигурационный файл зоны обычно помещается в каталог `/var/named`, и ему присваивается имя, связанное с именем зоны. Обычно для такого файла выбирается имя `db.`*имя-зоны* или `named.`*имя-зоны*. Конфигурационному файлу можно присвоить любое имя, необходимо лишь, чтобы оно совпадало с именем, указанным в `/etc/named.conf`.

Формирование описания зоны

Для описания зоны используется запись SOA (Start of Authority — начало полномочий). В поле, определяющем тип записи, указано значение SOA. Наличие этой записи означает, что сервер имен поддерживает данный домен. В поле имени указывается имя зоны, совпадающее с именем, заданным в файле `/etc/named.conf` (не забывайте о завершающей точке!). Содержимое записи в данном случае состоит из трех частей.

- **Ведущий сервер имен.** Первое имя (в листинге 18.2 это `spruce.threeroomco.com.`) представляет имя ведущего сервера имен для зоны. В листинге 18.2 за этим именем следует обратная косая черта (\). Как и во многих конфигурационных файлах, этот символ означает, что продолжение записи находится на следующей строке. Часто в конфигурационных файлах зоны две строки объединяются в одну, и в этом случае обратная косая черта не нужна.

- **Почтовый адрес администратора.** Второе имя (в листинге 18.2 это `admin.threeroomco.com.`) определяет почтовый адрес администратора, отвечающего за поддержку зоны. Этот адрес представляется в несколько необычном формате. Чтобы его можно было использовать, надо заменить первую точку на символ `@`, так, адрес `admin.threeroomco.com.` будет преобразован в `admin@threeroomco.com`.

- **Временные соотношения.** Числовые значения, помещенные в скобки и располагающиеся в последующих строках, задают информацию о временных соотношениях. В листинге 18.2 приведены комментарии, поясняющие назначение соответствующих чисел. В строке, помеченной комментариями `serial`, содержится последовательный номер, который необходимо увеличивать при каждом редактировании файла. Ведомый сервер на основании данного значения определяет, следует ли обновлять свой конфигурационный файл. Многие администраторы задают последовательный номер как дату (в формате YYYYMMDD), сопровождая ее номером изменений, произведенных в течение текущего дня. В строке `refresh` задается время в секундах между обращениями ведомого сервера к ведущему. Значение 3600, приведенное в листинге 18.2, соответствует одному часу, следовательно, ведомый сервер будет каждый час проверять, не изменились ли параметры зоны на ведущем сервере. Значение `retry` представляет время в секундах, по истечении которого ведомый сервер предпримет повторную попытку обращения к ведущему серверу в случае, если первая попытка окажется неудачной. Значение `expire` также

определяет время в секундах, в течение которого запись считается действительной, если ведомому серверу не удается связаться с ведущим. По истечении указанного времени запись не может быть использована для преобразования имен. Величина `expire` обычно соответствует одной неделе и, конечно же, должна превышать значение `refresh`. Значение `default_ttl` задает время жизни. В течение этого времени сервер DNS должен хранить информацию о результатах преобразования. Обычно величина времени жизни составляет от одного дня (86400 в листинге 18.2) до одной недели (604800). Если IP-адреса вашей сети часто изменяются или если вы предполагаете, что вскоре придется произвести много изменений, имеет смысл задать время жизни порядка часа.

Определение адресов и имен

Большинство записей в конфигурационном файле зоны предоставляет информацию о соответствии между именами и IP-адресами. В поле имени, как правило, задается имя компьютера либо другое имя, связанное с IP-адресом. В поле имени также можно задавать символ @, заменяющий имя домена. Данный символ обычно используют в записях MX и NS (листинг 18.2). Эти записи не несут информации о взаимосвязи имен и адресов, а определяют специальные имена для всего домена.

Ниже описаны основные типы записей.

- A. В записи A (address — адрес) в поле имени задается имя узла, а содержимое записи представляет собой IP-адрес. В качестве имени узла можно использовать полное доменное имя (с завершающей точкой), например `gingko.threeroomco.com.`, либо имя узла без указания имени домена, например `birch` или `spruce`. Можно также в качестве имени компьютера указать имя домена, так, например, в листинге 18.2 задано соответствие между именем `threeroomco.com.` и IP-адресом 192.168.1.4.

- `CNAME`. Запись `CNAME` (canonical name — каноническое имя) ставит в соответствие имени другое имя. В поле содержимого может быть указано либо полное доменное имя с завершающей точкой, либо имя узла без имени домена. Если вы задаете полное доменное имя, оно не обязательно должно принадлежать домену, определяемому посредством файла зоны. Например, в листинге 18.2 имя `kelp` связывается с компьютером в другом домене. Записи CNAME обычно применяются в тех случаях, когда важные IP-адреса могут изменяться без вашего участия. Например, если вы размещаете Web-страницу на внешнем компьютере, то можете связать имя `www` с именем этой машины. Если адрес внешнего компьютера изменится, ваша запись останется корректной.

- PTR. В листинге 18.2 записи типа PTR отсутствуют. Эти записи применяются для обратного преобразования и будут рассматриваться ниже.

- `NS`. Запись NS (name server — сервер имен) задает сервер имен для домена. В конфигурационном файле должна присутствовать хотя бы одна запись NS, указывающая на компьютер, заданный в качестве ведущего сервера имен в записи SOA. В поле имени данной записи указывается либо имя домена, либо символ @. IP-адрес компьютера, содержащего сервер имен, задается с помощью записи A.

- MX. Запись MX (mail exchanger — обмен почтой) предоставляет информацию о почтовом сервере для зоны. В поле имени этой записи указывается символ @ либо имя домена. В поле содержимого записи содержатся два компонента: код приоритета и имя узла. Когда удаленный почтовый сервер собирается передать сообщение пользователю в домене (например, `lorax@threeroomco.com`), он запрашивает у сервера имен записи MX. Затем уделенный сервер пытается свзязаться с компьютером, для которого указано наименьшее значение приоритета (в листинге 18.2 это `birch.threeroomco.com.`). Если этот компьютер не доступен, удаленный сервер предпринимает попытку установить соединение с тем узлом, приоритет которого выражается следующим по величине значением (в листинге 18.2 это `mail.pangaea.edu.`). Перебор компьютеров продолжается до тех пор, пока сообщение не будет доставлено либо пока не выяснится, что все узлы, указанные в записях MX, не доступны. Очевидно, что компьютер, имя которого задано в записи MX, должен быть настроен для приема почты. Вопросы передачи почтовых сообщений будут рассматриваться в главе 19.

СОВЕТ ✔ Некоторые типы записей указывают на компьютеры, расположенные за пределами домена. Так, например, вы можете задать в качестве почтового сервера узел внешней сети.

В листинге 18.2 приведены примеры многих из перечисленных выше записей. В реальном конфигурационном файле зоны содержится информация о гораздо большем количестве компьютеров.

Конфигурация зоны для обратного преобразования

В листинге 18.1 указано несколько зон, некоторые из них предназначены для обратного преобразования. Эти зоны позволяют серверу DNS определять доменное имя по IP-адресу. Для того чтобы это стало возможным, необходимо создать псевдодомен `in-addr.arpa`. В файле `/etc/named.conf` содержатся указатели на конфигурационные файлы, описывающие подмножества этого домена. Поскольку имя домена уточняется при движении справа налево, а IP-адрес уточняется по мере движения слева направо, в имени псевдодомена адрес должен быть указан в обратном порядке. Например, имя зоны для диапазона адресов 192.168.1.0/24 будет иметь вид `1.168.192.in-addr.arpa`.

Зона для обратного преобразования, или обратная зона, настраивается подобно зоне прямого преобразования. Конфигурационный файл зоны содержит записи SOA и NS, но основное место в нем занимают записи PTR. При обратном преобразовании не возникает необходимость в записях MX, A и CNAME. В листинге 18.3 содержится конфигурационный файл обратной зоны, соответствующий файлу, приведенному в листинге 18.2.

В поле имени записи PTR указывается либо сокращенный вариант адреса (например, 1 для 192.168.1.1), либо полный IP-адрес, расположенный в обратном порядке и сопровождаемый именем `in-addr.arpa`. В листинге 18.3 продемонстрированы оба подхода. В поле содержимого включается полное доменное имея с точкой в конце. Поскольку обратная зона отличается от зоны, используемой для прямого преобразования, попытка задать в поле содержимого сокращенное имя приведет к некорректному преобразованию, например, при указании `birch` вместо `birch.threeroomco.com.` будет получен результат `birch.1.168.192.in-addr.arpa`.

Листинг 18.3. Пример конфигурационного файла обратной зоны

```
1.168.192.in-addr.arpa.  IN SOA spruce.threeroomco.com. \
                             admin.threeroomco.com. (
                         2002043004 ; serial
                         3600       ; refresh
                         600        ; retry
                         604800     ; expire
                         86400      ; default_ttl
                         )
1                              IN PTR  gingko.threeroomco.com.
2.1.168.192.in-addr.arpa.      IN PTR  birch.threeroomco.com.
3.1.168.192.in-addr.arpa.      IN PTR  spruce.threeroomco.com.
4.1.168.192.in-addr.arpa.      IN PTR  threeroomco.com.
@                              IN NS   spruce.threeroomco.com.
```

Настройка сервера, предназначенного только для кэширования

В небольших сетях часто используются серверы DNS, основная задача которых — кэширование результатов преобразования имен. Сервер такого типа не поддерживает конкретный домен (за исключением домена для обратного преобразования `localhost`). Вместо этого сервер перенаправляет запросы внешним серверам DNS и записывает полученные от них сведения в кэш. Такая конфигурация сервера может ускорить работу клиент-программ, в частности, Web-броузеров, если сеть связана с Internet посредством линий с низкой пропускной способностью или большой задержкой. Так, например, линия спутниковой связи характеризуется задержкой порядка половины секунды при двусторонней передаче данных. Задержка при использовании коммутируемых линий составляет около 200 миллисекунд, что лучше, чем для линий спутниковой связи, но все же существенно замедляет преобразование имен.

Следует заметить, что время преобразования снижается только в том случае, когда необходимые данные уже содержатся в кэше сервера. Поэтому рассматриваемую здесь конфигурацию имеет смысл использовать только в сетях с относительно большим количеством пользователей, которые часто обращаются к одним и тем же узлам глобальной сети.

Основной конфигурационный файл сервера, предназначенного лишь для кэширования, имеет тот же формат, что и файл, представленный в листинге 18.1, однако в нем присутствует лишь определение зоны, предназначенной для обратного преобразования `localhost` (`0.0.127.in-addr.arpa`), и корневой зоны (`.`). Даже эти зоны не являются обязательными.

Наиболее важными компонентами конфигурационного файла сервера DNS, предназначенного только для кэширования, являются опции `forwarders` и `forward`, расположенные в разделе `options`. Опция `forwarders` должна содержать список серверов DNS провайдера. BIND использует эти серверы для выполнения преобразования. Вместо выражения `forward first`, приведенного в листинге 18.1, необходимо указать

forward only. В этом случае сервер прекратит попытки преобразования, если серверы, указанные в качестве значения forwarders, окажутся не доступными.

ВНИМАНИЕ Если вы включите в состав конфигурационного файла корневую зону и зададите опцию forward first, то в случае, когда серверы, предназначенные для перенаправления запроса, станут не доступны, BIND предпримет попытку выполнения стандартной процедуры преобразования адресов. Само по себе это неплохо, но при этом процедура выявления ошибки и генерации соответствующего сообщения окажется длительной. В особенности этот будет заметно в том случае, если соединение с Internet осуществляется посредством линии с большой задержкой.

Ранее в данной главе упоминались серверы, предназначенные лишь для кэширования результатов преобразований. Эти серверы занимают меньше памяти, чем BIND, тем не менее, они используются достаточно редко. Причина в том, что BIND поставляется в составе многих дистрибутивных пакетов и гораздо проще в настройке по сравнению с другими серверами. Если вы захотите сэкономить ресурсы, вы можете вместо BIND использовать dnscache или pdnsd.

Сконфигурировав и запустив сервер имен (либо полнофункциональный, либо предназначенный только для кэширования), вам необходимо указать его IP-адрес для всех компьютеров, которые должны работать с ним. Если вы забудете сделать это, узлы вашей сети по-прежнему будут обращаться к внешним серверам DNS.

Взаимодействие с сервером DHCP

Если в вашей сети IP-адреса распределяются посредством сервера DHCP, вы не сможете задавать фиксированные адреса в конфигурационном файле зоны, так как адрес, выделенный клиенту DHCP, становится известным лишь при его загрузке и может измениться при следующей загрузке. В главе 5 рассматривались два решения этой проблемы: настройка сервера DHCP для предоставления клиенту одного и того же IP-адреса и настройка DHCP и серверов DNS для совместной работы. Если вы выберете первый способ, т. е. сконфигурируете сервер DHCP так, что он будет выделять конкретным клиентам фиксированные IP-адреса, вам придется уделять много внимания сопровождению этих серверов. Предположим, что вам необходимо указать, что компьютеру birch.threeroomco.com должен соответствовать адрес 192.168.1.2. Для этого вам надо изменить как конфигурационный файл сервера DHCP, так и конфигурационные файлы зон сервера DNS, используемых для прямого и обратного преобразования. При этом приходится следить за тем, чтобы значения в обоих файлах совпадали. Несмотря на то что данное решение реализуется относительно просто, для больших доменов оно неприемлемо.

В главе 5 рассматривалась конфигурация сервера DHCP для совместной работы с сервером DNS. Чтобы соответствующим образом настроить BIND, вам надо внести изменения в файл named.conf. Вы должны добавить в определение соответствующей зоны опцию allow-update. В результате определение зоны примет следующий вид:

```
zone "threeroomco.com" {
    type master;
    file "named.threeroomco.com";
```

```
    allow-update { 192.168.1.1; }
};
```

Теперь BIND будет принимать информацию об IP-адресах от компьютера с адресом 192.168.1.1. Как нетрудно догадаться, это должен быть адрес узла сети, на котором выполняется сервер DHCP. Аналогичные изменения надо внести в определение обратной зоны.

ВНИМАНИЕ Если ваш сервер DNS доступен из Internet или если вы не полностью доверяете пользователям локальной сети, получение информации об обновлении DNS-данных представляет угрозу безопасности системы. Хакер может взломать сервер DHCP или, фальсифицировав адрес, внести необходимые ему изменения в конфигурацию сервера DNS. Чтобы уменьшить опасность атаки, желательно разместить серверы DNS и DHCP на одном компьютере и разрешить обновление только с локального узла (127.0.0.1).

Запуск и тестирование сервера

Для запуска сервера DNS можно применить любой из способов, рассмотренных в главе 4, но чаще всего сервер DNS запускается с помощью сценария SysV или локального сценария запуска. Запуск сервера DNS посредством суперсервера снизит скорость преобразования имен.

Для тестирования сервера имен хорошо подходит инструмент `host`. Программа `host` поставляется в составе большинства версий Linux; обычно она располагается в пакете `bind-utils`. Данная утилита передает указанному серверу DNS запросы на преобразование имен. В простейшем случае `host` взаимодействует с сервером, указанным в файле `/etc/resolv.conf` на том компьютере, на котором она выполняется. Для вызова данной программы надо ввести ее имя, а также имя узла либо IP-адрес.

```
$ host www.awl.com
www.awl.com is a nickname for awl.com
awl.com has address 165.193.123.224
```

В данном примере первая строка, отображаемая программой, сообщает о том, что `www.awl.com` представляет собой каноническое имя (заданное с помощью записи CNAME) узла awl.com. Этот компьютер имеет адрес 165.193.123.224. Такой тест подтверждает, что сервер DNS может преобразовывать имена внешних узлов. Если вы сконфигурировали сервер для поддержки собственного домена, вы должны указать при проверке локальные имя и адрес. Убедитесь в том, что сервер выполняет как прямое, так и обратное преобразование. При необходимости вы также можете просмотреть записи требуемого типа, задавая при вызове утилиты опцию `-t`. Например, чтобы проверить записи MX для домена, вам потребуется выполнить следующую команду:

```
$ host -t MX awl.com
awl.com mail is handled by 100 mailhost.uu.net.
awl.com mail is handled by 10 oldtms702.pearsontc.com.
awl.com mail is handled by 20 oldtms701.pearsontc.com.
```

Данная проверка показывает, что в домене `awl.com` определены три почтовых сервера: `oldtms702.pearsontc.com` (приоритет 10), `oldtms701.pearsontc.com` (при-

оритет 20) и **mailhost.uu.net** (приоритет 100). Если вы хотите указать сервер DNS для тестирования, вам надо при вызове программы задать имя или IP-адрес этого сервера.

```
$ host www.awl.com spruce
Using domain server:
Name: spruce.threeroomco.com
Address: 192.168.1.3
Aliases:

www.awl.com is a nickname for awl.com
awl.com has address 165.193.123.224
```

При этом выводятся те же данные, что и в предыдущем примере, кроме того, отображается подтверждение того, что программа **host** передала запрос конкретному серверу DNS.

Дополнительную информацию о программе **host** вы можете найти в справочной системе. Для тестирования сервера также можно использовать утилиту **nslookup**. Она выполняет практически те же функции, что и **host**, но в настоящее время считается устаревшей.

Резюме

Если в сети имеется достаточно большое число компьютеров, целесообразно установить в этой сети сервер DNS. Этот сервер может обслуживать как внешних клиентов, которым необходимо определять IP-адреса компьютеров, принадлежащих вашему домену, так и внутренних клиентов, которым нужны IP-адреса внешних компьютеров. При настройке сервера DNS надо задать базовый набор опций, включая опции общего назначения, информацию о корневых серверах и определения зон, поддерживаемых вашим сервером. Если вы хотите, чтобы сервер выполнял преобразования имен компьютеров вашего домена, вам надо создать конфигурационный файл зоны. В этом файле содержится информация о соответствии имен и IP-адресов, а также указываются компьютеры, выполняющие специальные функции в домене, например почтовые серверы.

Глава 19

Передача почты: протокол SMTP

В главе 11 рассматривались протоколы POP и IMAP, которые частично решают задачу доставки писем. Эти и другие протоколы получения почты позволяют пользователю копировать адресованные ему сообщения с центрального сервера на свой компьютер. Для того чтобы обеспечить доставку писем, надо решить еще две задачи: копирование сообщений на центральный почтовый сервер и передачу их адресатам. Обе эти задачи решаются посредством *протоколов передачи почты*. При использовании этих протоколов сообщения передаются по инициативе сервера. Взаимосвязь между протоколами передачи и получения почты рассматривалась в главе 11.

Из протоколов передачи почты в настоящее время наиболее популярным является SMTP (Simple Mail Transfer Protocol — простой протокол передачи почты). Все письма, передаваемые в среде Internet, проходят как минимум через один сервер SMTP. Обычно серверы получения и передачи почты выполняются на одном и том же компьютере, так что письмо, полученное сервером SMTP, может быть в любой момент передано на пользовательский компьютер. Если в системе Linux предполагается обработка почты, сервер SMTP выполняет при этом основную роль. В состав каждого дистрибутивного пакета Linux входит по меньшей мене один сервер SMTP, однако различные системы содержат разные серверы. В данной главе рассматриваются наиболее популярные в настоящее время серверы SMTP: `sendmail`, Exim и Postfix. Кроме того, в этой главе пойдет речь об инструменте Procmail, часто используемом совместно с сервером SMTP. Прежде чем вплотную приступить к настройке сервера SMTP, необходимо рассмотреть доступные программы, поддерживающие функционирование подобного сервера, и вопросы настройки домена для работы сервера.

Часто для обеспечения работы сервера в сети достаточно внести лишь минимальные изменения в состав его конфигурационного файла, но в некоторых случаях приходится выполнять сложные действия по настройке сервера. Для тех, кому необходима дополнительная информация о серверах SMTP, можно порекомендовать следующие книги: Косталес (Costales) и Эллмен (Allman) *Sendmail* (O'Reilly, 1997), Хант (Hunt) *Linux Sendmail Administration* (Sybex, 2001), Хэзел (Hazel) *Exim: The Mail Transfer Agent* (O'Reilly, 2001),

Блюм (Blum) *Postfix* (Sams, 2001), Стилл (Sill) *The qmail Handbook* (APress, 2001) и Маккарти (McCarthy) *The Procmail Companion* (Addison Wesley, 2001).

Использование сервера SMTP

Серверы SMTP часто называют агентами передачи почты (MTA — mail transfer agent). В процессе обмена почтой подобная программа может выступать как в роли клиента, так и в роли сервера. Сервер SMTP может получить сообщение, переданное другим сервером, поддерживающим этот протокол. Сервер хранит полученные сообщения на локальном компьютере, а при необходимости может передать их другому серверу SMTP. Возможные варианты использования сервера SMTP в системе Linux описаны ниже.

- Получение почты. Чаще всего сервер SMTP выполняет функции центрального почтового сервера сети. Письма, полученные сервером SMTP, могут быть просмотрены с помощью специальных программ (например, `pine` или `mutt`), выполняющихся на локальном компьютере, либо скопированы на другой узел сети посредством протокола получения почты.

- Передача писем в режиме ретранслятора. У пользователей локальной сети часто возникает необходимость передать сообщение другому пользователю, работающему в Internet. Сервер SMTP может быть сконфигурирован для перенаправления почты. В частности, он может получать письма из локальной сети, временно хранить их на локальном компьютере, а затем передавать удаленным системам. Такая конфигурация очень важна для обеспечения передачи почты в сети, и в то же время она может стать источником проблем. Вопросы перенаправления писем будут более детально рассмотрены далее в этой главе.

- Организация почты на локальном компьютере. Программы, выполняющиеся на том же компьютере, что и почтовый сервер, обращаются к серверу для передачи почты. Подобные программы могут быть также сконфигурированы для работы с удаленным сервером SMTP. Многие из программ, работающие на локальном компьютере, используют почтовый сервер для взаимодействия с пользователями. Так, например, программы, автоматизирующие выполнение задач администрирования, посылают по почте отчеты о своих действиях пользователю `root`.

Выполнение перечисленных выше функций имеет настолько большое значение для системы, что при инсталляции многих дистрибутивных пакетов Linux сервер SMTP устанавливается по умолчанию. В частности, данный сервер необходим для организации обмена почтой в пределах локального компьютера. (Теоретически вы можете настроить все программы так, чтобы они работали посредством удаленного сервера, но инсталлировать сервер SMTP на локальном компьютере гораздо проще, чем перенастраивать многочисленные программы.)

Во многих дистрибутивных пакетах конфигурация сервера SMTP, установленная по умолчанию, обеспечивает передачу почты в пределах локального узла сети. Чтобы сервер выполнял другие функции, в частности мог поддерживать весь домен, его конфигурацию в большинстве случаев приходится пересматривать. В настоящее время электронная почта стала настолько важной службой, что ни одна локальная сеть не может обойтись без средств доставки писем. Однако установка собственного сервера — не единственный

способ поддержки обмена сообщениями. При необходимости вы можете также воспользоваться услугами внешних серверов. Такое решение может быть приемлемым для небольшой сети в случае, если вы посчитаете, что затраты на установку центрального почтового сервера не оправдывают себя. Однако наличие собственного сервера обеспечивает дополнительные возможности настройки средств доставки почты и позволяет контролировать процесс передачи писем. Так, например, при наличии сервера SMTP в своей сети вы можете быстро добавлять или удалять пользователей, блокировать получение нежелательных сообщений, устанавливать ограничения на размер сообщений и выполнять другие подобные действия. На поддержку собственного сервера потребуется затратить меньше средств, чем на оплату почтовых услуг сторонней организации. Таким образом, решение по установке почтового сервера в локальной сети в большинстве случаев оправдано, особенно если число пользователей сети велико и им требуются специальные услуги.

Программы, реализующие сервер SMTP в системе Linux

- `sendmail`. В составе системы Linux часто поставляется наиболее популярный в настоящее время почтовый сервер sendmail. Этот пакет предоставляет обширные возможности и многие программы по умолчанию считают, что он установлен в системе. Для обеспечения совместимости в состав некоторых пакетов даже включается исполняемая программа sendmail. Конфигурационный файл sendmail имеет сложный формат, и это является причиной того, что некоторые пользователи отдают предпочтение альтернативным пакетам. Web-узел sendmail расположен по адресу `http://www.sendmail.org`.

- **Exim.** Формат конфигурационного файла данного сервера проще, чем у sendmail, кроме того, Exim поддерживает разнообразные правила фильтрации почты. Этот сервер используется в Debian и системах, созданных на ее основе. Адрес Web-узла Exim — `http://www.exim.org`.

- **Postfix.** Как sendmail, так и Exim реализованы в виде большой "монолитной" программы. В отличие от этих продуктов, Postfix имеет модульную структуру. Это означает, что частные задачи, возникающие перед почтовым сервером, решаются с помощью отдельных небольших программ. При этом повышается как производительность сервера, так и уровень безопасности системы. Модульная структура и простота конфигурационного файла являются основными преимуществами Postfix по сравнению с sendmail. Данный сервер используется в качестве сервера по умолчанию в системе Mandrake. Дополнительную информацию о Postfix можно получить, обратившись по адресу `http://www.postfix.org`.

- `qmail`. Подобно Postfix,`qmail` представляет собой модульный сервер, разработчики которого ставили перед собой задачу обеспечить высокую производительность и повышенный уровень защиты. Структура конфигурационного файла qmail проще, чем у свервера sendmail, но, в отличие от Exim и Postfix, данный сервер плохо совместим с sendmail. Поэтому замена sendmail на qmail представляет собой достаточно сложную задачу. Несмотря на то что qmail по популярности уступает только sendmail, этот сервер редко включается в дистрибутивные пакеты Linux

в качестве сервера по умолчанию, поэтому в данной главе он не будет подробно рассматриваться. Web-узел qmail расположен по адресу http://www.qmail.org.

Помимо перечисленных выше, в системе Linux могут использоваться и другие почтовые серверы. В качестве примера можно привести Smail (http://www.gnu.org/software/smail/smail.html), Courier (http://www.courier-mta.org) и OpenMail (http://www.openmail.com/cyc/om/00/). Многие из почтовых серверов распространяются в исходных кодах, но некоторые доступны лишь на коммерческой основе. Большинство пользователей отдают предпочтение упомянутым выше четырем серверам: sendmail, Exim, Postfix и qmail. Все четыре продукта представляют собой мощные программы, способные обслуживать даже большие домены.

Если вы еще не имеете большого опыта администрирования почтовых серверов, вам предпочтительнее использовать тот из них, который поставляется в составе вашей системы. Во многих дистрибутивных пакетах содержится несколько серверов SMTP. В этом случае лучше работать с сервером, установленным по умолчанию.

Если вам необходимо предоставить пользователям специальные услуги по обработке почты, внимательно ознакомьтесь с документацией на различные серверы и решите, какой из них наилучшим образом подходит для решения данной задачи. Возможно, вам придется заменить сервер, установленный по умолчанию, другим. В большинстве случаев это означает, что вместо сервера sendmail необходимо установить другой сервер. Проще всего заменить sendmail сервером Exim или Postfix. Несмотря на различия в структуре конфигурационных файлов, программы, непосредственно обращающиеся к sendmail, обычно хорошо взаимодействуют с Exim и Postfix, а формат очереди почтовых сообщений этих двух программ совпадает с форматом очереди sendmail. (Как и sendmail, Exim и Postfix используют формат mbox, т. е. хранят все письма в одном файле.) Заменить sendmail сервером qmail гораздо труднее, так как qmail по умолчанию поддерживает maildir (формат, в котором сообщения хранятся как отдельные файлы). Поэтому, чтобы установить qmail вместо sendmail, надо изменить стандартную конфигурацию qmail или заменить почтовые программы в вашей системе (в том числе и серверы получения почты, рассмотренные в главе 11).

Настройка домена для использования почтового сервера

Многие почтовые серверы получают почту с внешних компьютеров. Существуют два способа адресации почтового сервера.

- Непосредственная адресация. Письмо может быть направлено пользователю, учетная запись которого находится на почтовом сервере. Например, если почтовый сервер имеет имя mail.threeroomco.com, то почтовый адрес пользователя будет выглядеть так: jennie@mail.threeroomco.com. В этом случае для сервера имен потребуется только запись A, связывающая имя почтового сервера с его адресом. Недостаток подобного способа состоит в том, что адрес получается несколько длиннее, чем он мог бы быть.

- Указание адреса домена. Для того чтобы сократить почтовый адрес, а также для того, чтобы обеспечить работу резервных почтовых серверов, в конфигурацион-

ном файле сервера DNS предусмотрена запись MX. Если в почтовом адресе указано только имя домена, запись MX позволят направить это письмо на конкретный компьютер. Предположим, например, что в конфигурационном файле сервера имен, управляющего доменом threeroomco.com, содержится запись MX, указывающая на компьютер mail.threeroomco.com. В этом случае письмо, адресованное пользователю jennie@threeroomco.com, будет доставлено на узел mail.threeroomco.com. Указав в конфигурационном файле сервера DNS несколько записей MX, администратор может организовать использование резервных почтовых серверов. По сравнению с непосредственной адресацией данный способ несколько усложняет администрирование домена.

Формат конфигурационного файла сервера DNS, в частности структура записей MX, рассматривалась в главе 18. Если вы устанавливаете почтовый сервер по адресу mail.threeroomco.com, соответствующая запись MX будет выглядеть следующим образом:

```
@              IN      MX      10      mail.threeroomco.com.
```

Эта строка содержится в конфигурационном файле зоны, имя которого указано в файле /var/named. Символ @ в начале строки означает, что запись применима ко всему домену. Идентификатор IN представляет собой стандартный компонент записи, описывающей домен Internet, а MX определяет тип записи. Число 10 представляет приоритет записи. Сервер, осуществляющий передачу почты, сначала старается установить соединение с почтовыми серверами, которым соответствуют малые значения приоритетов, а если очередной сервер не отвечает, он обращается к компьютеру с более высоким приоритетом. Это позволяет использовать в составе домена несколько почтовых серверов. В конце записи указывается полное доменное имя почтового сервера, завершающееся точкой.

Передавая письма на почтовый сервер вашей сети, внешним пользователям ДОстаточно указать лишь имя домена. Однако локальные пользователи при настройке своих клиентских программ должны задавать полное имя сервера SMTP.

Передача данных с помощью протокола SMTP

Для того чтобы понять материал данной главы, надо хотя бы в общих чертах представлять себе принцип передачи почтовых сообщений с помощью протокола SMTP. В частности, необходимо знать различия между *заголовками конверта* (envelope header), *заголовками сообщения* (message header) и *телом сообщения* (message data). Заголовком конверта считаются поля From и To и содержащиеся в них адреса, которые указываются передающим компьютером при установлении SMTP-соединения. В особенности важен заголовок конверта To, так как именно его анализирует принимающая система, определяя, кому адресовано данное сообщение.

В отличие от заголовка конверта, заголовок сообщения входит в состав письма. Нередко этот заголовок составляет значительную часть сообщения, доставляемого адресату. Среди них также присутствуют поля From: и To:, но полагаться на их значения нельзя, так как они могут быть фальсифицированы. К заголовку сообщения относятся также поле Received:, которое отражает путь, проделанный письмом, и поле Subject:, отображаемое большинством программ просмотра писем.

В заголовке **сообщения** значение поля отделяется от его имени двоеточием. В заголовке конверта двоеточие обычно не используется. Если сервер SMTP использует формат **maildir,** данные, содержащиеся в заголовке конверта, хотя и используются при выполнении **SMTP-транзакции,** но не сохраняются в сообщении. Некоторые серверы могут быть сконфигурированы так, чтобы адреса, указанные в полях From и To заголовка **конверта,** сохранялись в составе поля **Received:** заголовка сообщения. Это помогает в тех случаях, когда надо выяснить причину возникновения проблем при передаче писем.

Чтобы понять, как действует протокол SMTP, надо проследить ход SMTP-транзакции. В листинге 19.1 показан процесс передачи сообщения, осуществляемый вручную с помощью программы **telnet.** (Как нетрудно догадаться, в обычной SMTP-транзакции программа **telnet** не участвует).

Листинг 19.1. Пример SMTP-сеанса

```
$ telnet louiswu.rodsbooks.com 25
Trying 192.168.1.5...
Connected to louiswu.rodsbooks.com.
Escape character is '^]'.
220 louiswu ESMTP Exim 3.12 #1 Wed, 30 Oct 2002 12:01:29 -0500
HELO nessus.rodsbooks.com
250 louiswu Hello nessus.rodsbooks.com [192.168.1.3]
MAIL FROM:<rodsmith@nessus.rodsbooks.com>
250 <rodsmith@nessus.rodsbooks.com> is syntactically correct
RCPT TO:<rodsmith@louiswu.rodsbooks.com>
250 <rodsmith@louiswu.rodsbooks.com> is syntactically correct
DATA
354 Enter message, ending with "." on a line by itself
From: <rodsmith@nessus.rodsbooks.com>
To: <rodsmith@louiswu.rodsbooks.com>
Subject: A Sample SMTP Session

This is the text of the message.
.
250 OK id=15z87H-OOOOCX-00
QUIT
221 louiswu closing connection
Connection closed by foreign host.
```

В большинстве случаев SMTP-соединение начинается по инициативе клиентской программы, в роли которой выступает программа подготовки писем или другой сервер SMTP (в сеансе, представленном в листинге 19.1, действия клиентской программы имитирует пользователь с помощью программы **telnet).** Клиент объявляет о своем намерении начать взаимодействие, передавая серверу команду HELO или EHLO. Затем с помощью команд MAIL FROM: и RCPT TO: клиент задает заголовки конверта From и To. В ответ на каждую из этих команд сервер SMTP передает числовой код, посредством которого

он сообщает результаты обработки очередной команды. Текст, следующий за числовым кодом, предназначен для пользователя, который следит за ходом взаимодействия. Команда DATA указывает на то, что передающий компьютер готов пересылать тело сообщения. Получив ответ от сервера, клиент начинает передачу заголовков и данных сообщения. (Заголовки сообщения передавать не обязательно. Если исключить их из листинга 19.1, письмо все равно будет доставлено.) Заголовки отделяются от тела сообщения пустой строкой. Строка, содержащая только точку, является признаком окончания сообщения.

Ниже описаны некоторые характерные особенности SMTP-транзакции и передаваемых сообщений, которые должны учитываться при настройке сервера.

- **Идентификация отправителя.** Передающая система передает свой адрес принимающему компьютеру различными способами, в частности, адрес содержится в составе команд HELO и MAIL FROM, а также поля заголовка `From:`. Следует заметить, что если передающая система работает в режиме ретранслятора, в команде MAIL FROM и в поле заголовка `From:` будет содержаться адрес другого компьютера. Так или иначе, IP-адрес отправителя становится известен принимающей системе; в листинге 19.1 этот адрес указывается в ответе на команду HELO.

- **Заголовки конверта и сообщения.** В листинге 19.1 заголовки конверта и сообщения соответствуют друг другу, но в других случаях ситуация может быть иной. Если вы получите сообщение, адресованное не вам, причина этого может состоять в том, что в поле To заголовка конверта был указан ваш адрес, а в поле `To:` заголовка сообщения — адрес другого пользователя. Поскольку действия по доставке письма определяются значением поля To заголовка конверта, то такое письмо будет передано вам. Почтовая программа, сохраняющая все данные о заголовках, помогла бы вам выяснить причины происходящего.

- **Возможность отказаться от сообщения.** Принимающий сервер SMTP может прекратить работу по доставке письма на любом этапе, начиная с установки соединения и заканчивая обработки сообщения после его получения. Чаще всего управление доставкой сообщения осуществляется после ответа на команду RCPT `TO:`, но существует также возможность контроля на более ранних этапах взаимодействия. Если получатель прерывает соединение перед получением команды RCPT `TO:`, некоторые отправители предпринимают повторную попытку передачи письма, что увеличивают нагрузку на сеть. Отказ от обработки после передачи содержимого письма имеет свой недостаток — если объем сообщения велик, такой подход также приведет к неоправданному увеличению трафика в сети.

- **Предоставление информации о сервере.** При проведении сеанса, приведенного в листинге 19.1, отправитель получает лишь частичную информацию о сервере и никаких других данных. В данном примере сервер объявляет себя как Exim 3.12. Некоторые программы позволяют скрыть номер версии; если в средствах защиты сервера имеются недостатки, такая мера повышает безопасность системы. Сохраняя в секрете номер реализации сервера, вы лишаете хакера информации о возможных путях незаконного проникновения в систему. В листинге 19.1 в ответ на команды MAIL `FROM:` и RCPT `TO:` принимающий сервер возвращает код 250 и сообщение `is syntactically correct`. Некоторые серверы могут быть сконфигурированы так, что в случае, если в системе отсутствует учетная запись пользователя,

которому адресовано письмо, взаимодействие с передающим компьютером прекратится после обработки команды RCPT TO:. Такая конфигурация предоставляет информацию, полезную для злоумышленника: посредством команд почтового сервера он может подобрать корректное имя пользователя. В этом примере сервер Exim не предоставляет подобных данных, однако это порождает другую проблему. Если в письме указан несуществующий пользователь, сервер должен обработать, а затем проигнорировать письмо.

Специальные функции сервера SMTP

В последующих разделах описываются различные характеристики почтового сервера, которые задаются при его настройке. Чтобы не описывать эти характеристики для каждого сервера, рассмотрим их здесь.

Маскировка адреса

При настройке почтового сервера нередко задается маскировка адреса. Согласно исходной конфигурации, устанавливаемой при инсталляции сервера SMTP, он сообщает другим серверам имя узла, которое было задано при настройке сети и возвращается по команде hostname. Это имя сервер указывает при передаче команд HELO и MAIL FROM:, оно же включается в поле From: заголовка сообщения. Кроме того, сервер сообщает имя узла, передавая другим программам ответы на полученные от них команды.

В большинстве случаев такая конфигурация устраивает системного администратора, но иногда возникает необходимость использовать для сервера другое имя. Предположим, например, что ваш почтовый сервер расположен на компьютере с именем franklin.threeroomco.com. Возможно, вы захотите, чтобы сообщения выглядели как отправленные не с конкретного компьютера, а из домена threeroomco.com. (Это может понадобиться в том случае, если вы используете для отправки и получения почты различные серверы, или для того, чтобы упростить перенос почтовых серверов на другие компьютеры.) Не исключено, что вы захотите изменить имя узла в сети, защищенной брандмауэром, чтобы сообщения, переданные в ответ, адресовались на компьютер, доступный извне. Сделать это можно, используя средства маскировки адресов. В результате их применения сервер, расположенный на компьютере franklin.threeroomco.com, будет объявлять себя как threeroomco.com. Маскировку адресов поддерживают все серверы, рассматриваемые в данной главе, но детали настройки различаются в разных продуктах. Некоторые серверы предоставляют в распоряжение администратора несколько опций, позволяющих настраивать команды, приветственные сообщения и заголовки для работы с другим именем, а в других серверах настройка осуществляется с помощью одной опции. В одних серверах относительно просто изменить заголовки существующих сообщений для отображения нового адреса, а при работе с другими решить данную задачу достаточно сложно.

Некоторые администраторы считают изменение адресов недопустимым, так как по их мнению заголовки для того и существуют, чтобы дать возможность проследить путь к исходной системе. Другие утверждают, что данное средство имеет право на жизнь, так как оно позволят устранить проблемы, возникающие, например, при передаче сообщений в сеть, защищенную брандмауэром. Если вы еще не имеете достаточного опыта в администрировании почтовых серверов, вам не следует увлекаться маскировкой адресов. Если настройка компьютера будет выполнена неправильно, это приведет к некорректной обработке сообщений.

Обработка локальных сообщений

Одна из проблем, возникающих при работе почтового сервера, состоит в том, что вам необходимо сообщать ему, какие адреса должны рассматриваться как локальные. Предположим, что почтовый сервер выполняется на компьютере `franklin.threeroomco.com`. По умолчанию сервер SMTP настраивается для получения почты, адресованной пользователям компьютера `franklin.threeroomco.com`. Если при настройке домена вы указали запись MX, которая указывает на этот компьютер, он будет обрабатывать письма для пользователей домена `threeroomco.com`. Кроме того, вам, возможно, понадобится настроить почтовый сервер для обработки почты, направленной на другие компьютеры и в другие домены. В этом случае вам придется добавить к списку доменов, в которых сервер должен поддерживать передачу локальной почты, домен `fourroomco.com`.

Все почтовые серверы, рассматриваемые в данной главе, позволяют указать, какие имена узлов сервер должен интерпретировать как локальные. Когда вы составите подобный список, сервер будет принимать письма для пользователей и распределять их по локальным почтовым ящикам (или, если сервер настроен соответствующим образом, перенаправлять письма другим системам). Особенности настройки различаются для разных серверов.

Ретрансляция писем

Наиболее сложные действия по настройке почтового сервера выполняются в том случае, когда необходимо сконфигурировать его для передачи сообщений в режиме ретранслятора. Кроме того, в этом случае может возникнуть угроза безопасности системы. В режиме ретранслятора почтовый сервер получает письмо с одного компьютера и доставляет его на другой узел сети. Ретрансляторы используются во многих сетях, в особенности в тех, в которых линии связи отличаются низкой надежностью. Если клиент пытается непосредственно доставить письмо получателю и соединение внезапно разрывается, клиент должен повторять попытки передачи письма. Если сервер SMTP сконфигурирован как ретранслятор, он может получать сообщения посредством надежных соединений и хранить их на локальном диске до тех пор, пока состояние сети не позволит передать письма по назначению.

Настраивая сервер SMTP в качестве ретранслятора, следует помнить, что подобная конфигурация представляет опасность для системы. Если почтовый сервер сконфигурирован так, чтобы принимать со многих компьютеров письма для перенаправления, он легко может быть использован для рассылки спама. Таким образом, вам необходимо найти компромисс между требованиями к защите системы и потребностями в использовании

сервера в качестве ретранслятора. Конкретные установки зависят от роли, которую ваш сервер играет в сети, а эта роль, в свою очередь, частично зависит от структуры сети.

Иногда возникает необходимость настроить сервер так, чтобы сторонние пользователи могли передавать через него свои сообщения. Как правило, такая конфигурация нежелательна, но возможны ситуации, когда у вас попросту не будет другого выхода. Например, руководство может потребовать от вас предоставить доступ к серверу пользователям, работающим с портативными компьютерами. Серверы, которые используются для обработки сообщений, передаваемых с локальных компьютеров, можно настроить для предоставления доступа удаленным пользователям.

Пользователи, которые работают с портативными компьютерами и часто перемещаются с места на место, регистрируясь на серверах различных провайдеров, создают множество проблем для системных администраторов. Предоставлять таким пользователям доступ к почтовому серверу опасно, так как каждый, кто зарегистрировался у того же провайдера, сможет использовать ваш почтовый сервер для пересылки спама. Гораздо лучше, если пользователь, работающий с потративным компьютером, будет передавать почту через сервер того провайдера, в сети которого он зарегистрировался в настоящий момент. Если же передачу через сервер провайдера по каким-то причинам реализовать не удается, вы можете создать на своем сервере конфигурацию под названием SMTP после POP. В этом случае сервер POP сообщает серверу SMTP о том, что тот может принимать письма с определенного IP-адреса в течение ограниченного периода времени. Чтобы сервер POP сгенерировал такое сообщение, пользователь должен успешно зарегистрироваться на нем и получить свою корреспонденцию. Еще одно решение состоит в использовании SSH для регистрации в локальной системе или в настройке соответствующих средств для туннелирования SMTP-соединений.

Не исключено, что вы захотите сконфигурировать сервер SMTP для передачи всех или отдельных сообщений через другой сервер SMTP. Например, несмотря на то, что почтовые серверы рабочих станций могут непосредственно передать письма по назначению, их часто настраивают так, чтобы они передавали почту через сервер отдела. Если система подключена по коммутируемой линии, использование сервера-ретранслятора остается единственным возможным решением. Многие провайдеры заносят адреса, которые выделяются компьютерам, подключаемым через выделенные линии, в специальный список, предназначенный для борьбы со спамом, в результате чего исключается возможность передачи писем, минуя сервер SMTP провайдера. Часто благодаря использованию сервера в режиме ретранслятора удается повысить надежность передачи почты в сети.

Если вы сконфигурировали операционную систему для работы в сетях различных провайдеров, передача почты может быть затруднена. Большинство провайдеров настраивают почтовые серверы так, чтобы они могли перенаправлять только письма, полученные из своей сети. Таким образом, если вы настроите свой сервер SMTP для передачи сообщений почтовому серверу одного из провайдеров, серверы остальных провайдеров останутся для вас недоступными. Чтобы решить эту проблему, надо сформировать несколько конфигурационных файлов сервера SMTP (по одному для каждого провайдера) и вклю-

чить в сценарий установки PPP-соединения команду, которая будет выполнять копирование требуемого файла в стандартный конфигурационный файл и перезапускать сервер. Предположим, что вы работаете с двумя провайдерами и используете `sendmail`. В этом случае вы должны создать два конфигурационных файла, сохранив их, например, под именами `sendmail-isp1.cf` и `sendmailisp2.cf`. В сценарий, используемый для установки PPP-соединения, вы должны включить команду копирования требуемого файла в файл `sendmail.cf`.

Настройка сервера для борьбы со спамом

Сразу после своего появления электронная почта стала широко применяться для обмена информацией между пользователями и до сих пор остается одним из наиболее популярных средств сетевого взаимодействия. К сожалению, недобросовестные рекламодатели также оценили E-mail как чрезвычайно удобное и недорогое средство, позволяющее доставлять рекламные сообщения потенциальным покупателям. Слово "к сожалению" приводится здесь по двум причинам. Во-первых, по электронной почте в настоящее время в основном распространяется реклама чрезвычайно низкого уровня: приглашения посетить порнографические Web-узлы, приобрести продукты сомнительного качества и т. д. Во-вторых, подобные сообщения создают дополнительную нагрузку на почтовые серверы во всем мире, поэтому вред, наносимый спамом, нельзя недооценивать. Передача почтового сообщения стоит недорого; гораздо дороже обходится его получение, так как письмо занимает место на диске, получатель тратит время, чтобы прочитать или удалить его, и т. д. Если механизм спама возьмут на вооружение большинство бизнесменов, электронная почта станет практически бесполезной: нужные письма потеряются в потоке рекламных сообщений.

В настоящее время подавляющее большинство системных администраторов прилагают большие усилия для того, чтобы блокировать поступление спама на свои серверы. Соответствующие действия можно разделить на две категории: фильтрация спама при его поступлении на сервер и удаление рекламных сообщений при передаче почты с сервера.

Блокирование поступающих рекламных сообщений

Вас, как системного администратора, больше всего должно заботить, чтобы спам не проник в вашу сеть. Для контроля за спамом разработаны различные способы. Наиболее популярные из них описаны ниже.

- **Сравнение входящих сообщений с шаблоном.** Большинство почтовых серверов предоставляют возможность отвергать письма на основании определенных критериев, например, отказываться от сообщений, направленных от некоторых пользователей или из определенных сетей. Эти средства можно использовать для блокирования сообщений от известных спамеров либо от провайдеров, которые не генерируют никакой информации, кроме спама. Принимая подобные меры, необходимо помнить, что если вы запретите получение сообщений от определенного провайдера, к вам также не будут приходить письма от обычных пользователей, работающих в его сети.

- **Списки IP-адресов.** В настоящее время некоторые организации публикуют списки IP-адресов, которые можно использовать для блокирования нежелательных почто-

вых сообщений. В подобном списке содержатся адреса спамеров, открытых ретрансляторов (т. е. почтовых серверов, предоставляющих всем желающим возможность передавать через них письма), компьютеров, подвергшихся взлому и используемых для распространения спама, и т. д. Многие почтовые серверы настроены для работы с одним из таких списков. Письма, переданные с адресов, указанных в списке, отвергаются. Информация о некоторых из списков IP-адресов, применяемых для борьбы со спамом, приведены в табл. 19.1.

- **Сравнение с шаблоном сообщений, обработанных почтовым сервером.** Система Procmail, которая будет описываться далее в этой главе, может быть использована для проверки писем на соответствие заданным шаблонам. На базе Procmail построены сложные системы, предназначенные для борьбы со спамом, например SpamBouncer (`http://www.spambouncer.org`). При необходимости вы можете самостоятельно сформировать фильтры Procmail.

- **Сравнение с распределенными шаблонами.** Одним из последних инструментов, предназначенных для борьбы со спамом, является Vipul's Razor (`http://razor.sourceforge.net`). Эта система использует базу данных SHA-кодов (Secure Hash Algorithm — защищенный алгоритм хеширования) сообщений спама. Вы можете сконфигурировать свой почтовый сервер для вычисления SHA-кодов полученных писем и блокировать рекламные сообщения на основании сравнения их с кодами Vipul's Razor.

Для блокирования большей части спама достаточно использовать один или два способа, например, один из списков IP-адресов и сравнение с шаблонами сообщений, обработанных почтовым сервером. Основная проблема борьбы со спамом состоит в том, что в настоящее время не существует технологии, позволяющей гарантированно распознать спам и отделить его от обычных сообщений. Большинство способов блокирования основаны на тех или иных обобщениях. Эти способы позволяют отвергать сообщения спама, но вместе с ними блокируются также некоторые корректные сообщения. Этот эффект принято называть *ложными срабатываниями* (false positive). Иногда с потерей некоторых писем можно смириться, в других случаях ложные срабатывания не допустимы. Для того чтобы уменьшить риск блокирования корректных сообщений, необходимо выполнять проверку на базе четко определенных критериев. Среди списков IP-адресов наименьшее число ложных срабатываний обеспечивают RBL и RSS. Списки IP-адресов RBL, RSS и DUL, поддерживаемые MAPS (Mail Abuse Prevention System — система борьбы с захватом почтовых программ), распространяются по подписке.

Как предотвратить использование вашего сервера для передачи спама

Помимо мер, принимаемых для блокирования рекламных сообщений, направленных вашим пользователям, необходимо также следить за тем, чтобы ваша система не использовалась для распространения спама. В больших сетях необходимо предусмотреть в сетевой политике санкции против пользователей, занимающихся рассылкой спама. Соответствующий документ следует довести до сведения всех пользователей сети. Возможно, некоторые из сотрудников организации не осознают опасности, связанной с распространением по почте рекламных сообщений.

Таблица 19.1. Списки IP-адресов, используемые для борьбы со спамом

Название списка	URL	Адрес сервера	Описание
Dial-Up List (DUL)	`http://mail-abuse.org/dul/`	`dialups.mail-abuse.org`	В данный список помещаются IP-адреса, выделяемые провайдерами для поддержки PPP-соединений. Считается, что, зарегистрировавшись в сети провайдера, пользователь не должен применять собственный почтовый сервер, так как он может воспользоваться сервером провайдера. Попытки передать письма, минуя почтовый сервер провайдера, часто предпринимают распространители спама
Realtime Blackhole List (RBL)	`http://mail-abuse.org/rbl/`	`blackholes.mail-abuse.org`	В этот список включаются адреса серверов, замеченных в распространении спама, и серверов, конфигурация которых позволяет спамерам воспользоваться их услугами
Relay Spam Stopper (RSS)	`http://mail-abuse.org/rss/`	`relays.mail-abuse.org`	В данном списке содержатся адреса серверов, замеченных в распространении спама, а также серверов, работающих как открытые ретрансляторы. Проверка таких серверов выполняется в полуавтоматическом режиме
Open Relay Database (ORDB)	`http://www.ordb.org`	`relays.ordb.org`	Данный список выполняет те же функции, что и RSS, но для принятия решения о занесении в этот список адреса сервера используются менее строгие критерии. В результате при использовании этого списка оказываются заблокированными многие корректные сообщения
RFC Ignorant	`http://www.rfc-ignorant.org`	Адреса указаны на Web-узле	Организация RFC Ignorant поддерживает несколько списков IP-адресов серверов, которые не соответствуют требованиям стандартов, изложенных в документах RFC. Спамеры часто используют некорректно настроенные программы, поэтому данные списки могут быть использованы для борьбы со спамом

Если на отдельных рабочих станциях в сети установлены почтовые серверы, необходимо следить за тем, чтобы они не были сконфигурированы как *открытые ретрансляторы* (open relay). Открытым ретранслятором называется почтовый сервер, который принимает сообщения от любого компьютера, подключенного к Internet, и передает их по назначению. Многие дискуссии, посвященные использованию sendmail, по сути сводились к вопросу о предоставлении другим компьютерам доступа к серверу. Настраивая почтовый сервер, необходимо открыть его для некоторых систем, но при этом надо следить за тем, чтобы число таких систем не оказалось слишком велико.

Чтобы поверить, не является ли почтовый сервер открытым ретранслятором, следует с компьютера, на котором расположен этот сервер, обратиться с помощью telnet по адресу relay-test.mail-abuse.org. В результате удаленный компьютер обратится к вашему почтовому серверу и начнет проверку. Ход тестирования отобразится на экране, а в конце будет выведено сообщение о состоянии вашего сервера. Следует заметить, что этот тест не является исчерпывающим; существует конфигурация, не идеальная с точки зрения безопасности, не выявляемая при проверке.

Дополнительную информацию о конфигурациях, предназначенных для борьбы со спамом, можно найти по адресу http://mail-abuse.org/tsi/.

Настройка sendmail

В настоящее время sendmail является самым популярным почтовым сервером в мире. Эта программа входит в состав различных дистрибутивных пакетов Linux, в том числе Caldera, Red Hat, Slackware, SuSE и TurboLinux. Несмотря на то что в Debian и Mandrake по умолчанию устанавливаются другие серверы SMTP, sendmail также входит в комплект поставки и при необходимости может заменить существующий почтовый сервер. Как было сказано ранее, в настоящее время доступна версия 8.12.2 sendmail, но некоторые дистрибутивные пакеты Linux до сих пор поставляются с 8.11.x и более ранними версиями.

Формат конфигурационного файла sendmail чрезвычайно сложен, но в распоряжение администратора предоставляются специальные утилиты, посредством которых можно преобразовать файл, заданный в простом формате, в рабочий вариант конфигурационного файла, используемого sendmail. Помимо структуры конфигурационных файлов, в данном разделе будут рассматриваться специальные вопросы настройки sendmail: маскировка адреса, обработка локальных сообщений и обеспечение работы в режиме ретранслятора.

Конфигурационные файлы sendmail

Основной конфигурационный файл sendmail называется sendmail.cf; обычно он располагается в каталоге /etc. Этот файл содержит большое количество опций, представленных в виде, неудобном для восприятия, поэтому анализировать содержимое данного файла и редактировать его чрезвычайно сложно.

Обойти трудности, вызванные сложным форматом sendmail.cf, можно, создавая конфигурационный файл в простом и понятном формате, а затем преобразуя его с помощью утилиты т4 в файл sendmail.cf. Исходный файл, предназначенный для обработки программой m4, заканчивается символами .mc, но конкретное его имя и расположение может изменяться в зависимости от версии операционной системы. В Red Hat

это файл /etc/sendmail.mc, в Slackware — /usr/src/sendmail/cf/cf/linux.smtp.mc, а в SuSE — /etc/mail/linux.mc. Независимо от имени, исходный файл m4 гораздо меньше и удобнее для восприятия, чем создаваемый на его основе файл .cf. Например, если в системе SuSE 7.1 файл sendmail.cf содержит 1669 строк, то файл linux.mc состоит всего из 221 строки, причем основную часть файла занимают комментарии (строки комментариев начинаются символами dnl).

Для того чтобы создать файл sendmail.cf из файла m4, необходимо вызвать программу m4 и перенаправить ввод и вывод. В системе SuSE этот вызов имеет следующий вид:

```
# m4 < /etc/mail/linux.mc > /etc/sendmail.cf
```

В некоторых версиях Linux перед тем как приступать к созданию файла sendmail.cf из исходного файла m4, необходимо установить дополнительный пакет. Например, в Red Hat для создания конфигурационного файла нужен пакет sendmail-cf.

ВНИМАНИЕ Не следует изменять рабочий вариант файла sendmail.cf. Желательно скопировать файл sendmail.cf и исходный файл т4 в другой каталог. Если в результате редактирования вы повредите конфигурационный файл, то, используя созданную копию, вы сможете восстановить рабочую конфигурацию sendmail.

После изменения конфигурационного файла необходимо перезапустить sendmail. Во многих версиях Linux sendmail запускается с помощью сценария SysV, поэтому для перезапуска программы можно использовать опцию restart этого сценария.

Большинство записей в конфигурационном файле т4 задается в следующем формате:

```
ИМЯ_ХАРАКТЕРИСТИКИ('опция1'[, 'опция2' [,...])
```

Имя характеристики — это некоторое содержательное имя, например define или MASQUERADE_AS. В качестве опций могут быть указаны имена узлов, установки, специфические для sendmail, например always_add_domain, и т. д. В определениях некоторых характеристик одинарные кавычки можно не использовать.

ВНИМАНИЕ Кавычки, в которые помещаются опции, на первый взгляд выглядят несколько странно: в качестве открывающей и закрывающей используются различные типы кавычек. Необходимо следить за правильным их использованием. Если вы укажете в исходном файле m4 обычные одинарные кавычки, то либо файл не будет обработан, либо конфигурационный файл sendmail.cf будет сформирован некорректно.

Помимо sendmail.cf, программа sendmail также использует при работе другие файлы.

- access.db. Этот двоичный файл создается на базе текстового файла access. Файл access.db определяет, какие компьютеры могут обращаться к программе sendmail. Конфигурация sendmail в качестве ретранслятора во многом зависит от содержимого этого файла. Многие сценарии запуска sendmail вызывают makemap, и если файл access изменился с момента последнего создания access.db, автоматически генерируется новый файл access.db.

- `aliases.db`. Этот двоичный файл также создается на базе текстового файла с аналогичным именем (`aliases`). Он определяет псевдонимы — имена, эквивалентные другим именам. Так, например, во многих дистрибутивных пакетах для пользователя `root` определяется псевдоним `postmaster`. Возможно, вы захотите создать псевдоним для `root`, чтобы просматривать почту суперпользователя посредством обычной учетной записи. Подобно файлу `access.db`, при выполнении многих сценариев запуска файл `aliases.db` генерируется автоматически.

Рассмотренные выше файлы обычно размещаются в каталоге `/etc` или `/etc/mail`. Кроме того, в этом каталоге находятся другие файлы баз данных, определяющие особенности работы `sendmail`.

Маскировка адреса sendmail

Если вы хотите, чтобы сервер SMTP объявлял себя посредством имени, отличающегося от имени компьютера, на котором он выполняется, вам необходимо сконфигурировать сервер для выполнения маскировки адреса. Принцип маскировки адреса был описан выше в этой главе. Для активизации механизма маскировки адреса вам надо включить в исходный файл t4 следующие две строки:

```
MASQUERADE_AS('требуемый_адрес')
FEATURE(masquerade_envelope)
```

Запись `MASQUERADE_AS` активизирует базовые средства маскировки, которые включают адрес в поле заголовка `From:` в случае, если пользовательская программа не задает имя узла. Поскольку большинство почтовых программ корректно заполняет это поле, данное средство в основном используется, если пользовательская программа сконфигурирована неправильно. Запись FEATURE `(masquerade_envelope)` изменяет поле `From:`, даже если в нем был задан адрес узла.

Если вы хотите, чтобы маскировка применялась только для сообщений от пользователей определенного домена, вам надо включить дополнительные записи, ограничивающие использование средств маскировки.

```
MASQUERADE_DOMAIN('домен-источник')
FEATURE('limited_masquerade')
```

Эти опции сообщают `sendmail` о том, что маскировка должна применяться для адресов указанного домена-источника. Подобная конфигурация чаще всего устанавливается, если почтовый сервер обслуживает два домена.

Настройка sendmail для получения почты

Когда удаленный сервер передает почту вашему серверу, письма адресованы конкретным пользователям, работающим на определенных компьютерах. Чтобы обеспечить доставку локальной почты, программа `sendmail` должна распознавать локальные адреса. Почтовый сервер `sendmail` поддерживает файл, в котором указываются адреса локальных узлов. В различных дистрибутивных пакетах для данного файла используются разные имена. В Red Hat это файл `/etc/mail/local-host-names`, а в SuSE — `/etc/sendmail.cw`. Если вам не удается обнаружить его, найдите в `sendmail.cf` запись, которая начинается символами Fw. Эта запись содержит имя файла, в котором указаны имена локальных узлов. Независимо от имени, содержимое файла представля-

ет собой набор строк, в каждой из которых задано имя узла. Строки, начинающиеся с символа #, считаются комментариями.

Работа в режиме ретранслятора

Как было сказано ранее, ретрансляция является важным режимом работы почтового сервера. Как правило, настраивая sendmail, приходится обеспечивать ретрансляцию писем, созданных на локальной машине, почты из локальной сети и, возможно, сообщений с некоторых удаленных компьютеров. При этом необходимо следить за тем, чтобы сервер был закрыт для спамеров. Кроме того, не исключено, что вам потребуется сконфигурировать систему для передачи исходящей почты, используя в качестве ретранслятора внешний сервер. В конфигурационном файле sendmail предусмотрены различные опции, имеющие отношение к режиму ретрансляции.

Настройка sendmail для ретрансляции писем

При конфигурировании почтового сервера очень часто приходится обеспечивать передачу писем из локальной сети. Сервер получает сообщения от пользовательских программ и в случае возникновения проблем с передачей данных в сети может временно хранить эти сообщения на своем диске. Для обеспечения подобного взаимодействия адрес почтового сервера должен быть указан при настройке программ подготовки почты.

По умолчанию почтовый сервер, инсталлированный в сети, не настроен для работы в качестве ретранслятора. При попытке передать письмо на сервер программа подготовки почты получит в ответ сообщение "relaying denied" ("ретрансляция запрещена"). Для того чтобы программа sendmail работала в качестве ретранслятора, надо активизировать соответствующие компоненты. В частности, в исходном конфигурационном файле необходимо задать записи FEATURE, указав в них следующие опции.

- relay_entire_domain. Если указана данная опция, sendmail принимает сообщения из своего домена, а также письма, адресованные пользователям в его домене. Для определения принадлежности к домену sendmail использует сервер DNS. Опция relay_entire_domain представляет собой удобное средство обеспечения ретрансляции.

- relay_local_from. Эта опция указывает серверу sendmail, что он должен принимать все письма, из содержимого поля From: которых следует, что они передаются из локального домена. От предыдущей опции relay_local_from отличается тем, что для принятия решения об обработке письма используется лишь адрес в поле From:, посредством которого система представляется другим компьютерам. Этот адрес может достаточно просто быть фальсифицирован. Данная опция не обеспечивает приемлемого уровня защиты от спама.

- relay_based_on_MX. Данная опция означает, что сервер sendmail должен принимать письма в том случае, если в домене, которому принадлежит отправитель, присутствует запись MX, содержащая указание на этот сервер. Опция relay_based_on_MX обеспечивает простой и удобный способ управления ретрансляцией. Чтобы настроить почтовый сервер для поддержки еще одного домена, не надо вносить изменения в конфигурационные файлы sendmail, достаточно лишь изменить конфигурацию сервера DNS. Однако подобный подход имеет су-

щественный недостаток. Спамеры, поддерживающие собственные домены, могут легко создать запись MX и использовать ваш сервер в своих целях.

- relay_hosts_only. Если вы зададите этуопцию, sendmail будет использовать базу данных для принятия решений о предоставлении доступа. Письма будут приниматься лишь от тех пользователей, которые работают на компьютерах, указанных в базе. Данную опцию удобно использовать для того, чтобы ограничить доступ к серверу некоторым набором узлов сети.

- access_db. Данная опция часто устанавливается по умолчанию при настройке sendmail. Подобно опции relay_hosts_only, она сообщает sendmail о том, что решение о предоставлении доступа должно приниматься на основе содержимого базы данных. Однако в данном случае в базе могут указываться не только отдельные компьютеры, но и целые домены.

ВНИМАНИЕ Для управления ретрансляцией может использоваться также опция | promiscuous_relay, но применять ее не рекомендуется. Она открывает доступ к серверу для любого компьютера. Сконфигурированный подобным образом сервер рано или поздно будет обнаружен спамерами и использован для передачи рекламных сообщений.

Ниже приведен пример записи в конфигурационном файле m4.

```
FEATURE('access_db')
```

Данная запись часто устанавливается по умолчанию, но она не обеспечивает реальной ретрансляции писем, передаваемых с удаленных узлов, так как в файле access.db, автоматически создаваемом при установке системы, указывается только локальный домен.

Как вы уже знаете, при запуске программа sendmail читает содержимое файла access.db. Этот файл обычно хранится в каталоге /etc или /etc/mail и создается на базе файла access. Пример файла access приведен ниже.

```
# Разрешить прием писем с localhost...
localhost.localdomain        RELAY
localhost                    RELAY
127.0.0.1                    RELAY
# Разрешить прием писем из локальной сети
192.168.99                   RELAY
```

Первые три записи присутствуют практически в любой конфигурации. Они сообщают sendmail о том, что программа должна принимать письма с локального узла. Эти записи обеспечивают работу локальных почтовых программ. Последняя запись указывает на то, что сервер должен принимать для ретрансляции письма, отправленные из сети 192.168.99.0/24. Вместо IP-адресов можно указывать доменные имена, но IP-адреса сложнее фальсифицировать, поэтому при использовании их повышается уровень безопасности системы.

Все приведенные примеры оканчиваются опцией RELAY, но кроме нее в файле access могут также использоваться и другие опции.

- OK. Эта опция сообщает sendmail о том, что локальные письма должны приниматься, несмотря на то, что другие правила требуют отвергать их.

- RELAY. Как вы, возможно, догадались, данная опция обеспечивает обработку писем, переданных с указанного компьютера или из указанного домена. Она также сообщает, что сервер должен передавать письма, поступившие на эти компьютеры или в эти домены.
- REJECT. Если вы собираетесь блокировать почту, поступающую с определенного узла или из определенного домена, вам следует использовать данную опцию. При этом письма будут возвращаться отправителю.
- DISCARD. Данная опция выполняет те же действия, что и REJECT, но письма не возвращаются отправителю.
- ллл *текст*. Эта опция также работает подобно **REJECT**, но в возвращаемое сообщение она включает код ошибки ллл и указанный текст.

Отредактировав файл access, вам необходимо сгенерировать двоичный файл базы данных. Для этого надо использовать команду makemap, которая имеет следующий вид:

```
# makemap hash /etc/mail/access.db < /etc/mail/access
```

При инсталляции **sendmail** такая команда часто включается в сценарий запуска, поэтому вызывать ее вручную не всегда нужно. В любом случае после внесения изменений в файл access необходимо перезапустить программу sendmail.

Настройка sendmail для передачи почты через ретранслятор

В предыдущем разделе рассматривался вопрос об использовании программы sendmail для ретрансляции почты. Однако, настраивая почтовый сервер, необходимо также принимать во внимание и вопросы передачи писем через ретранслятор, функции которого выполняет другой сервер. Часто при организации работы небольшой сети и даже одного компьютера приходится использовать в качестве ретранслятора почтовый сервер провайдера. Несмотря на то что компьютер под управлением Linux, на котором установлена программа sendmail, может передавать почту самостоятельно, многие провайдеры запрещают это, включая IP-адреса, предоставляемые своим клиентам, в списки адресов, предназначенные для борьбы со спамом. Кроме того, некоторые компьютеры бывают выключены в течение длительного времени, в результате чего становится невозможным их использование в качестве почтовых серверов. В особенности это относится к портативным компьютерам.

В большинстве случаев конфигурация sendmail, установленная по умолчанию, не предполагает передачу писем через ретранслятор. Чтобы обеспечить такую возможность, надо включить в конфигурационный файл т4 следующую запись:

```
FEATURE('nullclient', 'outgoing.mail.relay')
```

В данном случае **outgoing.mail.relay** — это имя компьютера, используемого для ретрансляции почты. После того как вы создадите файл **sendmail.cf** и перезапустите sendmail, вся исходящая почта будет передаваться через указанный почтовый сервер. Выполнив настройку sendmail, убедитесь, что письма корректно доставляются адресатам.

Конфигурация sendmail для противодействия попыткам передачи спама

Существует несколько способов настройки sendmail для блокирования поступающих рекламных сообщений и предотвращения попыток использования сервера для передачи спама. Один из способов состоит в использовании файла access и его двоичного аналога access.db. С помощью файла access.db можно блокировать спам на основании анализа адресов отправителей. Если вы зададите для некоторых доменов или компьютеров, указанных с помощью имени или IP-адреса, опции REJECT или DISCARD, сообщения из этих источников будут отвергаться. Если вы регулярно получаете рекламные сообщения с определенных адресов, этот способ позволит избавиться от них. Следуя описанному подходу, необходимо соблюдать осторожность, так как вместе со спамом будут отвергнуты и корректные сообщения, приходящие с тех же адресов или из тех же доменов. Если вы блокируете почту, поступающую из сети, которая принадлежит популярному провайдеру, вы рискуете потерять нужные вам письма.

Другой способ блокирования спама состоит в применении списков IP-адресов. Для того чтобы реализовать этот способ, надо указать в конфигурационном файле т4 опцию dnsbl.

```
FEATURE(dnsbl, 'blackholes.mail-abuse.org', 'Rejected - see \
http://www.mail-abuse.org/rbl/')
```

Данная запись указывает sendmail на то, что при проверке входящей почты должен использоваться список MAPS RBL. Если вы хотите использовать другой список, вам надо изменить вторую опцию в данной записи. Последнее поле записи содержит строку, которая включается в состав возвращаемого сообщения. В этой строке вы можете указать отправителю адрес Web-узла, содержащего список IP-адресов, на основании которого было отвергнуто его сообщение. Если окажется, что корректное сообщение было заблокировано по ошибке, его автор сможет принять меры для того, чтобы разрешить проблему.

В версии 8.10 программы sendmail порядок использования списков IP-адресов был существенно изменен. В этой главе описаны правила, применяемые в этой и последующих реализациях. Дополнительную информацию по данному вопросу вы найдете по адресу http://mail-abuse.org/rbl/usage.html.

Чтобы предотвратить использование почтового сервера для неавторизованной рассылки почты, необходимо ограничить доступ к нему. Проще всего сделать это, указав в файле access IP-адреса или диапазоны адресов компьютеров, которые могут использовать сервер для передачи писем. В некоторых случаях можно также указать другие опции управления ретрансляцией. Применение опции promiscuous_relay — не допустимо.

ВНИМАНИЕ Версии sendmail, предшествующие 8.9.0, настроены так, чтобы любой компьютер мог воспользоваться сервером для передачи писем. Такой сервер необходимо заменить новой версией или перенастроить его, чтобы неограниченные услуги ретранслятора не предоставлялись другим узлам. Информацию по этому вопросу вы найдете по адресу http://mail-abuse.org/tsi/ar-fix.html#sendmail_8. Версии sendmail, предшествующие 8.8.4, перенастроить крайне сложно. Гораздо проще обновить почтовый сервер.

Настройка Exim

Сервер Exim применяется по умолчанию в Debian GNU/Linux и пользуется умеренной популярностью. Данный сервер можно использовать и с другими пакетами. Так, например, Exim поставляется в составе расширения PowerTools системы Red Hat, поэтому его достаточно легко установить в Red Hat и других подобных дистрибутивных пакетах. Подобно **sendmail,** Exim представляет собой единую программу, но формат конфигурационного файла Exim сравнительно прост. Exim обладает приблизительно такими же возможностями, как и sendmail; в данном разделе рассматриваются некоторые из них, например, маскировка адресов, обработка писем, адресованных в разные домены, и использование режима ретрансляции почты.

Поскольку Exim является сервером по умолчанию только для Debian, в данном разделе в основном принимается во внимание конфигурация данного сервера, устанавливаемая в системме Debian. В других системах для Exim может быть по умолчанию выбрана другая конфигурация.

Конфигурационные файлы Exim

Главный конфигурационный файл Exim называется **exim.conf.** Обычно он располагается в каталоге /etc. В состав этого файла входят записи, представленные в следующем формате:

опция = значение

Как обычно, строки, содержащие комментарии, начинаются с символа #. Файл **exim.conf,** который используется в системе Debian, в основном состоит из комментариев, поясняющих назначение каждой записи. Комментарии существенно упрощают редактирование конфигурационного файла.

При инсталляции Exim в системе Debian запускается сценарий с именем **eximconfig,** в процессе выполнения которого генерируется файл **exim.conf.** Данный сценарий можно использовать для изменения конфигурации Exim; при этом нет необходимости непосредственно редактировать файл **exim.conf.** Если вам надо внести лишь незначительные изменения в конфигурацию системы, удобнее модифицировать **exim.conf** вручную, так как при использовании eximconfig приходится отвечать на целый ряд вопросов. Во многих случаях eximconfig может оказаться очень полезным инструментом, в особенности если вы мало знакомы со структурой конфигурационного файла Exim. В частности, данный сценарий помогает выбрать значения опций, наиболее подходящие для вашей системы.

Помимо **exim.conf,** Exim использует в качестве источников дополнительной информации другие файлы. Файлы, применяемые данным сервером в системе Debian, перечислены ниже.

- /etc/aliases. Этот файл выполняет те же функции, что и аналогичный файл sendmail. Он позволяет связать две учетные записи так, что письмо, адресованное одному пользователю, будет направлено другому. Например, если в этом файле присутствует запись **root: amelia,** то письмо, адресованное root, получит пользователь **amelia.** В файле **aliases** можно также указывать адреса, не принад-

лежащие локальным пользователям. Например, наличие записи `root: amelia@pangaea.edu` приведет к тому, что письмо, принятое для локального пользователя root, будет перенаправлено по адресу `amelia@pangaea.edu`. В отличие от `sendmail`, в сервере Exim файл `aliases` не преобразовывается в двоичный формат.

- `/etc/email-addresses`. Записи в этом файле используются для изменения содержимого полей `From:` в заголовках исходящих сообщений. Например, наличие записи `ben: bfranklin@pangaea.edu` приведет к тому, что письмо, отправленное с локального компьютера пользователем ben, придет к получателю как сообщение от `bfranklin@pangaea.edu`.

Сценарий `eximconf` создает в файле `/etc/aliases` записи, посредством которых почта, адресованная `postmaster` перенаправляется root, а письма, непосредственно направленные root, будут переданы пользователю, которого вы укажете. Содержимое описанных выше файлов можно удалять или модифицировать, а при необходимости вы можете включать в эти файлы новые записи. Файл `/etc/email-addresses`, создаваемый по умолчанию в системе Debian, содержит лишь комментарии.

Маскировка адресов

Как было сказано ранее, вам может потребоваться, чтобы в сообщениях вместо имени, возвращаемого по команде `hostname`, отображалось другое имя узла или домена. Основные средства маскировки адресов включаются посредством опции qualify_domain. С помощью данной опции задается имя домена. Если почтовая программа не сгенерирует информацию об адресе, имя домена будет автоматически включено в сообщение. Предположим, что в файле `exim.conf` присутствует следующая запись:

```
qualify_domain = threeroomco.com
```

Если пользователь ben отправит письмо, а программа, с помощью которой это письмо было подготовлено, не укажет в поле From: имя домена, то Exim добавит имя `threeroomco.com`. Если доменное имя адреса не соответствует имени `threeroomco.com`, то Exim заменит адрес. Таким образом, содержимое поля From: будет выглядеть так: `ben@threeroomco.com`.

Еще одна опция, которую можно использовать для маскировки адресов, называется `primary_hostname`. Она применяется подобно `qualify_domain`, и ее значение принимается в качестве значения по умолчанию для `qualify_domain`. Значение primary_hostname используется при переговорах о взаимодействии Exim и удаленного сервера имен. Имя, задаваемое посредством данной опции, применяется при формировании заголовка `Received:`.

Для более сложной маскировки адресов применяется файл `/etc/email-addresses`. Строго говоря, на файл `/etc/email-addresses` ссылается запись, расположенная в конце конфигурационного файла `exim.conf`. Эта запись имеет следующий вид:

```
*@threeroomco.com  ${lookup{$1}lsearch{/etc/email-addresses}\
                   {$value}fail} bcfrF
```

Это одна из наиболее сложных записей, содержащихся в файле `exim.conf`. При настройке сервера не следует редактировать ее, допустимо лишь изменить имя домена в начале строки. С помощью данной записи Exim проверяет каждый адрес на принадлежность

домену threeroomco.com, а затем использует файл /etc/email-addresses для замены адреса. В первом поле записи, содержащейся в файле /etc/email-addresses (перед двоеточием), указывается почтовый адрес, предназначенный для сравнения, а во втором поле (после двоеточия) — адрес для замены. Данное средство позволяет выполнять маскировку для каждого пользователя; чтобы сделать это, достаточно лишь отредактировать файл email-addresses. При необходимости вы можете обрабатывать письма из разных доменов. Для этого надо либо продублировать приведенную выше запись в exim.conf, либо включить всю информацию, необходимую для замены адресов, в один файл email-addresses.

В данном разделе рассмотрены лишь некоторые средства маскировки адресов, предоставляемые Exim. Дополнительную информацию по этому вопросу вы можете получить в документации на Exim, обратившись по адресу http://www.exim.org/exim-html-3.30/doc/html/spec_34.html.

Настройка Exim для приема почты

В конфигурационном файле exim.conf предусмотрены различные опции, позволяющие указать серверу, следует ли интерпретировать адрес как локальный. Эти опции кратко описаны ниже.

- local_domains. В качестве значения данной опции задается список доменных имен, разделенных двоеточиями. Эти имена Exim должен интерпретировать как локальные. Например, запись local_domains = localhost:threeroomco.com сообщает Exim о том, что адреса localhost и threeroomco.com являются локальными и письма, в которых они указаны, необходимо непосредственно доставлять пользователям. По умолчанию значение данной опции принимается равным значению qualify_recipient. Опция qualify_recipient задает имя узла для входящих сообщений, в которых такая информация отсутствует.

- local_domains_include_host. Если значение данной опции равно true, Exim принимает письма, в адресе которых указано имя компьютера. Тот же результат можно получить, добавив имя узла к списку local_domains.

- local_domains_include_host_literals. Если значение данной опции равно true, Exim принимает письма, в которых указан IP-адрес компьютера. Например, если Exim выполняется на компьютере с адресом 172.24.98.2 и на этом компьютере имеется учетная запись пользователя ben, Exim примет письмо с адресом ben@[172.24.98.2]. Если вы не хотите, чтобы подобные письма обрабатывались сервером, необходимо установить значение false опции local_domains_include_host_literals.

Сценарий eximconfig устанавливает эти опции исходя из ответов администратора на вопросы о домене, для которого необходимо принимать письма. Таким образом, если вы внимательно отнесетесь к вопросам, задаваемым данным сценарием, вы обнаружите, что необходимые для вас значения опций уже установлены.

Конфигурация Exim для ретрансляции писем

Подобно sendmail, в сервере Exim предусмотрен ряд опций, предназначенных как для ретрансляции писем, переданных другими программами, так и для использования

в качестве ретрансляторов других серверов. Сценарий eximconfig задает администратору вопросы, касающиеся ретрансляции писем, и в большинстве случаев устанавливает приемлемую конфигурацию сервера. Уточнить настройку Exim можно с помощью непосредственного редактирования файла exim.conf.

Настройка Exim для работы в режиме ретранслятора

Основные опции exim.conf, предназначенные для реализации режима ретрансляции писем, описаны ниже.

- host_accept_relay. Для того чтобы сервер Exim мог ретранслировать письма, переданные определенными компьютерами, вам надо указать в качестве значения данной опции их адреса (адреса отделяются друг от друга двоеточиями). В конфигурационном файле должно быть как минимум указано выражение host_accept_relay = localhost, позволяющее Exim передавать письма, подготовленные локальными почтовыми программами. По мере расширения списка (в котором могут быть указаны доменные имена, IP-адреса, а также использоваться символы групповых операций) увеличивается число компьютеров, которым позволено пользоваться услугами сервера для передачи почты. Например, выражение host_accept_relay = localhost:192.168.99.0/24:*.pangaea.edu указывает на то, что письма для передачи должны приниматься с локального узла, со всех узлов сети 192.168.99.0/24, а также со всех компьютеров домена pangaea.edu. Использование данной опции для указания IP-адресов компьютеров, принадлежащих домену, — один из самых безопасных способов обеспечения ретрансляции писем.

- relay_domains. В качестве значения данной опции можно указать одно или несколько имен доменов, разделенных двоеточиями. В результате сервер Exim будет обрабатывать письма, направленные *с* любого компьютера, принадлежащего указанным доменам. Эта опция полезна тогда, когда необходимо, чтобы сервер обслуживал несколько доменов или один большой домен. Аналогичных результатов можно добиться, включая символ групповой операции (*) в имена, задаваемые в качестве значения опции host_accept_relay.

- relay_domains_include_local_mx. Если вы зададите значение yes данной опции, доступ к почтовому серверу автоматически получат компьютеры, указанные в записях MX серверов DNS. Такой подход очень удобен, так как избавляет от необходимости перенастраивать Exim при изменении конфигурации домена. Однако в этом случае повышается опасность использования сервера спамерами, которые имеют возможность управлять доменом и включать в конфигурационный файл сервера DNS записи MX.

- sender_address_relay. В качестве значения данной опции задается список почтовых адресов, разделенных двоеточиями, для которых разрешено использование сервера в качестве ретранслятора. В обычных условиях письмо должно соответствовать как значению данной опции, так и адресу узла, указанному с помощью опции host_accept_relay. (Вы можете задать проверку на соответствие любой из этих опций, включив в конфигурационный файл выражение relay_match_host_or_sender = yes, но такая конфигурация опасна для системы, так как почтовый адрес легко подделать.) Данную опцию можно применить для того, что-

бы ограничить круг пользователей, имеющих право использовать сервер в качестве ретранслятора.

Приведенные здесь опции позволяют настроить Exim для работы в режиме ретранслятора и указать, какие компьютеры локальной сети или внешних доменов имеют право доступа к данному серверу. Данные опции позволяют решать большинство задач по обеспечению ретрансляции почты. Если же вам необходимо установить специальную конфигурацию сервера, вы можете воспользоваться дополнительными опциями, например, `host_auth_accept_relay` (которая выполняет аутентификацию удаленной системы перед ретрансляцией писем) и `tls_host_accept_relay` (которая требует, чтобы удаленная система использовала средства аутентификации и кодирования TLS).

Настройка Exim для передачи почты через ретранслятор

Если ваш почтовый сервер должен передавать почту через ретранслятор, его надо настроить соответствующим образом. В конфигурационном файле Exim не предусмотрена специальная опция, позволяющая решить эту задачу, однако сценарий `eximconfig` генерирует набор записей, обеспечивающих необходимые установки. Опции, созданные с помощью `eximconfig`, выглядят следующим образом:

```
smarthost:
    driver = domainlist
    transport = remote_smtp
    route_list = "* franklin.threeroomco.com bydns_a"
end
```

Приведенная выше группа записей сообщает Exim о том, что письма, адресованные внешним пользователям, надо передавать через узел `franklin.threeroomco.com`. Чтобы использовать другой ретранслятор, надо изменить значение соответствующей опции.

Настройка Exim для противодействия распространению спама

В сервере Exim предусмотрен набор правил фильтрации. С помощью этих правил вы можете задавать адреса узлов, которым должна быть запрещена передача писем, указывать пользователей, от которых почта не должна приниматься, а также выполнять другие проверки на основе самых разнообразных критериев. Основные опции фильтрации описаны ниже.

- `host_reject`. Данная опция задается в конфигурационном файле `exim.conf`. Ее значение представляет собой список имен узлов и доменов, а также IP-адресов, разделенных двоеточиями. Почта, переданная с компьютеров, указанных посредством данной опции, должна блокироваться. Например, запись `host_reject = *.badspammer.net: 10.16.8.0/24` указывает на то, что письма из домена `badspammer.net`, а также из сети 10.16.8.0/24 должны отвергаться. Система отказывается взаимодействовать с удаленным компьютером, заданным с помощью опции `host_reject`, уже на этапе установления соединения. В результате удаленный компьютер предпринимает повторные попытки обращения к вашему серверу, но связанная с этим дополнительная нагрузка на линии связи и компьютеры небольшая.

- host_reject_recipients. Данная опция действует так же, как и host_reject, но почтовые сообщения отвергаются лишь в процессе взаимодействия с удаленной программой, в частности, в тот момент, когда она передает команду RCPT TO:. В результате попытки пересылки писем немедленно прекращаются.

- sender_reject. Данная опция блокирует письма от указанных отправителей. Роль отправителя может выполнять либо целый домен, либо отдельный пользователь в домене. Например, опция sender_reject = spammer@abigisp.com: badspammer.net указывает на то, что письма из домена badspammer.net и от пользователя spammer@abigisp.com должны отвергаться. Сервер Exim прекращает взаимодействие сразу же, как только сможет идентифицировать отправителя. В некоторых случаях это приводит к повторным попыткам передачи сообщений, предпринимаемым удаленными программами.

- sender_reject_recipients. Данная опция действует подобно опции sender_reject, но взаимодействие с удаленным компьютером прекращается после того, как выполняющаяся на нем программа укажет адрес получателя, т. е. передаст команду RCPT TO:. Данный подход более эффективен по сравнению с использованием опции sender_reject, так как при этом удаленная система больше не предпринимает попыток передачи сообщений.

- **Фильтры, определяемые пользователем.** Сервер Exim предоставляет пользователям возможность создавать собственные фильтры. Для их формирования используются файлы .forward, находящиеся в рабочих каталогах пользователей. Возможность создания новых фильтров превращает Exim в чрезвычайно мощный и гибкий инструмент. Фильтры, определяемые пользователем, во многом похожи на фильтры Procmail, которые будут рассматриваться далее в этой главе. Подробное описание средств создания пользовательских фильтров содержится файле filter.txt.gz, который поставляется в составе Exim. В Debian GNU/Linux этот файл располагается в каталоге /usr/doc/exim; распаковать его можно с помощью утилиты gunzip.

Если вам необходимо задать длинный список отправителей, письма от которых не должны приниматься, вы должны указать их в отдельном файле, а ссылку на этот файл задать в качестве значения соответствующей опции. Помимо перечисленных выше средств, в сервере Exim также предусмотрен ряд опций, предназначенных для работы со списками IP-адресов. Эти опции, располагающиеся в файле exim.conf, кратко описаны ниже.

- rbl_domains. Значением этой опции является перечень адресов серверов, поддерживающих списки IP-адресов (эти серверы описаны в табл. 19.1). Адреса серверов разделяются двоеточиями и могут сопровождаться последовательностями символов /warn или /reject. Значение /warn указывает серверу Exim на то, что он должен добавить поле заголовка с предупреждающим сообщением (который впоследствии может быть использован фильтром Procmail), а /reject означает, что письмо должно быть отвергнуто. Кроме того, в составе данной опции могут также использоваться последовательности символов /accept (формирование "белого списка") и /skiprelay (если домен отправителя указан в опции host_accept_relay, то список IP-адресов не должен использоваться).

- rbl_hosts. По умолчанию принимается значение * данной опции; оно указывает на то, что сервер должен проверять все узлы, с которыми он взаимодействует, на со-

ответствие спискам IP-адресов, указанных посредством опции rbl_domains. При необходимости вы можете освободить некоторые серверы от этой проверки. Чтобы сделать это, вам надо указать их имена перед символом *; каждому имени должен предшествовать символ !. Например, выражение rbl_hosts = !ok.pangaea.edu:* освобождает ok.pangaea.edu от проверки на принадлежность спискам IP-адресов.

- rbl_reject_recipients. Последовательности символов /warn и /reject в составе значения опции rbl_domains указывают, следует ли добавить к письму поле заголовка с предупреждающим сообщением или отказаться от получения письма. Если эти последовательности не указаны, Exim по умолчанию отказывается от письма. Изменить поведение сервера позволяет опция rbl_reject_recipients. Если вы зададите в конфигурационном файле выражение rbl_reject_recipients = no, Exim будет по умолчанию добавлять в заголовки писем предупреждающие сообщения.

- recipients_reject_except. Данная опция позволяет задавать исключения из списков IP-адресов. Например, если указана опция recipients_reject_except = postmaster@threeroomco.com, сервер Exim будет получать письма, адресованные пользователю postmaster@threeroomco.com, даже в том случае, если они были отправлены с компьютера, указанного в списке IP-адресов.

Дополнительную информацию об опциях, предназначенных для работы со списками IP-адресов, вы найдете в документации, поставляемой в комплекте с сервером Exim. Помимо опций, рассмотренных выше, Exim поддерживает дополнительные опции, имеющие лишь косвенное отношение к борьбе со спамом. Эти опции перечислены ниже.

- headers_check_syntax. Exim может проверить формат сообщений и отвергнуть их, если они составлены некорректно. Серверы некоторых спамеров неправильно формируют заголовки, поэтому, отвергая подобные сообщения, вы избавляетесь от писем с рекламными сообщениями. Чтобы сервер блокировал некорректно составленные письма, необходимо задать значение true опции headers_check_syntax.

- helo_verify. В процессе взаимодействия по протоколу сервер SMTP передает команду HELO или EHLO, указывая в ее составе свое имя. Обычно Exim не требует этой команды, но при необходимости вы можете задать список узлов, которые должны соблюдать все правила взаимодействия серверов. Так, например, выражение helo_verify = * указывает, что все удаленные серверы должны строго следовать протоколу обмена. Опция helo_verify не только требует передавать команду HELO или EHLO, но также включает проверку соответствия IP-адресов доменным именам. Системы спамеров часто бывают неверно сконфигурированы, поэтому передаваемые ими письма не выдерживают подобной проверки. Однако следует учитывать, что серверы, с которых поступают обычные письма, также бывают некорректно настроены. В этом случае будут потеряны нужные вам сообщения.

- message_size_limit. Данная опция также имеет лишь отдаленное отношение к борьбе со спамом, но с ее помощью можно избавиться от некоторых рекламных сообщений. По умолчанию устанавливается значение 0 опции message_size_

limit, которое отменяет ограничения на размер писем. Если вы зададите положительное значение данной опции, оно будет определять максимальный размер письма. Это предотвратит получение рекламных сообщений большого объема.

Средства фильтрации сообщений, предоставляемые Exim, и в особенности фильтры, определяемые пользователем, позволяют настроить систему в соответствии с вашими потребностями.

Настройка Postfix

Как и в сервере Exim, конфигурационный файл Postfix достаточно прост. Настроить сервер Postfix сможет каждый, кто хотя бы поверхностно знает терминологию SMTP и способен понять назначение имен. Сервер Postfix имеет модульную структуру, т. е. его функции обеспечиваются совместным выполнением нескольких программ. Postfix предоставляет приблизительно те же возможности, что и Exim. Подобно другим серверам SMTP, Postfix обеспечивает маскировку адресов, прием писем, адресованных в локальные домены, работу в режиме ретрансляции почты, а также предоставляет возможность противодействия распространению спама.

Postfix по умолчанию используется в Mandrake, но также может быть установлен и в других системах, например в Debian и SuSE. Этот сервер также входит в состав PowerTools. RPM-пакет, предназначенный для Mandrake, может быть установлен в других дистрибутивных пакетах Linux, но сценарии SysV, содержащиеся в данном пакете, работать не будут. Поскольку Postfix чаще всего применяется совместно с Mandrake, материал данного раздела будет излагаться с учетом конфигурации Postfix, устанавливаемой по умолчанию для данной версии системы. Настройка Postfix для остальных систем отличается от конфигурации для Mandrake лишь отдельными деталями.

Конфигурационный файл Postfix

Особенности выполнения Postfix определяются содержимым конфигурационного файла main.cf, который обычно располагается в каталоге /etc/postfix. Большинство записей в этом файле представлены в следующем формате:

опция = значение

Некоторые записи main.cf определяют переменные, используемые далее в этом файле. Чтобы ссылаться на значение опции как на переменную, надо указать перед именем опции символ S и включить полученное имя в правую часть записи. В качестве примера рассмотрим следующие две записи (между которыми могут находиться другие строки):

```
myhostname = franklin.threeroomco.com
myorigin — $myhostname
```

В первой записи переменной myhostname присваивается имя узла franklin.threeroomco.com, затем это же значение присваивается переменной myorigin. Подобные цепочки определений часто используются в Postfix, поэтому, чтобы определить значение переменной, надо проследить его, перемещаясь назад по конфигурационному файлу.

Файл main.cf в основном состоит из комментариев, которые содержатся в строках, начинающихся в символа #. Комментарии подробно описывают назначение каждой опции,

поэтому вы можете достаточно подробно изучить конфигурацию Postfix, просматривая лишь содержимое конфигурационного файла.

В файле `main.cf` содержатся ссылки на другие файлы. Как и в сервере `sendmail`, некоторые из этих файлов (оканчивающиеся символами `.db`) представлены в двоичном формате. Они создаются на базе текстовых файлов с теми же именами, за исключением суффикса .db. В процессе использования сервера наиболее часто приходится редактировать файл `aliases` (который преобразуется в файл `aliases.db`). Как и в одноименном файле сервера sendmail, в файле `aliases` задаются псевдонимы, используемые при доставке писем. Например, запись `root: amelia` указывает на то, что все письма, адресованные `root`, должны быть доставлены пользователю `amelia`. Для того чтобы преобразовать текстовый файл `aliases` в двоичный файл `aliases.db`, надо вызвать команду `postalias aliases`, указав перед этим в качестве текущего каталог, в котором содержится файл `aliases`.

После того как вы модифицируете содержимое текстового файла и создадите файл .db, пройдет некоторое время перед тем, как Postfix учтет внесенные изменения. Для того чтобы ускорить этот процесс, необходимо задать команду `postfix reload` либо перезапустить Postfix, используя для этого сценарий SysV.

Маскировка адресов

Опция `myorigin` позволяет задать имя, под которым Postfix будет представляться при взаимодействии с другими системами. По умолчанию в качестве значения данной опции задается переменная `$myhostname`, которая, в свою очередь, определяет доменное имя компьютера. Конфигурация по умолчанию приемлема во многих случаях, но если вашему компьютеру соответствует несколько имен или если вы хотите вместо имени узла использовать имя домена, вам придется изменить настройку сервера. Для этого надо задать новое значение опции myorigin, например:

```
myorigin = threeroomco.com
```

При желании вы можете указать в качестве значения опции переменную, например $mydomain. По умолчанию значением переменной `$mydomain` является значение `$myhostname`, из которого исключен компонент, находящийся слева. Например, если переменная `$myhostname` имеет значение `franklin.threeroomco.com`, то значение `$mydomain` будет равно `threeroomco.com`. В файле `main.conf` содержится много закомментированных записей. В некоторых случаях вы можете изменить конфигурацию, выбрав подходящую для вас запись и удалив символ комментариев.

Опция myorigin определяет только базовые средства маскировки адресов. Значение данной опции используется лишь в ходе начальных переговоров с удаленным сервером, предусмотренных протоколом SMTP, и для указания доменного имени в поле `From:`, если соответствующие данные не были включены в это поле программой подготовки писем. Если ваш почтовый сервер выступает в качестве ретранслятора по отношению к другим системам вашего домена, которые, возможно, настроены для включения в заголовок полного адреса, вам потребуется выполнять более сложные действия по маскировке адресов. Предположим, например, что клиентская программа, использующая сервер Postfix для передачи писем, включает в поле `From:` адрес `ben@client.threeroomco.com`. Предположим также, что вы хотите удалить идентификатор `client` так, чтобы адрес имел вид `ben@client.threeroomco.com`. Учитывая, что значением переменной `$mydomain`

является имя домена threeroomco.com, вы можете использовать для получения требуемого результата следующую запись:

```
masquerade_domains = $mydomain
```

Данная опция указывает серверу Postfix на то, что при обработке сообщения должна быть удалена часть доменного имени, не относящаяся к имени домена, указанного посредством переменной $mydomain. В результате в поля From: и To: вместо имени узла, принадлежащего домену $mydomain, будет записано имя домена.

Postfix позволяет выполнять еще более сложные действия по маскировке адресов. В частности, вы можете указать Postfix на необходимость изменить содержимое файла базы данных. Для этого используется опция sender_canonical_maps.

```
sender_canonical_maps = hash:/etc/postfix/sender_canonical
```

В файл sender_canonical необходимо включить записи, используемые для преобразования имен. Каждая строка этого файла должна содержать адрес, который может находиться в составе заголовка, и адрес, которым он должен быть заменен. Следующие две строки заменяют имена client.threeroomco.com и localhost на threeroomco.com:

```
@client.threeroomco.com @threeroomco.com
@localhost @threeroomco.com
```

Аналогичный подход можно использовать для преобразования пользовательских имен. Например, ваш сервер может выполнять роль посредника между сетями, в которых для представления имен применяются различные форматы. Включив в файл, предназначенный для преобразования, записи, отображающие имя каждого пользователя, вы обеспечите соответствие имен в разных форматах.

После создания файла sender_canonical его необходимо преобразовать в двоичный формат посредством команды postmap sender_canonical. Чтобы внесенные в файл изменения были учтены сервером Postfix, надо вызвать команду postfix reload либо перезапустить сервер.

Выполняя настройку сервера, необходимо ограничиваться минимально допустимым уровнем маскировки адресов. В большинстве случаев конфигурация, установленная по умолчанию, позволяет Postfix выполнять свои функции, иногда приходится лишь скорректировать значение myorigin. Опция masquerade_domains в основном применяется в тех случаях, когда сервер принимает для передачи письма, которые уже были обработаны почтовым сервером, выполняющимся в системе Linux или UNIX. Средства преобразования адресов воздействуют не только на заголовок From:, они также затрагивают содержимое заголовка Received:. Многие администраторы неохотно используют данные средства, но в ряде случаев они могут быть очень полезны, особенно если ваши программы требуют, чтобы имена и адреса в поле From: были представлены в специальном формате.

Настройка Postfix для получения почты

Подобно другим почтовым серверам, Postfix считает локальными только адреса некоторых узлов. Чтобы определить, какой из компьютеров является локальным, Postfix использует опцию mydestination. По умолчанию для данной опции приняты значения $myhostname и localhost.$mydomain. Например, если переменная $mydomain имеет значение threeroomco.com, а $myhostname — franklin.threeroomco.

com, то Postfix будет принимать письма, направленные на компьютеры **franklin.threeroomco.com** и **localhost.threeroomco.com**.

При необходимости вы можете изменить или дополнить значения данной опции. Например, если почтовый сервер обслуживает домен, вам необходимо добавить переменную **$mydomain**. Неплохо также указать для данной опции значение localhost. Значения опции mydestination отделяются друг от друга запятыми. Например, для почтового сервера, обслуживающего один домен, в конфигурационный файл можно включить следующее выражение:

```
mydestination = localhost, localhost.$mydomain, $myhostname,
                $mydomain
```

Для того чтобы указать, что опция mydestination занимает несколько строк, символ \ использовать не надо. Строка считается продолжением предыдущей, если она начинается с пробела или знака табуляции.

Вы можете настроить сервер Postfix для обслуживания нескольких доменов, указав их посредством одной опции mydestination. В этом случае имена большинства доменов задаются явно.

Конфигурация Postfix для ретрансляции писем

Подобно большинству почтовых серверов, Postfix поддерживает опции, предназначенные для управления ретрансляцией писем. Эти опции, расположенные в файле **main.cf**, позволяют настроить сервер как для работы в режиме ретрансляции, так и для использования в качестве ретранслятора другого сервера.

Настройка Postfix для работы в режиме ретранслятора

По умолчанию Postfix передает письма, которые удовлетворяют следующим критериям.

- Отправитель находится в одной из сетей, указанных с помощью переменной $mynetworks. По умолчанию в качестве значения этой переменной заданы адреса сетей, которым принадлежат все сетевые интерфейсы компьютера, в том числе интерфейс localhost.
- Отправитель принадлежит домену, указанному в переменной $relay_domains. По умолчанию значение данной переменной равно значению переменной $mydestination.
- Отправитель пытается передать письмо на компьютер, принадлежащий одному из доменов, указанных в переменной $relay_domains, или их поддоменов.

Конфигурация по умолчанию указывает на то, что Postfix должен обрабатывать почту из того домена, которому принадлежит сам сервер, и от компьютеров, непосредственно связанных с сервером, посредством сетевых интерфейсов. В большинстве случаев такая конфигурация вполне приемлема, но иногда приходится изменять ее. Чтобы сделать это, вам надо изменить значение $mynetworks или $relay_domains (либо модифицировать обе переменные). Предположим, например, что Postfix должен обслуживать рабочую станцию **work.threeroomco.com**. Для этого вам надо переопределить значения переменных следующим образом:

```
mynetworks = 127.0.0.0/8
relay_domains = work.threeroomco.com
```

Возможно, вам потребуется расширить набор компьютеров, обслуживаемых сервером. В этом случае значения переменных могут выглядеть так:

```
mynetworks = 192.168.99.0/24, 172.24.0.0/16, 127.0.0.0/8
relay_domains = $mydestination, pangaea.edu
```

Данные опции сообщают о том, что письма должны приниматься из сетей 192.168.99.0/24, 172.24.0.0/16 и localhost (127.0.0.0/8), а также с компьютеров, принадлежащих доменам $mydestination и pangaea.edu.

Для управления действием mynetworks, relay_domains и некоторых других опций может использоваться опция **smtpd_sender_restrictions**. По умолчанию эта опция отсутсвует в **main.cf**, но при необходимости вы можете включить ее в состав конфигурационного файла. Значение **permit_mx_backup** данной опции соответствует опции **relay_based_on_MX** сервера **sendmail**. Подробные сведения о smtpd_sender_restrictions вы найдете в документации на сервер Postfix.

Настройка Postfix для передачи почты через ретранслятор

В простейшем случае, чтобы сконфигурировать Postfix для передачи почты посредством другого сервера, достаточно установить значение опции relayhost. Эта опция, находящаяся в файле **main.cf**, указывает на компьютер, выполняющий функции ретранслятора. Если в конфигурационном файле сервера имен, управляющего доменом, присутствует запись MX, указывающая на сервер-ретранслятор, то в качестве значения опции relayhost можно задать имя этого домена. Например, если в роли ретранслятора выступает сервер, расположенный на компьютере **franklin.threeroomco.**com, в файл **main.**cf необходимо включить следующую запись:

```
relayhost = franklin.threeroomco.com
```

Если ваш сервер находится в том же домене, что и сервер-ретранслятор, и если на ретранслятор, указывает запись MX, то вместо имени **franklin.threeroomco.com** вы можете использовать переменную **$mydomain**. Такой подход предпочтительнее тем, что переносе почтового сервера, обслуживающего домен, на другой компьютер перенастраивать Postfix не приходится.

В обычных условиях при передаче почты Postfix обращается к серверу DNS. Если же сервер имен в вашей сети отсутствует (например, если преобразование имен осуществляется с помощью файлов /etc/hosts), вам необходимо включить в конфигурационный файл следующую запись:

```
disable_dns_lookups = yes
```

Эта опция указывает серверу Postfix на то, что он не должен обращаться к серверу DNS для преобразования имен. В этом случае Postfix определяет адрес ретранслятора с помощью записи в файле /etc/hosts.

Настройка Postfix для противодействия распространению спама

Подобно sendmail и Exim, Postfix содержит средства, позволяющие бороться с распространением спама. Вы можете блокировать рекламные сообщения, сравнивая информацию в заголовках писем с шаблонами, либо использовать списки IP-адресов.

Инструменты для сравнения с шаблонами, предоставляемые сервером Postfix, достаточно сложны, в частности, они позволяют использовать для анализа содержимого заголовков регулярные выражения. Регулярные выражения часто указываются в отдельном файле, но при желании вы можете задавать их непосредственно в конфигурационном файле main.cf. Пример опции, предназначенной для проверки заголовков, приведен ниже.

```
header_checks = regexp:/etc/postfix/bad_headers
```

В файле bad_headers указываются регулярные выражения, подобные приведенным в листинге 19.2. Если заголовки почтового сообщения соответствуют регулярным выражениям, содержащимся в файле, и если в файле указано, что письмо должно быть отвергнуто, оно возвращается отправителю. Регулярные выражения могут задаваться либо в стиле POSIX (regexp: *описание),* либо в стиле PCRE (pcre: *описание).*

Листинг 19.2. Файл с регулярными выражениями Postfix, используемыми для фильтрации спама

```
#### Поля Subject: заголовков сообщений, полученных от спамеров
/^Subject: ADV:/ REJECT
/^Subject: Accept Visa/ REJECT
#### Поля From: и Received: заголовков сообщений,
#### полученных от спамеров
/^(From|Received):.*badspammer\.net/ REJECT
/^From: spammer@abigisp\ .net/ REJECT
```

Регулярные выражения будут подобно рассматриваться далее в этой главе. Дополнительную информацию о них вы можете получить, обратившись к страницам справочной системы, посвященным программе egrep.

Опция header_checks предоставляет большие возможности, но она сложна в использовании. Более простое решение проблемы спама состоит в применении списков IP-адресов. Для работы с такими списками предназначены две приведенные ниже опции.

```
maps_rbl_domains = relays.mail-abuse.org, dialups.mail-abuse.org
smtpd_client_restrictions = reject_maps_rbl
```

Опция maps_rbl_domains позволяет задавать адреса серверов, управляющих списками IP-адресов (эти серверы описаны в табл. 19.1). В качестве значения данной опции можно указать несколько доменных имен, разделенных запятыми или пробелами. Опция, содержащаяся во второй из приведенных выше строк, указывает на то, что информация, предоставляемая серверами, должна использоваться как основание для блокирования писем. Кроме reject_maps_rbl, опция smtpd_client_restrictions может также принимать другие значения. Например, значение reject_unknown_client сообща-

ет Postfix, что если для адреса отправителя не может быть выполнено обратное DNS-преобразование, письма не должны обрабатываться. Подробнее эти опции описаны в документации на Postfix.

Помимо описанных выше опций, Postfix предоставляет также опции, имеющие лишь косвенное отношение к борьбе со спамом. Некоторые из таких опций описаны ниже.

- **smtpd_helo_required**. По умолчанию для данной опции задается значение no. Если вы измените его на **yes**, Postfix будет обрабатывать письмо только в том случае, если при обмене по протоколу SMTP отправитель передаст команду HELO или EHLO. Этот подход позволяет блокировать действия некорректно написанных программ, часто используемых для распространения спама, но также отвергает обычные письма, отправляемые посредством неправильно сконфигурированного сервера SMTP.

- smtpd_helo_restrictions. Данная опция позволяет Postfix более строго контролировать использование команды HELO или EHLO при SMTP-взаимодействии. Для опции **smtpd_helo_restrictions** предусмотрено несколько значений. Например, **reject_unknown_hostname** означает, что Postfix должен прекратить взаимодействие, если для указанного доменного имени не может быть обнаружена запись A или MX. Значение **reject_non_fqdn_hostname** требует, чтобы отправитель указал полное доменное имя узла. Более подробное описание опции smtpd_helo_restrictions приведено в документации на сервер Postfix.

- **smtpd_sender_restrictions**. Если в конфигурационном файле Postfix указана данная опция, это означает, что информация в поле **From:** заголовка должна соответствовать определенным критериям. Например, значение **reject_unknown_**sender_domain указывает на то, что если в поле **From:** не задано имя узла, письмо должно быть отвергнуто, а значение **reject_non_fqdn_sender** требует, чтобы отправитель включал в адрес полностью определенное доменное имя.

В конфигурационном файле сервера Postfix предусмотрены различные опции, позволяющие настроить сервер для решения разнообразных задач. В большинстве случаев работоспособность Postfix обеспечивает конфигурация, установленная по умолчанию. Задавая слишком строгие ограничения, вы рискуете потерять нужные вам письма.

Меры, предотвращающие использование Postfix для распространения спама, немногим отличаются от соответствующих мер для других серверов. Следует лишь заметить, что конфигурация Postfix по умолчанию предоставляет более свободный доступ к серверу по сравнению с последними версиями **sendmail**, так как Postfix обрабатывает письма из своего домена, а также из сетей, к которым подключен компьютер. Средства, позволяющие ограничить доступ к серверу, рассматривались выше.

Использование фильтров Procmail

Серверы SMTP, описанные в данной главе, могут обрабатывать письма, отправленные с внешних компьютеров. До сих пор мы не рассматривали вопрос о том, что происходит с письмом после того, как оно принимается сервером. В простейшем случае почтовый сервер присоединяет сообщения к файлу, в котором хранятся принятые письма. При необходимости система Linux может быть сконфигурирована так, что в процесс доставки

писем будет включен инструмент Procmail. Procmail позволяет осуществлять сложную обработку писем, принятых почтовым сервером. Вы можете использовать готовые средства фильтрации, поставляемые в составе Procmail, либо создавать собственные фильтры. Изучив принципы работы Procmail, вы можете решать такие задачи, связанные с доставкой почты, которые не могут быть решены другими способами.

Роль Procmail в процессе доставки почты

Большинство почтовых серверов предназначено для передачи сообщений с одного компьютера на другой или от одного пользователя другому, работающему на том же компьютере. В некоторых случаях среда доставки писем оказывается достаточно сложной, и возникает необходимость в фильтрации принятых сообщений. Одна из причин, по которой возникает потребность в фильтрации, — распространение спама — была рассмотрена ранее. Вы, вероятно, захотите отказаться от получения рекламных сообщений или, по крайней мере, собирать их в одной специально предназначенной для них папке. Средства для блокирования спама предусмотрены во всех рассмотренных ранее почтовых серверах, но они уступают по своим возможностям фильтрам, реализуемым посредством Procmail.

Фильтры Procmail позволяют не только отвергать сообщения, как это происходит при блокировании спама. Вы можете использовать фильтры для автоматического распределения писем по папкам. Например, если вы подписаны на несколько списков рассылки, то, вероятно, захотите отделить материалы рассылки от обычных сообщений, помещая их в разные папки. К папкам, содержащим сообщения списков рассылки, нет необходимости обращаться так же часто, как к папкам, в которые поступают обычные письма.

Procmail позволяет даже передавать письма для обработки другим программам. С помощью Procmail вы можете, например, реализовать шлюз, посредством которого принятые письма будут автоматически передаваться от почтового сервера серверу новостей. В результате у вас появится возможность читать письма с помощью программ, предусмотренных для просмотра материалов групп новостей. Чтобы защитить вашу сеть от вирусов и "троянских коней", распространяемых по почте, вы можете организовать обработку поступающих сообщений с помощью антивирусных программ. Эту задачу также позволяет решить Procmail. При необходимости можно создать фильтр, который будет воспроизводить звуковой сигнал, оповещая вас о поступлении писем, содержащих определенные ключевые слова.

Во всех описанных выше примерах используется способность Procmail сканировать сообщения. Например, фильтры для блокирования спама, созданные на основе Procmail, могут искать последовательности символов, специфические для рекламных сообщений. Для этого Procmail использует регулярные выражения, подобные тем, которые применяются сервером Postfix для обработки данных в заголовках сообщений или программой `egrep`.

С помощью Procmail могут создаваться фильтры для всей системы или для отдельных пользователей. Например, фильтры, действующие в пределах системы, могут использоваться для блокирования спама и борьбы с вирусами. Отдельные пользователи могут применять Procmail для распределения сообщений по папкам и выполнения других подобных действий. Конфигурация Procmail для использования в рамках системы задается с помощью файла `/etc/procmailrc`. Для индивидуального применения Procmail настраивается посредством файлов `.procmailrc`, расположенных в рабочих каталогах

пользователей. Файлы, предназначенные для отдельных пользователей, имеют тот же формат, что и файлы, ориентированные на применение во всей системе.

ВНИМАНИЕ

В большинстве версий Linux файл `/etc/procmailrc` используется тогда, когда Procmail запускается по инициативе пользователя `root`. Поэтому надо внимательно следить за тем, чтобы команды, выполняемые Procmail, не нанесли вреда системе. Кроме того, перенаправляя почту, необходимо принимать меры для того, чтобы создаваемые файлы были доступны для чтения пользователям, которым они предназначены. При работе с файлами `.procmailrc`, расположенными в пользовательских каталогах, подробные проблемы не возникают, так как в этом случае Procmail выполняется с привилегиями обычного пользователя.

В конфигурационном файле Procmail содержатся записи трех типов.

- **Комментарии.** Как и во многих других конфигурационных файлах, строки, содержащие комментарии, начинаются с символа #.

- **Записи, определяющие переменные окружения.** В процессе работы Procmail использует значения переменных окружения, например `$HOME` (расположение рабочего каталога пользователя) и `$MAILDIR` (каталог, в котором содержатся пользовательские папки для хранения почтовых сообщений). Значения переменных окружения устанавливаются в конфигурационном файле так же, как и в оболочке. Например, запись `MAILDIR = $HOME/Mail` задает для переменной окружения `$MAILDIR` значение, указывающее на подкаталог Mail, находящийся в рабочем каталоге пользователя.

- **Рецепты.** Правила фильтрации Procmail называются *рецептами* (recipe). Основная работа по построению фильтра сводится к созданию рецепта. Каждый рецепт содержит правила, определяющие обработку сообщения, соответствующего некоторому регулярному выражению. Таким образом, полный набор правил состоит из многих рецептов. Рецепты разделяются на две категории: *рецепты с доставкой* (delivering) и *рецепты без доставки* (nondelivering). Рецепты с доставкой ориентированы на включение сообщения в состав почтового ящика, блокирование сообщения или обработку его с помощью другой программы. Рецепты без доставки определяют вложенные рецепты, т. е. приводят к повторной обработке сообщения с помощью Procmail.

Описанные три типа записей могут располагаться в пределах конфигурационного файла в любой последовательности. Многие конфигурационные файлы Procmail начинаются с определения переменных окружения, за которыми следует набор рецептов. В процессе обработки поступающей почты Procmail сканирует письма и проверяет их на соответствие рецептам. Если письмо не соответствует ни одному рецепту, Procmail доставляет его в файл, определяемый посредством переменной `$DEFAULT`. Обычно это почтовый ящик, используемый по умолчанию, например `/var/spool/mail/`*`имя_пользователя`*.

Создание рецепта

Создание рецепта может показаться очень сложной задачей, в особенности для тех, кто не знаком с регулярными выражениями. Формат рецепта имеет следующий вид:

```
:0 [флаги] [:[файл_блокировки]]
[условия]
действие
```

Рецепт можно условно разбить на три части: идентификационную строку, условия и действие.

Идентификационная строка

Каждый рецепт начинается с символов :0. Цифра 0 не имеет специального значения, и рецептов, начинающихся с :1 или больших номеров, не существует. После :0 вы можете задать один или несколько флагов, которые изменяют поведение Procmail. Наиболее часто используются следующие флаги.

- H. Данный флаг указывает на то, что сравнению с шаблоном должны подвергаться заголовки сообщения. Этот флаг используется по умолчанию
- B. Этот флаг задает сравнение тела сообщения с шаблоном.
- D. По умолчанию при сравнении с шаблоном не учитывается регистр символов. Флаг D отменяет это соглашение.
- c. Данный флаг указывает на то, что рецепт должен работать с "копией" исходного сообщения. Его "оригинал" сохраняется для обработки другими рецептами.
- w. Этот флаг сообщает о том, что Procmail должен ожидать завершения действия, указанного в рецепте. Если действие не окончилось успешно, сообщение остается в очереди для обработки посредством других рецептов.
- W. Данный флаг действует подобно w, но подавляет сообщения об ошибках.

После флагов можно указать двоеточие и имя файла блокировки. Файл блокировки — это специальный файл, который сообщает о том, что в данный момент происходит работа с другим файлом. При наличии файла Procmail откладывает обработку сообщения до тех пор, пока этот файл не будет удален. Файл блокировки удобно использовать в тех случаях, когда в очереди содержится много сообщений; ,без него может возникнуть ситуация, когда сообщения, принятые одно за другим, будут записаны в неверном порядке. По умолчанию имя файла блокировки строится на основе имени файла, в который помещается почта (этот файл указывается в строке действия). Если в строке действия задается обработка сообщения другой программой, вы можете указать имя файла блокировки после двоеточия.

Условия

Условия в составе рецепта состоят из любого (возможно, нулевого) числа строк, обычно начинающихся с символа *. Как правило, в составе условий задаются регулярные выражения — строки символов, с которыми Procmail сравнивает входные данные (заголовок и тело сообщения). Большинство символов используется литерально, но некоторые символы имеют специальные значения. Специальные символы и выполняемые ими действия описаны ниже.

- ^. Указывает на начало строки. Этот символ указывается во многих условиях Procmail после символа *.

- `$`. Данный символ указывает на конец строки.
- `..` Точке соответствует любой символ, кроме символа новой строки. Например, выражению `d.g` удовлетворяют dog, dig, dug и любая другая трехсимвольная последовательность, которая начинается с d и заканчивается д.
- a*. Данному выражению соответствует любое (в том числе нулевое) число символов, указанных перед звездочкой, следующих друг за другом. Очевидно, что вместо a вы можете подставить любой символ. Например, если вам надо найти последовательность, начинающуюся с цифр 802, за которыми следует произвольное количество неизвестных символов, а затем 1618, то сделать это поможет выражение `802.*1618`.
- a+. Это выражение выполняет те же действия, что и `a*`, но количество символов в последовательности не может быть нулевым.
- `a?`. Данное выражение означает, что указанный символ может отсутствовать.
- *`последовательность1 \последовательность2`*. Чтобы указать на то, что в строке может присутствовать одна из двух последовательностей символов, надо разделить эти последовательности символом |. При необходимости вы можете задать выбор более чем из двух альтернативных вариантов, использовав несколько символов |.
- *`(последовательность)`* *. Это выражение похоже на `a*`, но оно означает многократное повторение не одного символа, а целой последовательности.
- `[`*`символы`*`]`. Набор символов, помещенных в квадратные скобки, означает, что в строке должен присутствовать любой из них. Например, выражению `[aeiou]` соответствуют символы a, e, `i`, o или и. Если два символа разделены дефисом (-), они задают диапазон символов. Например, выражению `[m-q]` соответствуют символы т, `n`, *o, p* или q.
- \. Обратная косая черта отменяет специальное значение символа. Например, выражение `\.` соответствует обычной точке.

Дополнительную информацию о регулярных выражениях вы найдете на страницах справочной системы, посвященных Procmail. Объединяя обычный текст и специальные символы, вы можете создавать достаточно сложные выражения. Как было сказано ранее, условия в составе рецепта могут занимать одну или несколько строк. В большинстве случаев используются условия, состоящие из одной строки. Если условия занимают несколько строк, письмо соответствует рецепту в том случае, если оно соответствует каждому из условий. Если условия отсутствуют, рецепту соответствует любое сообщение.

В составе условий могут быть использованы дополнительные символы, указывающие на то, что рецепт должен быть интерпретирован специальным образом. Некоторые из них описаны ниже.

- `!`. Данный символ инвертирует результат сравнения. Если условие начинается с символа !, то, для того, чтобы письмо соответствовало рецепту, оно не должно соответствовать данному условию. Например, вы можете создать рецепт, которому соответствуют все сообщения, кроме адресованных пользователю postmaster.

- <. Условие применяется в том случае, если длина сообщения меньше указанного числа байтов.
- >. Условие применяется в том случае, если длина сообщения больше указанного числа байтов.

Действие

Действие в составе рецепта занимает одну строку и указывает Procmail, как следует обрабатывать сообщение. Простое действие лишь задает имя файла, в который Procmail должен поместить сообщение. Действия Procmail хорошо сочетаются с sendmail, Exim, Postfix и другими серверами, использующими формат mbox. Если же вы работаете с qmail или другим сервером, поддерживающим формат maildir, описание действия Procmail необходимо завершать косой чертой (/), которая указывает на то, что Procmail должен сохранить сообщение в формате maildir. Procmail также поддерживает еще один формат хранения сообщений, для использования которого описание действия должно заканчиваться косой чертой и точкой.

Помимо записи писем в папки, Procmail также может выполнять другие действия, для описания которых в начале строки указываются перечисленные ниже символы.

- !. Если описание действия начинается с восклицательного знака, Procmail интерпретирует содержимое строки как список почтовых адресов, по которым следует перенаправить сообщение. Вы можете использовать данную возможность для автоматического создания сообщений, предназначенных для группы пользователей.
- |. В оболочках UNIX вертикальная черта используется для организации конвейерной обработки данных. В Procmail данный символ имеет аналогичное назначение. Если описание действия начинается с вертикальной черты, Procmail запускает указанную программу и передает ей сообщение для обработки. Вы можете использовать данную возможность для выполнения более сложных действий над сообщениями.
- {. Открывающая фигурная скобка является признаком начала блока. В состав блока могут входить рецепты, которые применяются только к сообщениям, соответствующим условиям включающего рецепта. (Включающий рецепт является рецептом без доставки. Если сообщение не соответствует ни одному из включаемых рецептов, оно не доставляется.) Такая возможность может использоваться в том случае, если у вас есть несколько рецептов и вы хотите применять их только при выполнении некоторых предварительных условий. Например, вложенные рецепты можно использовать для распознавания рекламных сообщений по некоторым признакам, каждый из которых не позволяет принять окончательное решение о типе письма. Признаком окончания блока является закрывающая фигурная скобка.

В каждом из рецептов может содержаться лишь одно действие. Если вы хотите, чтобы над сообщением выполнялось несколько операций, вам следует создать сценарий и передать ему сообщение для обработки. При этом вам необходимо следить за тем, чтобы сценарий прочитал все сообщение, в противном случае Procmail проверит сообщение с помощью других правил. В некоторых случаях в составе рецепта задается флаг c, указывающий на то, что операции должны выполняться над копией сообщения. При этом в зависимости от действий, производимых над сообщением, оно может быть доставлено несколько раз.

Пример использования рецептов

Приведенные выше сведения были необходимы для создания общего представления о работе Procmail. В листинге 19.3 приведен чрезвычайно простой пример файла Procmail, предназначенного для фильтрации сообщений. Содержащиеся в нем рецепты пригодны для пользовательского файла `.procmailrc`, поскольку они предусматривают доставку сообщений в папку рабочего каталога пользователя.

Листинг 19.3. Пример конфигурационного файла Procmail

```
MAILDIR = $HOME/Mail

# Поиск рекламных сообщений. Проверка не затрагивает письма,
t адресованные пользователю postmaster или отправленные им
:0
*! (From|To) : . *postmaster
{
    :0 B
    *  .*301.*S.*1618
    /dev/null

    :0
    * From: .*badspammer\.net
    /dev/null

    :0
    *  Subject:.*\$\$\$
    /dev/null
}

# Проверка по ключевым словам rug и david и
# перенаправление писем ату
:0 c
* From: . *david@pangaea\.edu
* Subject: . *rug
! amy@threeroomco.com

# Сообщения списков рассылки помещаются в отдельную папку
:0:
* To: . *list@mailinglist\.example\.com
$MAILDIR/mailinglist
```

Листинг 19.3 иллюстрирует некоторые важные особенности рецептов Procmail.

- **Вложенные рецепты.** Рецепты, выполняющие блокировку спама, содержатся в составе другого рецепта, в результате чего эти фильтры применяются только для тех сообщений, которые адресованы пользователям, отличным от postmaster. (Это достигается посредством оператора отрицания, указанного в условиях вклю-

чающего рецепта.) Аналогичный результат можно получить, включив условие `*! \ To: .*postmaster` в состав каждого из фильтров, предназначенных для блокирования спама. В данном простом примере это может несколько сократить объем конфигурационного файла. В более сложных фильтрах при использовании вложенных рецептов объем файла уменьшается. Кроме того, применение вложенных фильтров уменьшает вероятность ошибки, так как некоторые условия при этом указываются однократно.

- **Регулярные выражения.** В листинге 19.3 содержатся три рецепта, предназначенные для фильтрации рекламных сообщений. Первый из них проверяет тело сообщения (на это указывает флаг B) на наличие строки, содержащей последовательности 301, S и 1618. Этот рецепт предназначен для перехвата писем, содержащих указание на раздел 301 и номер S.1618, которые часто используются спамерами для создания иллюзии официального сообщения. Второй из рецептов, предназначенных для фильтрации спама, блокирует все письма из домена `badspammer.net`, а третий фильтр блокирует сообщения, содержащие в поле `Subject:` последовательность $. Обратите внимание на использование обратной косой черты для отмены специального значения символов. Все три рассматриваемых здесь рецепта направляют сообщения в файл `/dev/null`, т. е. удаляют их. После копирования в файл `/dev/null` письма уже не могут быть восстановлены. Файл блокировки для этих рецептов не требуется, так как сообщения не сохраняются ни в одной папке.

- **Копирование сообщений.** Вместо того чтобы записывать сообщение в файл, второй рецепт, приведенный в рассматриваемом примере, передает его другому пользователю. На это указывают флаг c и восклицательный знак в начале описания действия. Сообщение должно удовлетворять двум критериям: отправителем его должен быть пользователь `david@pangaea.edu`, и оно должно содержать слово rug в поле заголовка Subj`ect:`. Если хотя бы одно из условий не выполняется, сообщение не копируется.

- **Сортировка сообщений.** Последний рецепт распределяет сообщения по папкам. Письма, адресованные `list@mailinglist.example.com`, помещаются в отдельную папку, расположенную в подкаталоге рабочего каталога пользователя, предназначенного для работы с почтой. Во многих списках рассылки поле `To:` заголовка сообщения используется для идентификации самого списка, а информация о получателе сообщения включается в поле To заголовка конверта. Для того чтобы выбрать наиболее удобный способ распределения писем по папкам, вам следует выяснить, какие данные содержатся в заголовках писем, распространяемых посредством списков рассылки.

Рецепты, приведенные в листинге 19.3, предельно просты, и их вряд ли можно использовать для решения конкретных задач, однако на их основе вы можете создать реальные рецепты. Кроме того, при необходимости вы можете внести изменения в фильтры, полученные из других источников.

Использование существующих наборов фильтров

Создание фильтров Procmail — достаточно сложная задача, отнимающая много сил и времени. Вместо того чтобы заниматься созданием фильтра с нуля, вы можете попы-

таться применить для своих целей готовые фильтры. Некоторые из источников фильтров Procmail описаны ниже.

- **SpamBouncer.** Этот пакет представляет собой набор фильтров Procmail, предназначенных для блокирования спама. Фильтры, входящие в состав SpamBouncer, достаточно сложны, но при необходимости вы можете адаптировать некоторые из них для решения собственных задач. Подробно эти фильтры описаны в документации, поставляемой в составе пакета. Чтобы скопировать SpamBouncer, надо обратиться на его Web-страницу, расположенную по адресу `http://www.spambouncer.org`.

- **SmartList.** Этот пакет, реализующий список рассылки, создан на основе Procmail. Дополнительную информацию о нем вы можете получить из документа SmartList FAQ, доступного по адресу `http://www.hartzler.net/smartlist/SmartList-FAQ.html`.

- **Советы и рецепты Тимо**. Тимо Салми (Timo Salmi) поддерживает Web-страницу (`http://www.uwasa.fi/~ts/info/proctips.html`),посредством которой он распространяет информацию о простых рецептах Procmail. Информацию, представленную на этой странице, нельзя рассматривать как готовый к использованию пакет, такой как SpamBouncer или SmartList, однако вы можете найти на ней "заготовки" для своих фильтров.

- **Примеры рецептов Procmail с комментариями.** Web-узел `http://handsonhowto.com/pmail102.html` содержит примеры рецептов Procmail, снабженные комментариями, которые поясняют их работу.

Дополнительную информацию о фильтрах и рецептах, пригодных для использования, вы получите, выполнив в Internet поиск по ключевым словам Procmail recipes. Многие полезные ссылки можно найти на Web-узле Procmail по адресу `http://www.procmail.org`.

Простые наборы фильтров можно разместить в рабочем каталоге пользователя в файле `.procmailrc`. Если фильтр должен воздействовать на систему в целом, его надо включить в файл `/etc/procmailrc`. Некоторые пакеты, например SpamBouncer, содержат специальные файлы, поэтому при инсталляции необходимо следить, чтобы они были установлены корректно.

ВНИМАНИЕ | В ряде случаев, чтобы выполнить задачу, недостаточно инсталлировать фильтр. Некоторые фильтры приходится настраивать, указывая в них имена у шов и даже имена пользователей. Разработчики многих фильтров ориентировались на потребности конкретных администраторов, в то время как перед вами могут стоять несколько другие задачи.

СОВЕТ ✔ Как системный администратор, вы можете позволить себе роскошь создать специальную учетную запись для тестирования. Такую запись удобно использовать для проверки фильтров Procmail. Пробные сообщения можно передавать, изменив настройку программы подготовки писем и даже непосредственно взаимодействуя с сервером SMTP посредством программы `telnet` (для этого надо при установлении соединения указать порт 25).

Запуск Procmail

При обсуждении принципов фильтрации предполагалось, что программа Procmail уже запущена и обрабатывает поступающие сообщения. В большинстве версий Linux почтовые серверы уже сконфигурированы для использования Procmail при доставке почты. Действия, необходимые для настройки серверов, описаны ниже.

- `sendmail`. Чтобы настроить сервер для использования Procmail, необходимо включить в конфигурационный файл `m4` три записи. Первая из них,

  ```
  define('PROCMAIL_MAILER_PATH', '/usr/bin/procmail'),
  ```

  сообщает sendmail о том, где расположены двоичные файлы Procmail. Записи `FEATURE(local_procmail)` и `MAILER(procmail)` указывают sendmail на необходимость использования Procmail при доставке почты.

- Exim. Около двух третей конфигурационного файла `exim.conf`, используемого по умолчанию, занимает раздел `procmail_pipe`. Этот раздел посвящен использованию Procmail для доставки почты. Убедитесь, что данный раздел присутствует в конфигурационном файле и что в нем указан требуемый двоичный файл.

- `Postfix`. В конфигурационном файле `main.cf`, используемом по умолчанию, для вызова Procmail применяется опция `mailbox_command`. Если вы не укажете эту опцию, Postfix будет доставлять почту, минуя Procmail.

Во многих версиях Linux почтовые серверы по умолчанию настроены для взаимодействия с Procmail. При работе с этими версиями вам не потребуется изменять конфигурацию сервера. Если же на вашем компьютере Procmail по умолчанию не применяется, но вы хотите использовать этот инструмент при доставке почты некоторым пользователям, вам надо создать в рабочих каталогах этих пользователей файл с именем `.forward`, содержащий следующую строку:

```
"|IFS=' '&&p=/usr/bin/procmail&&test -f $p&&exec $p \
-Yf-||exit 75 #имя_пользователя"
```

Следите за тем, чтобы одинарные и двойные кавычки были указаны точно так, как в этом примере.

Резюме

Электронная почта — одна из наиболее важных служб Internet. Linux обеспечивает работу самых различных почтовых серверов, как всем известного `sendmail`, так и новых продуктов, например Exim и Postfix. Независимо от того, какой сервер SMTP вы используете, можете настроить его для выполнения некоторых стандартных действий, скажем, для получения писем и перенаправления их другим системам. При настройке сервера необходимо уделять внимание различным деталям, например, указывать набор компьютеров, с которых сервер должен получать письма для передачи другим системам, принимать меры для блокировки спама и т. д. После получения сообщения сервером SMTP оно может быть обработано с помощью Procmail. Procmail позволяет отвергать рекламные сообщения, передавать письма другим программам, пересылать копии сообщений другим пользователям и выполнять прочие, самые разнообразные задачи. Procmail обеспечивает большую гибкость в работе, в частности, правила фильтрации может создавать не только системный администратор, но и любой пользователь.

Глава 20

Поддержка Web-сервера

Многие пользователи не видят различий между системой World Wide Web и Internet в целом, хотя каждый специалист знает, что функционирование Internet обеспечивается большим количеством серверов, поддерживающих различные протоколы. Если же говорить о наиболее популярной службе Internet, то ею, несомненно, окажется Web. Наличие Web-сервера — одно из условий успешной работы практически каждой организации. Web-сервер, по сути, является "лицом" компании в среде Internet.

В системе Linux работа Web-сервера может обеспечиваться различными программами, однако наибольшей популярностью среди администраторов пользуется продукт Apache. По этой причине в данной главе основное внимание уделяется этому серверу. Кроме того, здесь рассматриваются Web-серверы, выполняющиеся как процессы ядра, формы, сценарии, защита при обмене данными, виртуальные домены и прочие вопросы, связанные с работой Web-сервера. В этой главе также обсуждаются подготовка материалов для представления на Web-узле и анализ трафика, связанного с работой Web-сервера.

Организовать выполнение основных функций Web-сервера в системе Linux не трудно, но разобраться с опциями, предназначенными для поддержки расширенных возможностей сервера, достаточно сложно. Описать эти опции в одной главе невозможно. Если вам потребуется подробная информация о работе Web-сервера, вам следует обратиться к документации на программный продукт, который вы собираетесь использовать, или к изданиям, специально посвященным этому вопросу: Энгельскелла (Engelschall) *Apache Desktop Reference* (Addison Wesley, 2001) или Олда (Auld) *Linux Apache Server Administration* (Sybex, 2001). Есть также книги, в которых описываются специальные функции Web-сервера. В качестве примера можно привести книгу Мелцера (Meltzer) и Микалски (Michalski), *Writing CGI Applications with Perl* (Addison Wesley, 2001).

Использование Web-сервера

Несмотря на то что Web-серверы очень важны для различных организаций, Web-сервер не обязательно должен присутствовать на каждом компьютере и даже в каждой сети. Более того, установка сервера, в котором нет необходимости, приведет к неоправданным затратам времени и усилий и даже может создать угрозу безопасности системы.

Web-сервер — это программа, реализующая обмен данными посредством HTTP (Hypertext Transfer Protocol — протокол передачи гипертекстовой информации). Web-сервер принимает запросы от клиентов через некоторый порт (как правило, это порт с номером 80). HTTP-клиент (обычно его называют *Web-броузером)* передает Web-серверу запрос на получение документа. Получив запрос, Web-сервер читает документ с жесткого диска, или, если запрос предполагает вызов **CGI-сценария,** получает документ, сгенерированный этим сценарием. Затем документ передается клиенту. В протоколе HTTP предусмотрена также возможность передачи Web-серверу данных для обработки.

Web-сервер используется для организации *Web-узла* — набора документов, к которым пользователи могут обращаться, указывая *URL* (Uniform Resource Locator — унифицированный локатор ресурса). (Следует различать понятия Web-узел и узел сети. Если Web-узел представляет собой набор документов, то узел сети — это лишь компьютер или другое устройство, подключенное к сети.) Чаще всего URL начинаются с `http://`, но в некоторых случаях в начале URL указываются другие протоколы, например `ftp://`. Web-серверы поддерживают `http://` и `https://`; другие протоколы поддерживаются другими серверами.

У читателя может возникнуть вопрос о том, как связаны между собой понятия Web-сервер и Web-узел. Если вы установите Web-сервер и разместите на компьютере документы, которые этот сервер будет предоставлять внешним пользователям, вы получите Web-узел. Такой сервер необходим для большинства коммерческих организаций и даже для частных лиц. Сервер предоставляет возможность взаимодействовать с потребителями и деловыми партнерами: с каждым, кому может потребоваться информация об организации, услугах и о продуктах.

Web-серверы часто используются для обеспечения внутреннего взаимодействия. Вам может понадобиться Web-узел, который будет доступен только в пределах локальной сети. На этом узле можно разместить график работы над проектом, расписание совещаний и другую информацию, предназначенную только для сотрудников организации. Внутренний Web-сервер и Web-сервер, предназначенный для обслуживания внешних пользователей, надо разместить на разных компьютерах. Если же они должны находиться на одной машине, вам придется организовать виртуальные узлы, которые будут рассматриваться далее в этой главе.

Следует хорошо представлять себе различия между Web-узлом и Web-сервером. Web-узел, как было сказано выше, — это набор документов, представленных в Web, при обращении к которым пользователи указывают URL. В этих URL содержится доменное имя, которое часто отражает название организации. В отличие от Web-узла, Web-сервер — это набор программных или аппаратных средств, обеспечивающих поддержку Web-узла. При желании можно организовать работу Web-узла, не устанавливая в локальной сети Web-сервер. Для этого надо воспользоваться услугами сторонней организации, предоставляющей дисковое пространство своего Web-сервера для размещения на нем требуемых документов и программ. Как вы узнаете из данной главы, Web-сервер можно настроить так, чтобы его действия зависели от имени, указанного в запросе. Чтобы при указании имени, принадлежащего вашему домену, обращение осуществлялось к внешнему серверу, на котором размещены ваши документы, вам надо специальным образом настроить сервер DNS (настройка сервера DNS описывалось в главе 18). Например, если вы разместили документы на Web-сервере по адресу 10.102.201.1 и хотите связать с этим сервером имя `www` в вашем домене, вам надо включить в конфигурационный файл DNS следующую запись:

```
www        IN A        10.102.201.1
```

Сервер, на котором размещаются ваши документы, необходимо настроить для обработки обращений по указанному вами адресу. Кроме того, вам необходимо иметь доступ к внешнему компьютеру, на котором выполняется Web-сервер, для того, чтобы записать данные на его диск.

Размещение данных на внешнем сервере имеет ряд преимуществ по сравнению с поддержкой собственного Web-сервера. Например, пропускная способность линии, посредством которой ваша локальная сеть подключена к Internet, может быть недостаточной для организации обмена данными с Web-сервером. (Так, например, линия с пропускной способностью 200 Кбод, которая может находиться в нерабочем состоянии до пяти часов в месяц, хороша для домашнего использования, но не подходит для большой компании, такой как ЮМ.) Размещая данные на сервере другой организации (например, на сервере **провайдера),** вы избавлены от необходимости поддерживать собственный Web-сервер. Недостатком такого подхода является необходимость платить деньги за аренду дискового пространства на сервере (от нескольких долларов до нескольких тысяч долларов в месяц, в зависимости от объема данных и трафика). Кроме того, некоторые внешние серверы не обеспечивают необходимых ресурсов. Например, может оказаться, что сервер не поддерживает CGI или SSL.

Существуют также другие способы размещения Web-узла за пределами вашей локальной сети. Один из таких способов состоит в том, что вы располагаете компьютер с вашим Web-сервером в локальной сети другой организации, обычно в сети провайдера. При этом сервер обменивается данными с клиентами по надежной линии с высокой пропускной способностью, и в то же время вы получаете возможность настроить сервер в соответствии со своими потребностями. Еще одно решение состоит в использовании Web-страниц, предоставляемых по умолчанию вместе с некоторыми учетными записями. В этом случае в составе URL вместо имени вашей организации будет указано имя провайдера, например `http://www.abigisp.net/~`*`имя`*`/`. Такой подход приемлем для частных лиц и небольших организаций. Однако при этом возможности по представлению данных в Web чрезвычайно ограничены, кроме того, для коммерческой организации нежелательно, чтобы в составе URL указывалось имя провайдера.

Несмотря на наличие альтернативных решений, вам все же необходимо рассмотреть целесообразность инсталляции Web-сервера в локальной сети. Если Web-сервер нужен вам для внутреннего взаимодействия, его, несомненно, придется установить. При этом необходимо уделить должное внимание его настройке: конфигурация, заданная по умолчанию, в этом случае вряд ли подойдет вам. При инсталляции некоторых версий Linux Web-сервер устанавливается по умолчанию. Если Web-сервер на этом компьютере не нужен, для его поддержки будут напрасно расходоваться ресурсы. Кроме того, неиспользуемый Web-сервер нежелателен с точки зрения безопасности. С другой стороны, в некоторых версиях Linux Web-сервер применяется для предоставления пользователям справочных данных, поэтому его присутствие оправдано даже на рабочей станции. Общие правила таковы: если без Web-сервера можно обойтись, его не следует устанавливать. Web-сервер необходимо инсталлировать только на том компьютере, который используется для организации работы Web-узла.

Программы, реализующие Web-сервер в системе Linux

В настоящее время существует несколько программных продуктов, позволяющих обеспечить функционирование Web-сервера в системе Linux. Некоторые программы имеют небольшой размер и поддерживают лишь ограниченный набор возможностей, другие представляют собой большие пакеты и позволяют реализовать разнообразные, даже самые "экзотические", функции. Ниже описаны наиболее популярные Web-серверы, предназначенные для Linux.

- **Apache.** Этот продукт поставляется в составе каждого дистрибутивного пакета Linux. Как правило, процедура установки системы запрашивает пользователя, следует ли инсталлировать сервер Apache. По данным Netcraft (`http://www.netcraft.com`), в марте 2002 г. 65% всех работающий в Internet Web-серверов составляли серверы Apache. По этой причине основное, внимание *в* данной главе уделяется данному продукту. Apache представляет собой полнофункциональный Web-сервер и реализует расширенные возможности, например поддерживает сценарии CGI и SSL-взаимодействие. Web-узел Apache расположен по адресу `http://httpd.apache.org`.

- **Roxen.** Этот продукт также представляет собой полнофункциональный Web-сервер; во многом он напоминает Apache. Его настройка осуществляется посредством Web-интерфейса, что привлекает некоторых начинающих администраторов. Дополнительную информацию о Roxen можно получить, обратившись по адресу `http://www.roxen.com/products/webserver/`.

- `thttpd`. Данный сервер отличается небольшим размером кода. Если объем Apache составляет около 300 Кбайт (в зависимости от набора используемых компонентов эта цифра может изменяться), то объем `thttpd` — всего 50 Кбайт. Данный сервер работает быстро и эффективно. Несмотря на размер, он поддерживает сценарии CGI, но не обеспечивает SSL-взаимодействие. Более подробные сведения об этом сервере можно получить по адресу `http://www.acme.com/software/thttpd/thttpd.html`.

- Zeus. Большинство Web-серверов, предназначенных для работы в системе Linux, бесплатно распространяются в исходных кодах, но Zeus является исключением. Это коммерческий продукт; его цена составляет 1700 долларов. Согласно информации, опубликованной на Web-узле Zeus (`http://www.zeus.co.uk/products/zws/`), данный сервер обеспечивает лучшую масштабируемость по сравнению с другими серверами. Это проявляется при интенсивных обращениях клиентов к Web-серверу.

- **Web-серверы на базе ядра.** Существуют Web-серверы, которые выполняются как процессы ядра Linux. Дело в том, что действия по предоставлению Web-страниц пользователям в основном сводятся к обращению к дискам и обмену данными через сетевое соединение. Большинство подобных задач могут решаться непосредственно ядром системы, причем выполняются они гораздо эффективнее, чем это происходит при использовании внешних программ. Подобные серверы будут подробно рассмотрены ниже.

- Нетрадиционные серверы. Некоторые программы используют протокол HTTP для выполнения специальных действий, не поддерживаемых обычными Web-серверами. Например, инструменты удаленного администрирования, которые рассматривались в главе 16, формально могут считаться Web-серверами. Взаимодействие с ними можно организовать посредством обычных Web-броузеров, но они принимают обращения от клиентов через порт, отличный от порта 80. Подобные серверы в данной главе рассматриваться не будут.

Если вам необходимо обеспечить выполнение специфических функций, ознакомьтесь с возможностями различных серверов. На сегодняшний день существует так много продуктов, поддерживающих протокол HTTP, что вы почти наверняка найдете подходящую для вас программу.

Как правило, требованиям большинства администраторов вполне удовлетворяет сервер Apache, поставляемый в комплекте со всеми версиями Linux. Если по каким-то причинам вам необходимо минимизировать объем памяти, занимаемый сервером, рассмотрите возможность использования сервера thttpd. Если вам не нужны расширенные возможности, предоставляемые Apache, то имеет смысл выбрать тот сервер, для инсталляции, настройки и поддержки которого требуется меньше усилий. Однако благодаря популярности Apache процедура его настройки многократно проверена, и при этом обычно не возникает проблем, потому большинство администраторов выбирают данный сервер для решения своих задач.

Если вам необходим максимально производительный сервер, это обеспечит сервер, использующий функции ядра, например kHTTPd. Этот продукт позволяет обработать больше запросов, не увеличивая объем ресурсов компьютера. Повысить эффективность обработки запросов позволяют также серверы thttpd и Zeus. Следует заметить, что в подавляющем большинстве случаев производительность сервера ограничивается не ресурсами компьютера, а пропускной способностью линии связи. Если число обращений к вашему серверу превышает возможности линии, установка более эффективно работающих программ не решит проблему. В этом случае вам надо искать способы уменьшения нагрузки на сеть (например, за счет сокращения объема графических материалов), увеличения пропускной способности линии либо рассмотреть возможность размещения Web-сервера в другой сети.

Несмотря на то что в данной главе основное внимание уделяется Apache, материал, изложенный здесь, поможет вам и при работе с другими продуктами, так как принципы конфигурирования различных Web-серверов часто совпадают. Поэтому, зная особенности настройки Apache, для перехода на другой сервер вам достаточно будет изучить структуру его конфигурационного файла.

Настройка основных функций Apache

Независимо от того, какие задачи должен решать ваш Web-сервер, конфигурирование его надо начать с настройки базовых функций Apache. Лишь после того, как сервер сможет предоставить клиентам статические документы (т. е. документы, не предполагающие использование сценариев), вы сможете приступить к созданию конфигурации, ориентированной на поддержку расширенных возможностей. Первоначальная настройка Apache сводится к установке значений нескольких основных опций конфигурационного файла. Кроме того, вам необходимо иметь хотя бы общее представление о модулях

Apache, которые представляют собой расширения, предназначенные для решения специфических задач. В большинстве дистрибутивных пакетов сервер Apache по умолчанию настроен так, что для обеспечения его работы достаточно внести в конфигурационные файлы лишь незначительные изменения.

Конфигурационные файлы Apache

В большинстве пакетов основной конфигурационный файл Apache носит имя **httpd.conf**. В зависимости от версии системы этот файл может находиться в разных каталогах, но формат его остается неизменным. В системах Caldera и SuSE файл httpd.conf содержится в каталоге /etc/httpd; в **Debian** и **Slackware** он размещается в /etc/apache (**Slackware** предоставляет файл-образец /etc/apache/httpd.**conf.default**; для обеспечения работы сервера надо лишь переименовать данный файл и внести в него необходимые изменения); в Red Hat и **TurboLinux** файл **httpd.conf** размещается в каталоге **/etc/httpd/conf/**.

Как обычно, строки файла httpd.conf, начинающиеся с символа #, содержат комментарии. Опции, определяющие конфигурацию сервера, задаются в следующем виде:

Директива *Значение*

Директива — это имя, с которым может быть связано некоторое значение. Значением может быть число, имя файла или произвольная строка символов. Некоторые директивы позволяют задавать несколько подопции. В этом случае имя директивы помещается в угловые скобки. Пример подобной директивы приведен ниже.

```
<Directory /home/httpd/html>
    Options FollowSymLinks
    AllowOverride None
</Directory>
```

В последней строке содержится имя той же директивы, которая указана в начале, но для нее не задается никакое значение. Имени директивы, завершающей блок, предшествует косая черта.

В некоторых случаях для настройки Apache используются дополнительные конфигурационные файлы, перечисленные ниже. Обычно они размещаются в том же каталоге, что и **httpd.conf**.

- **access.conf**. Ссылка на этот файл формируется с помощью директивы AccessConfig и содержится в файле **httpd.conf**. В файле **access.conf** чаще всего задаются директивы <Directory>, определяющие особенности доступа к указанным в них каталогам. В настоящее время этот файл обычно остается пустым, а иногда в качестве значения **AccessConfig** задается **/dev/null**, что запрещает использование **access.conf**.

- **mime.types**. Для того чтобы сообщить Web-броузеру о том, как должны обрабатываться данные, Web-сервер использует стандарт MIME (Multipurpose Internet Mail Extensions — многоцелевые почтовые расширений Internet). Например, MIME-тип text/plain означает, что данные представляют собой обычный текст, а **image/jpeg** определяет графические данные в формате JPEG (Joint Photographic Experts Group — объединенная группа экспертов по обработке фотоснимков). Файл **mime.types** содержит информацию о соответствии между MIME-типами и расши-

рениями файлов. Например, имена файлов, оканчивающиеся `.txt` и `.asc`, связываются с MIME-типом `text/plain`. Если такое соответствие задано неправильно, Web-броузер будет испытывать затруднения при обработке некоторых типов файлов. Файл, поставляемый в составе пакета, обеспечивает обработку практически любых типов данных, которые могут быть помещены на Web-страницу. Если же вам надо использовать редко встречающиеся типы, вам придется добавить в этот файл новые записи.

- magic. Этот файл также позволяет определять соответствие между MIME-типами и данными. При анализе информации можно обнаружить специфические признаки того или иного типа. Так, например, многие файлы содержат специальные ключи — "магические" байтовые последовательности. Эти последовательности, преобразованные в текстовый вид, указываются в файле magic. Если вы подробно не изучили формат этого файла, вносить изменения в него не рекомендуется. Структура файла magic в данной главе рассматриваться не будет.

Способы запуска сервера Apache

В главе 4 были описаны различные способы запуска серверов на выполнение. Apache может быть запущен любым из этих способов: с помощью суперсервера, сценария запуска SysV либо локального сценария. В большинстве дистрибутивных пакетов предусмотрен запуск сервера с помощью сценария SysV или локального сценария, так как эти способы обеспечивают постоянное присутствие сервера в памяти и, следовательно, уменьшают задержку при генерации ответа на запрос клиента. При необходимости вы можете также обеспечить запуск Apache посредством суперсервера; программа инсталляции системы Debian даже задает вопрос о том, каким способом должен запускаться сервер. Однако при использовании суперсервера скорость обработки запросов снизится, так как, получив запрос, суперсервер должен будет загрузить Apache.

СОВЕТ ✔ Если по каким-либо причинам, например по соображениям безопасности, вам придется организовать запуск Web-сервера с помощью суперсервера, то вместо Apache желательно использовать программу, которая занимает меньше места в памяти и, следовательно, быстрее загружается. Так, например, вы можете установить в системе `thttpd` или Web-сервер, выполняющийся как процесс ядра.

Если вы собираетесь изменить способ запуска Apache, вам следует скорректировать значение опции `ServerType`. Эта опция находится в конфигурационном файле Apache и может принимать значение `standalone` или `inetd`. Если вы неправильно укажете значение этой опции, Apache будет работать некорректно либо вовсе не будет обрабатывать запросы. Так, например, если вы захотите отказаться от сценария SysV и перейти к запуску Apache посредством `inetd`, вам следует сначала внести изменения в конфигурационный файл суперсервера, затем с помощью сценария SysV завершить выполнение Apache, запретить использование сценария SysV, потом отредактировать файл `/etc/inetd.conf` и, наконец, перезапустить `inetd`. Если вы забудете выполнить хотя бы одно из описанных здесь действий, сервер будет работать некорректно или продолжит работу с использованием старой конфигурации.

В некоторых пакетах исполняемый файл Apache называется apache, в других — httpd. Если вы собираетесь изменить сценарий запуска или завершить работу сервера, необходимо правильно указать имя программы.

Опции общего назначения

Конфигурация, устанавливаемая по умолчанию, во многих случаях обеспечивает работоспособность сервера. После инсталляции сервера и его запуска Apache готов предоставить пользователям файлы из каталога по умолчанию (обычно это каталог /home/httpd/html). В этот каталог при установке Apache помещаются файлы, содержащие в основном информацию о том, что сервер инсталлирован, но настройка его еще не закончена. Впоследствии вы, вероятно, замените их теми файлами, которые и будут составлять содержимое Web-узла.

Ниже описаны опции общего назначения, определяющие поведение Apache. Настройка сервера выполняется путем изменения их значений.

- ServerType. Эта директива уже рассматривалась ранее. Она может принимать значение standalone или inetd.

- User и Group. В системе Linux каждый сервер запускается от имени конкретного пользователя и группы. Эти директивы позволяют указать пользователя и группу, с полномочиями которых будет выполняться сервер Apache. В большинстве дистрибутивных пакетов Apache запускается от имени пользователя nobody либо с помощью учетной записи, специально созданной для данной цели и предусматривающей минимальные привилегии пользователя. Такой подход снижает вероятность того, что злоумышленник сможет воспользоваться недостатками в защите сервера для незаконного проникновения в систему. Рекомендуется принять значения этих опций, установленные при инсталляции системы.

В целях повышения безопасности системы большинство двоичных файлов Apache скомпилированы так, чтобы их нельзя было запустить от имени пользователя root.

- ServerTokens. Сервер Apache предоставляет клиентской программе информацию о платформе, на которой он выполняется. В большинстве пакетов по умолчанию для этой опции задается значение ProductOnly, которое запрещает передавать клиенту сведения о системе. При желании вы можете задать значение Min, OS или Full (эти значения расположены в порядке возрастания объема информации, передаваемой клиенту), но в целях повышения безопасности рекомендуется принять значение ProductOnly, установленное при инсталляции.

ВНИМАНИЕ | Не следует считать, что, установив значение ProductOnly опции ServerTokens, вы лишите взломщика возможности получить данные о системе. Он по-прежнему может анализировать трафик и не только выяснить тот факт, что вы используете Linux, но и узнать версию системы. Кроме того, сведения о платформе могут предоставлять другие серверы.

- `MinSpareServers` и `MaxSpareServers`. Если Apache должен постоянно присутствовать в сети, для более эффективного обслуживания клиентских запросов в системе обычно запускается несколько экземпляров сервера. Каждый экземпляр обрабатывает отдельный запрос. Директивы MinSpareServers и MaxSpareServers позволяют задать минимальное и максимальное число экземпляров сервера, не участвующих в обработке запросов. Если число экземпляров сервера меньше, чем значение директивы MinSpareServers, то даже если они не выполняют обработку запросов, главный процесс Apache порождает новые процессы. Аналогично, если число неиспользуемых экземпляров сервера становится больше, чем значение директивы MaxSpareServers, лишние процессы завершаются. Если значения этих директив слишком малы, то в моменты интенсивных обращений к серверу время, необходимое для получения ответа на запрос, будет увеличиваться. Если же значения директив окажутся слишком большими, то памяти компьютера может оказаться недостаточно для размещения процессов сервера. При установке сервера по умолчанию задаются значения 5 и 10. Если нагрузка на сервер небольшая, вы можете уменьшить значения MinSpareServers и MaxSpareServers и проверить, как система отреагирует на это. Если же клиенты часто обращаются к серверу, вам, возможно, потребуются более высокие значения данных директив. Заметьте, что в некоторый момент времени общее число процессов Apache может превысить значение MaxSpareServers, так как некоторые из них заняты обработкой запросов клиентов и не рассматриваются как свободные. Если количество запросов к серверу велико, для поддержки всех экземпляров сервера может потребоваться большой объем области подкачки. При больших значениях MaxSpareServers требования к памяти, а следовательно, и к объему области подкачки еще более возрастают.

- MaxClients. Данная директива задает максимальное количество клиентов, которые могут одновременно поддерживать соединение с сервером. По умолчанию задается значение порядка 150. Учитывая трафик, связанный с обращением к вашему серверу, вы можете увеличить или уменьшить его. Если ваш Web-узел пользуется большой популярностью у клиентов и вы задали неоправданно большое значение MaxClients, это может привести к снижению производительности Apache. Если же значение этой директивы слишком мало, то некоторые клиенты не смогут установить соединение с сервером. Большие значения MaxClients приведут к тому, что при увеличении трафика требования к объему памяти и области подкачки возрастут.

Число соединений, заданное с помощью директивы MaxClients, не то же самое, что число Web-броузеров, поддерживаемых Apache. Каждый Web-броузер может устанавливать несколько соединений с сервером, и все они учитываются при сравнении с MaxClients.

- Listen. По умолчанию сервер Apache принимает обращения через все активные интерфейсы, используя порт с номером 80. Данная директива позволяет изменить номер порта или ограничить число интерфейсов, через которые можно обратиться к серверу. Например, выражение `Listen 192.168.34.98:8080` сообщает Apache, что обращения клиентов должны приниматься только через интерфейс,

с которым связан адрес 192.168.34.98 с использованием порта 8080. Выражение Listen 8000 означает, что взаимодействие с клиентами может осуществляться через все интерфейсы посредством порта с номером 8000.

- **BindAddress**. Если компьютер, на котором выполняется сервер Apache, содержит несколько сетевых интерфейсов, то, используя данную директиву, вы можете связать Apache лишь с одним из интерфейсов. Например, если в конфигурационном файле задано выражение BindAddress 192 .168 . 34 . 98, сервер будет использовать лишь интерфейс 192.168.34.98. При установке Apache в конфигурационный файл включается выражение BindAddress *, посредством которого Apache связывается со всеми интерфейсами.

СОВЕТ ✔ Если вы хотите, чтобы сервер принимал обращения только с локального компьютера, вам надо задать опцию BindAddress 127.0.0.1. При этом взаимодействие с другими компьютерами поддерживаться не будет. Для обращения к локальному серверу можно использовать URL http://127.0.0.1 или http://localhost.

- Port. Данная директива указывает Apache, какой порт должен использоваться для взаимодействия с клиентами. По умолчанию принимается номер порта 80.

- **ServerAdmin**. С помощью данной директивы вы можете указать свой почтовый адрес. По умолчанию в конфигурационном файле задается адрес webmaster. Создав соответствующий псевдоним в конфигурационном файле сервера SMTP, вы перенаправите письма, приходящие от пользователей на свой адрес. В обычных условиях адрес, указанный в качестве значения данной директивы, не предоставляется пользователям, но он возвращается в составе некоторых сообщений об ошибке.

- ServerName. Если значение данной директивы отличается от имени вашего компьютера, вы можете устранить это несоответствие, указав правильное значение.

- **DefaultType**. Если Apache не может определить MIME-тип данных ни на основании расширения файла, ни с помощью "магической" последовательности, он возвращает MIME-тип, указанный в качестве значения данной директивы. Обычно это text/plain, но при необходимости вы можете задать другое значение. Изменять DefaultType имеет смысл в том случае, если на Web-узле находится много файлов, содержащих данные определенного типа, и есть опасность, что MIME-тип некоторых файлов не будет распознан.

- **HostnameLookups**. Данная директива может принимать значение On или Off. Если задано значение On, Apache будет преобразовывать адреса клиентов, обращающихся к серверу, в доменные имена и записывать их в файл протокола. Это упрощает анализ информации, содержащейся в файле. Однако преобразование адреса занимает дополнительное время и сетевые ресурсы, поэтому системные администраторы часто отказываются от такой возможности.

- LogLevel. Сервер Apache записывает информацию о своих действиях в файл протокола. Объем этой информации вы можете указывать, задавая значение debug, info, notice, warn, error, crit, alert или emerg директивы LogLevel.

(Здесь значения директивы перечислены в порядке убывания объема данных, записываемых в файл протокола.) По умолчанию используется значение warn.

- `CustomLog`. Для данной директивы задаются два значения: имя файла протокола и формат информации, записываемой в этот файл. В данном случае речь идет о файле протокола, в который помещаются сведения о клиентах, обращающихся к серверу за получением Web-страниц. Формат может быть задан с помощью ключевых слов common, `agent`, `referer` и combined. Для обеспечения большей степени гибкости в конфигурационном файле `httpd.conf` предусмотрены средства, позволяющие администратору определить собственный формат записи данных. Чтобы создать несколько файлов протоколов, надо включить в конфигурационный файл несколько директив CustomLog.

Помимо опций общего назначения, описанных выше, в файле `httpd.conf` содержатся также дополнительные опции. Многие из них не будут рассматриваться в данной книге. Если вам потребуется более подробная информация о настройке сервера, обратитесь к документации по Apache или к книгам, посвященным данному продукту.

Описание каталогов

В состав URL входит от двух до четырех компонентов.

- Протокол. Первый компонент URL (например,`http://` или`ftp://`) определяет протокол, используемый для взаимодействия. В данной главе в основном обсуждаются серверы, поддерживающие протокол HTTP (в этом случае URL начинается с символов `http://`). Для обращения к защищенным узлам используются URL, начинающиеся с `https://`.

- Имя узла. Имя узла, входящее в состав URL, представляет собой доменное имя компьютера, на котором выполняется Web-сервер. Например, в URL `http://www.threeroomco.com/thepage/index.html` именем узла является www.`threeroomco.com`. (Одному компьютеру может соответствовать несколько доменных имен. Такая ситуация возникает в том случае, если в конфигурационном файле сервера DNS для этого компьютера задано несколько записей A или `CNAME`. (Настройка сервера DNS описывались в главе 18.)

- Имя файла. В большинстве случаев HTTP-запрос предполагает передачу файла. В составе URL за именем узла следует имя файла (с указанием имени каталога). Например, в URL `http://www.threeroomco.com/thepage/index.html` ссылкой на файл является компонент `thepage/index.html`. Несмотря на то что имя файла отделяется от имени узла косой чертой, этот символ не является обозначением корневого каталога системы Linux. Путь к файлу йачинается от *корневого каталога документов,* определенного для Web-узла. Если имя файла в составе URL не указано, сервер возвращает клиенту Web-страницу по умолчанию, заданную с помощью директивы `DirectoryIndex`.

- Дополнительная информация. Некоторые URL содержат дополнительную информацию. Например, позиции в составе Web-документа может быть присвоено имя. Это имя указывается в URL после имени файла и отделяется от него символом #.

URL, в начале которого указан протокол FTP, может содержать пользовательское имя и пароль.

В конфигурационном файле Apache содержится несколько опций, которые позволяют указывать каталоги для хранения файлов, предназначенных для обработки Web-сервером. Если вы некорректно зададите значения этих опций, некоторые из Web-страниц станут не доступны. Директивы, описывающие каталоги, перечислены ниже.

- ServerRoot. С помощью этой директивы задается корень поддерева файловой системы, используемого для хранения двоичных файлов Apache. В большинстве случаев при инсталляции сервера устанавливается значение `"/usr"` этой опции. Изменять его не следует.

- DocumentRoot. В каталоге, указанном с помощью этой директивы, хранятся файлы, содержащие статические Web-страницы. По умолчанию для данной опции задается `"/home/httpd/html"` или другое подобное значение. (В файле `httpd.conf` имя каталога обычно помещается в кавычки.)

ВНИМАНИЕ Значение директивы DocumentRoot не следует завершать косой чертой. Несмотря на то что в системе Linux такая ссылка на каталог является корректной, для Apache она приведет к возникновению ошибки.

- `UserDir`. Если первый из каталогов, предшествующих имени файла в составе URL, начинается с символа , Apache интерпретирует его имя как имя пользователя и старается найти файл в рабочем каталоге соответствующего пользователя. Директива UserDir указывает имя подкаталога, в котором следует искать файл. Предположим, что для данной директивы задано значение `public_html` и удаленный пользователь ввел в поле адреса броузера URL `http://www.threeroomco.compilation/~abrown/photos.html`. Тогда Apache попытается вернуть пользователю файл `photos.html`, расположенный в подкаталоге `public_html` рабочего каталога пользователя abrown. Если задано значение `disabled` данной директивы, обращение к файлам, находящимся в рабочих каталогах пользователей, запрещено. Если вы хотите запретить доступ лишь к части пользовательских каталогов, вам надо после ключевого слова `disabled` указать имена пользователей, рабочие каталоги которых закрыты для обращения. Данная директива часто помещается в состав директивы `<IfModule>`, которая проверяет, загружен ли модуль Apache, предназначенный для поддержки пользовательских каталогов. (Модули Apache будут рассматриваться в следующем разделе.)

- `DirectoryIndex`. Некоторые URL не содержат имя файла; в них указано лишь имя каталога (в некоторых случаях оно завершается косой чертой). Когда сервер Apache получает подобный URL, он сначала старается найти *файл индекса*, имя которого задается с помощью директивы Directorylndex. В большинстве случаев по умолчанию принимается имя `index.html`, установленное в качестве значения данной опции при инсталляции сервера. При необходимости вы можете задать другое имя файла. Если пользователь введет URL `http://www.threeroomco.com/public/`, Apache вернет файл index.html, находящийся в подкаталоге `public` каталога, указанного с помощью директивы DocumentRoot. Если вы укажете несколько файлов индекса, Apache станет поочередно искать все файлы.

Во многих дистрибутивных пакетах при установке Apache задаются каталоги, которые вполне можно использовать в процессе работы сервера. Вам надо лишь просмотреть конфигурационный файл, выяснить имена этих каталогов и поместить в них файлы, которые Web-сервер должен предоставлять пользователям. Если вы предпочитаете размещать свои файлы в других каталогах, вам надо внести соответствующие изменения в состав конфигурационного файла. Возможно, вам потребуется изменить файл индекса. Необходимость в этом возникает в основном тогда, когда вы устанавливаете Apache взамен другого сервера, в котором использовалось другое имя файла индекса.

Загрузка модулей Apache

Одно из преимуществ Apache состоит в том, что этот Web-сервер является расширяемым. Программист может написать новый *модуль,* реализующий дополнительные возможности, при этом исходный код Apache остается неизменным. Более того, для использования нового модуля не нужно даже перекомпилировать сервер. Посредством модулей реализуются управление доступом, разбор дополнительной информации, передаваемой клиентами, и многие другие функции. Основная часть стандартных функций Apache также реализована в виде модулей.

Просмотрев содержимое конфигурационного файла `httpd.conf`, вы найдете в нем ссылки на модули, формируемые посредством директивы `LoadModule`. Пример подобной ссылки приведен ниже.

```
LoadModule mime_module        lib/apache/mod_mime.so
```

В качестве значения данной директивы задается внутреннее имя модуля (в данном примере `mime_module`) и имя файла, в котором содержится сам модуль (`lib/apache/mod_mime.so`). В данном случае имя файла указывается относительно каталога, заданного посредством директивы `ServerRoot`, но при желании вы можете указать полный путь.

Модули, которые используются часто, можно непосредственно встраивать в двоичные файлы Apache. Чтобы определить, какие модули уже содержатся в исполняемых файлах, надо задать команду `httpd -l` (или `apache -l`). В некоторых случаях модули, встроенные в состав Apache или загруженные посредством LoadModule, необходимо активизировать, включив для этого в конфигурационный файл директиву AddModule.

```
AddModule mod_mime.c
```

В качестве значения директивы AddModule задается имя файла с исходным кодом модуля. Для важных модулей в конфигурационном файле Apache содержится как директива LoadModule, так и директива AddModule.

Как правило, администраторам не приходится включать новые модули; стандартная конфигурация Apache позволяет решать большинство задач, связанных с организацией функционирования Web-узла. Более того, чтобы уменьшить риск незаконного проникновения в систему, иногда приходится исключать некоторые модули. Удаляя модули, следует соблюдать осторожность, так как, не зная структуры Apache, нельзя заранее сказать, как отсутствие некоторых из них повлияет на работоспособность сервера.

Если Apache не может выполнить необходимые вам действия, следует прочитать описания модулей и решить, какой из них пригоден для решения этой задачи. Дополнительную информацию о доступных модулях можно получить на Web-узле Apache Module Register по адресу `http://modules.apache.org`. Выполнив поиск по ключевым

словам, вы получите информацию о модулях, созданных сторонними организациями, и адреса Web-узлов этих организаций.

Настройка kHTTPd

В системах, подобных UNIX, и, в частности, в Linux, можно выделить два типа процессов: *процессы ядра* (kernel space processes) и *пользовательские процессы* (user space processes). Процесс ядра запускается очень быстро, а для запуска пользовательского процесса требуется относительно много времени, кроме того, пользовательский процесс часто должен осуществлять обмен важными данными с ядром. На практике такая особенность пользовательских процессов приводит к возникновению проблем, так как основная обработка информации осуществляется в пространстве пользовательского процесса. Задержка, связанная с запуском процесса, оправдывается повышением уровня безопасности и стабильности. Процессы ядра пользуются привилегиями при взаимодействии с аппаратными средствами, файловой системой и другими ресурсами, поэтому ошибка в программе или несанкционированное вмешательство извне могут привести к разрушению системы.

Пытаясь найти способы увеличения производительности, специалисты заметили, что, несмотря на то, что Web-сервер представляет собой пользовательский процесс, большая часть его функций выполняется процессами ядра, в результате работа сервера в основном представляет собой последовательность обращений к ядру. На рис. 20.1 условно показано взаимодействие Web-сервера (в качестве примера которого выбран сервер Apache) с ядром. Реально обмен с ядром происходит гораздо сложнее, чем показано на рисунке, например, для чтения файла и передачи данных по сети необходимо выполнить целый ряд операций. При этом расходуется время процессора, память и другие ресурсы.

Для того чтобы оптимизировать обслуживание HTTP-запросов, были созданы простые Web-серверы, выполняющиеся как процессы ядра. В результате исчезла необходимость постоянного взаимодействия ядра и пользовательского процесса, и скорость обработки запросов клиента существенно увеличилась. Начиная с версии 2.4.x в состав ядра входят компоненты, реализующие Web-сервер kHTTPd. Подробная информация о таких компонентах находится по адресу `http://www.fenrus.demon.nl`. Настройка сервера, выполняющегося в виде пользовательского процесса, осуществляется путем записи данных в конфигурационные файлы, находящиеся в каталоге `/proc/sys/net/khttpd`.

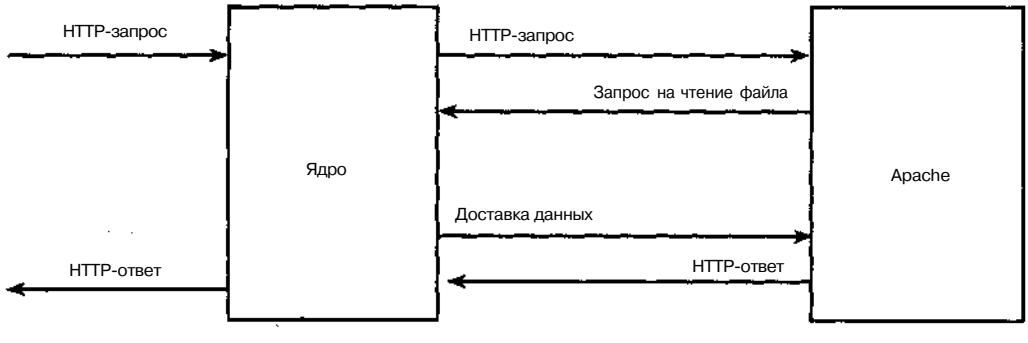


Рис. 20.1. Web-сервер, выполняющийся как пользовательский процесс, интенсивно взаимодействует с ядром

Для того чтобы обеспечить работу такого сервера, необходимо предпринять следующие действия.

1. Включите поддержку kHTTPd при конфигурации ядра Linux. Для этого используется опция Kernel HTTPd Acceleration, находящаяся в меню Networking Options. Вы можете сформировать требуемый компонент в виде модуля или непосредственно включить его в состав ядра.

2. Измените конфигурацию Apache так, чтобы этот сервер использовал для приема обращений клиентов порт 8080 или любой другой, отличный от порта 80.

3. Перезагрузите систему или загрузите модуль ядра kHTTPd. В зависимости от конфигурации он либо загрузится автоматически, либо вам придется использовать команду `insmod khttpd`.

4. Укажите серверу kHTTPd на то, что он должен принимать запросы клиентов через порт 80. Для этого надо выполнить команду `echo 80 > /proc/sys/net/khttpd/serverport`.

5. Введите команду `echo 8080 > /proc/sys/net/clientport`. В результате ее выполнения kHTTPd будет передавать запросы, которые не может обработать самостоятельно, серверу Apache, используя порт 8080. (Если на шаге 2 вы указали порт, отличный от 8080, то должны задать тот же порт в составе данной команды.)

6. Сообщите kHTTPd, в каком каталоге следует искать незакодированные статические файлы. Для этого выполните команду `echo /home/httpd/html > /proc/sys/net/khttpd/documentroot`. Вместо каталога /home/httpd/html вы можете указать другой каталог, следите лишь за тем, чтобы он совпадал с каталогом, который был задан в файле `httpd.conf` в качестве значения директивы `DocumentRoot`.

7. Если на вашем Web-узле содержатся PHP3 или защищенные HTML-документы, повторите предыдущее действие, но поместите имя каталога в файл `/proc/sys/net/khttpd/dynamic`.

8. Введите команду `echo 1 > /proc/sys/net/khttpd/start`, в результате которой сервер kHTTPd начнет работу. Указанный здесь файл является своеобразным аналогом сценария запуска SysV.

При желании вы можете создать сценарий SysV или локальный сценарий, который автоматизировал бы выполнение этапов 4-8 описанной выше процедуры. Независимо от того, будет ли сервер запущен вручную или с помощью сценария, он будет поддерживать простые запросы, предполагающие передачу клиентам статических файлов, находящихся в указанном каталоге. Запрос который не может быть обработан средствами kHTTPd (например, запрос, предполагающий запуск CGI-сценария), будет передан Web-серверу, выполняющемуся как пользовательский процесс. При этом будет использован номер порта, указанный на этапах 2 и 5. Такая передача запроса связана с большими накладными расходами, поэтому если Web-узел предполагает в основном выполнение CGI-сценариев, использовать для его поддержки сервер kHTTPd нецелесообразно. Более того, применять kHTTPd имеет смысл только в том случае, если сервер Apache не справляется с нагрузкой.

Если же интенсивность обращений к серверу не велика, предпочтительнее использовать один из серверов, выполняющихся как пользовательский процесс. Следует также помнить, что kHTTPd официально считается экспериментальным продуктом, поэтому он не может обеспечить такой надежности, как Apache или другие подобные серверы. Поскольку kHTTPd выполняется как процесс ядра, ошибка в программе может нанести системе гораздо больше вреда, чем ошибка в сервере, выполняющемся как пользовательский процесс. Таким образом, если у вас нет веских оснований применять kHTTPd, лучше использовать вместо него хорошо отлаженный и зарекомендовавший себя в работе сервер Apache.

Сервер kHTTPd — не единственный продукт подобного типа, реализованный как процесс ядра. В системе Red Hat применяется сервер TUX, кроме того, в настоящее время ведется работа над созданием других серверов, предназначенных для выполнения в виде процессов ядра Linux.

Поддержка форм и сценариев

Несмотря на то что статические данные часто используются на Web-узлах, типы информации, с которыми может работать Web-сервер, не исчерпываются ими. Многие Web-серверы динамически генерируют Web-страницы. Например, поисковые серверы позволяют пользователю ввести данные в поле редактирования и передать их после щелчка на кнопке, а затем формируют Web-страницу, содержащую результаты поиска. Если вы хотите создать Web-узел, выполняющий подобные действия, вам надо уметь сконфигурировать его соответствующим образом. В этом разделе будут кратко рассмотрены принципы динамического создания Web-страниц и опции Apache, используемые для активизации соответствующих средств сервера. Чтобы получить более подробную информацию, обратитесь к документам, в которых описаны эти вопросы.

Статические данные, формы и CGI-сценарии

В предыдущем разделе речь шла в основном о статических данных. Этот термин применяется для обозначения информации, при отображении которой не предполагается взаимодействие с пользователем. Ниже приведены примеры статических данных.

- HTML-файлы. В настоящее время основная часть данных в Internet представлена в формате HTML (Hypertext Markup Language — язык разметки гипертекста). HTML-файлы имеют расширение .htm либо .html и содержат текстовую информацию с элементами разметки. Элементы разметки выполняют форматирование текста. Например, элемент **<P>** помечает начало абзаца, а </P> — конец абзаца. В языке HTML также предусмотрена возможность связывания Web-страниц с другими документами, расположенными в Internet (в частности, в Web). Такое связывание осуществляется посредством гипертекстовых ссылок. После щелчка на гипертекстовой ссылке ресурс, на который она указывает, автоматически воспроизводится либо сохраняется на диске. Конкретные действия по обработке ресурса зависят от настройки Web-броузера.

- Текстовые файлы. Файлы такого типа чаще всего имеют расширение .txt. Текстовые файлы, которые Web-серверы предоставляют пользователям, отображаются

броузерами, но элементы форматирования и гипертекстовые ссылки в них отсутствуют.

- **Графические файлы.** Почти все HTML-документы содержат ссылки на графические файлы, представленные в различных форматах. Эти файлы также являются статическими. Некоторые файлы содержат анимационные данные, но несмотря на это, они все же считаются статическими файлами. Термин статический относится к содержимому файла, а не к способу его отображения.

- **Документы в различных форматах.** Иногда на Web-страницах содержатся ссылки на файлы PDF, Microsoft Word, архивы . zip и . tar, а также данные, представленные в других форматах. Некоторые броузеры передают эти файлы для обработки соответствующим приложениям, другие сохраняют их на диске.

Статические файлы содержатся в каталогах, заданных посредством директив DocumentRoot и UserRoot в подкаталогах этих каталогов. Взаимодействие клиента с сервером, предполагающее передачу статических файлов, осуществляется следующим образом: клиент передает серверу запрос, сервер находит этот файл на диске и передает клиенту. Если не принимать во внимание тот факт, что сам запрос содержит данные, можно сказать, что в данном случае информация передается в одном направлении: от сервера клиенту.

При обработке динамических данных информация передается как от сервера клиенту, так и от клиента серверу. Работая в Internet, вы наверняка встречались с динамическими данными. Соответствующие примеры приведены ниже.

- **Поисковые серверы.** Если вы укажете в поле адреса броузера URL поискового сервера, этот сервер предоставит Web-страницу, содержащую форму ввода. Форма позволяет вводить данные (в случае поискового сервера — ключевые слова). После щелчка на кнопке Search (или при активизации другого интерактивного элемента, запускающего процедуру поиска) введенные вами ключевые слова передаются Web-серверу, который осуществляет поиск и создает Web-страницу для представления результатов.

- **Internet-магазин.** При посещении узла электронной коммерции, или Internet-магазина, Web-сервер предоставляет вам возможность выбрать в интерактивном режиме товар и поместить его в "корзинку" покупателя. В процессе обмена данными между Web-сервером и Web-броузером сервер предоставляет броузеру информацию о товарах, а броузер передает серверу ваш адрес, номер платежной карточки и другие необходимые данные, а также подтверждает факт покупки. Особенности взаимодействия зависят от реализации конкретного Web-узла, но в любом случае Web-сервер динамически формирует Web-страницу, а Web-броузер передает серверу информацию, введенную пользователем.

- **Web-узлы, настраиваемые с учетом интересов пользователей.** Некоторые Web-узлы предоставляют пользователям специальные средства регистрации и передают данные, специально сформированные для этого пользователя. Например, обратившись на узел Slashdot (`http://slashdot.org`), вы можете зарегистрироваться и указать при регистрации тип и объем интересующих вас данных. При работе с подобными Web-узлами на стороне клиента создается запись cookie, которая иденти-

фицирует клиент при последующих обращениях. (Записи cookie часто создаются и при взаимодействии с серверами электронной коммерции.)

Приведенные выше примеры представляют лишь частные случаи применения динамических Web-узлов. Возможности подобных узлов ограничены лишь воображением разработчиков и их готовностью реализовать свои планы. С точки зрения Web-сервера основное различие между динамическими и статическими Web-узлами состоит в том, что на динамическом узле HTML-документ (или документ в другом формате) создается в процессе работы сервера на основании данных, полученных от клиента. Для реализации динамических Web-узлов используются следующие средства.

- **Web-формы.** Web-форма— это Web-страница, предоставляющая пользователю поля редактирования, списки, кнопки и другие интерактивные элементы, позволяющие вводить данные. Так, например, Web-страница, формируемая поисковым сервером, обычно содержит поле редактирования для ввода ключевых слов и кнопку для запуска процедуры поиска. Серверы, поддерживающие узлы электронной коммерции, помимо кнопок и полей редактирования, часто включают на генерируемые ими Web-страницы списки, предназначенные для указания страны или штата. Web-формы создаются посредством HTML-кода, который может содержаться в статическом файле либо генерироваться динамически.

- **CGI-сценарии.** CGI (Common Gateway Interface — интерфейс общего шлюза) определяет порядок взаимодействия программ, осуществляющих динамическую генерацию HTML-документов, с Web-сервером. CGI-сценарии могут быть написаны практически на любом языке. Для их создания используются не только компилируемые, но и интерпретируемые языки, например Perl. Web-сервер запускает CGI-сценарий на выполнение в том случае, если это предусмотрено в URL. В процессе выполнения сценарий получает от Web-сервера данные, введенные пользователем, при необходимости вызывает другие программы и генерирует Web-страницу, передаваемую в ответ на запрос клиента.

- SSI (Server Side Includes — включаемые средства на стороне сервера) также предназначены для динамической генерации содержимого документа, но, в отличие от CGI-сценариев, которые формируют всю Web-страницу, SSI лишь изменяют шаблоны. SSI не обеспечивают такой гибкости, как CGI, но их удобно использовать для внесения небольших изменений в состав статических Web-страниц, например, для включения информации о текущей дате.

Существуют также другие средства динамической генерации содержимого Web-страниц. Например, в настоящее время в распоряжение разработчика предоставляются многочисленные инструменты, которые можно успешно использовать вместо CGI, но CGI-сценарии до сих пор остаются самым популярным средством решения подобных задач. Заметьте, что Web-страницы, генерируемые CGI-сценариями, могут содержать формы ввода. Эти два инструмента не исключают друг друга. Напротив, данные, обрабатываемые CGI-сценариями, чаще всего вводятся посредством форм.

Поддержка CGI-сценариев

Если вы собираетесь использовать CGI-сценарии, то должны сообщить серверу Apache о своем намерении. При получении URL, содержащего имя сценария, сервер должен

запустить этот сценарий, а также организовать обработку данных, переданных клиентом, формирование Web-страницы и передачу ее броузеру. При использовании CGI-сценария Apache выполняет роль посредника между клиентом и сценарием на стороне сервера. Настроить сервер для выполнения подобных функций не сложно. Вы должны лишь разрешить поддержку CGI-сценариев и сообщить Apache типы запросов, при получении которых следует запускать сценарии.

Для обеспечения работы с CGI необходимо загрузить соответствующий модуль Apache.

```
LoadModule cgi_module          lib/apache/mod_cgi.so
```

Если компоненты, предназначенные для поддержки CGI-сценариев, включены в состав двоичных файлов Apache, вам надо активизировать их посредством директивы AddModule. (В некоторых случаях активизировать надо и компоненты, реализованные в виде модулей.)

```
AddModule mod_cgi.c
```

В результате сервер Apache получает возможность запускать CGI-сценарии и взаимодействовать с ними. Вам осталось лишь разрешить поддержку CGI для конкретных файлов и каталогов. Сделать это можно несколькими способами.

- ScriptAlias. Данная директива решает две задачи. Во-первых, она сообщает серверу Apache о том, что файлы, содержащиеся в указанном каталоге, должны интерпретироваться как CGI-сценарии. Во-вторых, посредством этой директивы задается соответствие между каталогом, расположенным на диске, и каталогом, который указывается в URL. Например, выражение `ScriptAlias /scripts/ "/home/httpd/cgi-bin/"` отображает физический каталог `/home/httpd/cgi-bin/` в каталог `/scripts` в составе URL. В результате, если пользователь укажет URL `http://www.threeroomco.com/scripts/test.pl`, сервер запустит на выполнение сценарий `test.pl`, содержащийся в каталоге `/home/httpd/cgi-bin/`. Часто при инсталляции Apache опции LoadModule и AddModule по умолчанию включаются в конфигурационный файл; вероятнее всего, вы встретите их, просматривая содержимое файла `httpd.conf`. Для работы с CGI-сценариями часто бывает нужен модуль `mod_alias`. Соответствующая директива обычно по умолчанию включается в состав конфигурационного файла. При возникновении проблем, проверьте, загружен ли данный модуль.

- Options +ExecCGI. Разрешить выполнение CGI-сценариев можно, указав значение +ExecCGI директивы Options. Данная опция не должна указываться для всей системы, ее имеет смысл применять только к отдельным каталогам (т. е. она должна присутствовать только в составе директивы <Directory>).

- .htaccess. Контролировать доступ к отдельному каталогу можно, размещая в нем файл `.htaccess`. Если в файле `.htaccess` содержится запись Options +ExecCGI, Apache будет запускать CGI-сценарии, находящиеся в этом каталоге. Чтобы это произошло, в файле `httpd.conf` должна находиться запись AllowOverride Options; эта запись должна воздействовать как минимум на каталог, содержащий файл .htaccess.

ВНИМАНИЕ Наличие записей Options +ExecCGI и **AllowOverride Options** представляет угрозу для системы. При неправильном использовании этих средств пользователи получают возможность создавать сценарии, предоставляющие полный доступ к системе. По этой причине в большинстве дистрибутивных пакетов использование файла `.htaccess` запрещено.

Часто при настройке Apache в конфигурационный файл включается директива ScriptAlias, отображающая каталог /home/httpd/cgi-bin файловой системы в каталог /cgi-bin в составе URL. Такая настройка удобна для администратора. Чтобы установить CGI-сценарий и сделать его доступным для пользователя, достаточно разместить соответствующий файл в каталоге /home/httpd/cgi-bin. При этом необходимо обратить внимание на права доступа к файлу. Поскольку сценарий предназначен для выполнения, для файла, содержащего код этого сценария, должен быть установлен соответствующий признак. Если вы написали сценарий самостоятельно или скопировали его с Web- или FTP-узла, то после размещения его в каталоге /home/httpd/cgi-bin надо выполнить команду **chmod** a+x *имя_сценария*.

Создание CGI-сценариев

Подобно другим сценариям, CGI-сценарии представляют собой программный код, предназначенный для выполнения. Данная глава не является руководством по написанию CGI-сценариев; в этом разделе приведены лишь некоторые общие рекомендации по работе с ними. Если вам потребуется дополнительная информация о создании CGI-сценариев, обратитесь к Web-странице по адресу `http://httpd.apache.org/docs/howto/cgi.html` либо к одной из книг, посвященных этой теме.

CGI-сценарии принимают входные данные через стандартный ввод и выводят сгенерированную Web-страницу через стандартный вывод. Вывод текста, который должен быть передан клиенту, ничем не отличается от обычного вывода на консоль. Необходимо лишь помнить, что клиент просматривает информацию посредством Web-броузера, поэтому ваш CGI-сценарий должен генерировать информацию в формате HTML либо в другом формате, поддерживаемом Web-клиентом. (Например, вы можете сформировать ответ в виде графического файла.)

Помимо HTML-кода, CGI-сценарий должен создать поле заголовка Content-Type и в качестве его значения указать MIME-тип данных, передаваемых клиенту. Это поле имеет следующий вид:

```
Content-type: text/html\r\n\r\n
```

В данном примере указан MIME-тип text/html, означающий, что в ответ на запрос клиента CGI-сценарий сгенерировал HTML-документ. Символы \r\n\r\n соответствуют двум переводам строки, в результате поле заголовка будет отделено от остальных данных пустой строкой. Код сценария зависит от используемого вами языка программирования. Простейший пример CGI-сценария, написанного на языке Perl, приведен в листинге 20.1. Как видно из листинга, в процессе выполнения сценарий выводит строку текста. Записав файл с этим кодом в каталог, предназначенный для размещения CGI-сценариев, установите права доступа. Если после этого вы зададите URL сценария в поле адреса броузера, то увидите строку "Hello, Web".

Листинг 20.1. Простой CGI-сценарий, написанный на языке Perl

```
#!/usr/bin/perl
print "Content-type: text/html\r\n\r\n";
print "Hello, Web";
```

С обработкой входных данных дело обстоит несколько сложнее. Ваш сценарий получит данные только в том случае, если они были введены пользователем посредством интерактивных элементов, содержащихся в форме. Данные поступают на вход сценария в виде набора пар имя-значение. Имя отделяется от значения символом =, а пары имя-значение разделяются символами &. Пример строки параметров, передаваемой CGI-сценарию, приведен ниже.

```
city=Oberlin&state=OH&zip=44074
```

Перед тем как использовать полученные данные, надо произвести разбор строки параметров. В языке Perl предусмотрены мощные средства работы со строками. Этот факт стал одной из причин популярности Perl среди разработчиков CGI-сценариев.

Повышение уровня защиты при использовании CGI-сценариев

Если на Web-узле присутствуют CGI-сценарии, любой пользователь, работающий с Web-броузером, имеет возможность запустить на стороне сервера программу. Это может стать источником проблем, связанных с безопасностью системы. Определенную опасность для системы представляет любой сервер, но при использовании на Web-узле CGI-сценариев шансы злоумышленников на успех существенно возрастают. Ни об одном достаточно сложном CGI-сценарии нельзя с уверенностью сказать, что он безупречен с точки зрения защиты. Разработчики серверов прилагают большие усилия для того, чтобы устранить возможность проникновения с его помощью в систему, но несмотря на это, время от времени в серверах обнаруживаются ошибки. В отличие от серверов, CGI-сценарии в основном создаются системными администраторами, которые часто не имеют большого опыта программирования. В результате сценарии получаются уязвимыми для атак извне.

Существуют способы, позволяющие уменьшить риск, связанный с использованием CGI-сценариев. Перед установкой сценариев необходимо еще раз поверить значения директив User и Group в файле `httpd.conf`. CGI-сценарии выполняются с полномочиями пользователя, указанного посредством этих директив, поэтому, используя учетную запись, предусматривающую минимальные права, вы ограничите возможности злоумышленника, если тому удастся получить контроль над CGI-сценарием. Идеальный вариант — создать учетную запись и группу, специально предназначенные для обеспечения работы сервера Apache; регистрация в системе с помощью этой учетной записи должна быть запрещена. Однако подобная мера не гарантирует безопасность системы. Недостатки в сценарии могут стать базой, используя которую взломщик продолжит действия по проникновению в систему.

Чтобы уменьшить опасность для системы, можно использовать готовые сценарии, поставляемые в составе библиотек. Такой подход, с одной стороны, упростит процедуру создания Web-узла, а с другой стороны, позволит избежать грубых ошибок в сценарии.

Библиотеки сценариев размещены на различных Web-узлах, например, вы можете обратиться по адресу `http://www.cpan.org`.

Чтобы злоумышленник, получивший контроль над Web-сервером, не смог нанести существенный вред компьютерам вашей сети, надо принять дополнительные меры. Например, желательно отключить ненужные серверы и ограничить доступ с компьютера, на котором выполняется Web-сервер, к другим компьютерам сети. Действия, направленные на повышение уровня защиты системы, рассматриваются в части IV.

Поддержка защищенных Web-узлов

При использовании сценариев часто осуществляется шифрование передаваемых данных. Действия по кодированию и декодированию информации при обмене между Web-сервером и Web-броузером определяется протоколом SSL (Secure Sockets Layer — уровень защищенного гнезда). Протокол SSL часто используется на узлах электронной коммерции для защиты важных данных. Для поддержки SSL-кодирования при работе Apache требуется дополнительное программное обеспечение, например, `mod_ssl` (`http://www.modssl.org`) или программы, разработанные в рамках проекта Apache-SSL (`http://www.apache-ssl.org`). Для поддержки SSL можно также использовать продукты, распространяемые на коммерческой основе.

Задачи, решаемые с помощью SSL

SSL — это технология кодирования, подобная той, которая используется при обеспечении работы протокола удаленной регистрации SSH. (Строго говоря, эти протоколы применяют одни и те же средства шифрования, так как работа популярного пакета OpenSSH основана на использовании пакета OpenSSL, который также применяется некоторыми реализациями Apache, поддерживающими SSL.) SSL позволяет решить следующие две проблемы, возникающие при обмене между Web-клиентом и Web-сервером.

- Шифрование. SSL позволяет обеим взаимодействующим сторонам выполнять шифрование данных, обеспечивая тем самым их сохранность. Это необходимо в тех случаях, когда Web-клиент должен обмениваться с Web-сервером важной информацией, например передавать номера платежных карточек и банковских счетов. Для кодирования применяется технология открытого ключа, согласно которой каждая из взаимодействующих сторон использует два ключа. Шифрование осуществляется с помощью *открытого,* или *общего,* ключа, предоставляемого другой взаимодействующей стороной, а для расшифровки применяется собственной *закрытый,* или *личный,* ключ. Таким образом, передавая данные, участник сетевого взаимодействия уверен, что расшифровать их сможет только тот, для кого они предназначены.

- Аутентификация. Даже при использовании шифрования передача важных данных по Internet связана с определенным риском. Может оказаться, что принимающий узел — не тот, за кого он себя выдает. Например, если вы ввели в поле адреса броузера URL `http://www.abigretailer.com`, можете ли вы быть уверены, что ваш запрос попадет на тот узел, на который вы его отправляете? Не исключено, что злоумышленнику удалось изменить настройку сервера DNS или маршрутизатора и перенаправить запрос на свой компьютер. SSL предоставляет средства аутентификации участников взаимодействия. Идентификация осуществляется посредством

сертификатов, предоставляемых организацией, специализирующейся на этом. Сертификат, полученный от сертифицирующей организации (CA — certificate authority), представляет собой цифровой код, используемый для создания общего ключа. Если один участник взаимодействия передал свой сертификат, полученный от CA, то другой участник может быть уверен, что его партнер по обмену данными — именно тот, за кого он себя выдает. (В последнее время стало ясно, что даже сертификаты не позволяют гарантированно идентифицировать участников взаимодействия. Так, в 2001 г. сертификаты Microsoft были по ошибке выданы организациям, не имеющим к корпорации никакого отношения.)

При необходимости вы сами можете выступать в роли сертифицирующей организации, однако ваши сертификаты будут пригодны только для внутреннего использования. Внешние пользователи не будут иметь никакой гарантии того, что сертификат не фальсифицирован. Поэтому, если вы собираетесь организовать узел электронной коммерции, вам необходимо получить сертификат от CA. Список CA можно найти по адресу `http://www.apache-ssl.org/#Digital_Certificates`. Пользователям, обращающимся к Web-узлам посредством броузеров, сертификаты не нужны, так как Web-серверы практически никогда не проверяют идентичность пользователей.

При работе посредством протокола SSL используется порт, отличный от порта 80. По умолчанию для взаимодействия по защищенному протоколу HTTP (HTTPS) применяется порт 443. Чтобы Web-броузер указал в запросе этот порт, URL, введенный пользователем, должен начинаться с символов `https://`. Настраивая Apache для поддержки SSL, вы можете установить один сервер, который будет по-разному реагировать на обращения через порты с номерами 80 и 443, либо использовать два сервера различных типов. Первый подход реализовать проще, но может возникнуть ситуация, при которой целесообразнее использовать два сервера (например, Apache для обработки SSL-запросов и `thttpd` для поддержки обычного HTTP-взаимодействия).

Настройка средств поддержки SSL

Для того чтобы сервер Apache мог поддерживать SSL-соединения, надо сконфигурировать SSL-пакет. В настоящее время в системе Linux чаще всего используются два таких пакета.

- `SSLeay` (`http://www2.psy.uq.edu.au/~ftp/Crypto/ssleay/`)
- `OpenSSL` (`http://www.openssl.org`)

Вскоре после своего появления OpenSSL приобрел статус стандарта в системе Linux. Он содержится в составе многих дистрибутивных пакетов Linux, включая Debian, Mandrake, Red Hat и SuSE. Пакеты SSLeay и OpenSSL выполняют одинаковые функции, но исполняемые файлы носят разные имена (`ssleay` и `openssl`) и для их настройки используются различные конфигурационные файлы.

После инсталляции OpenSSL вам необходимо получить сертификат. Для работы в Internet потребуется сертификат, выданный CA, но для тестирования сервера можно создать сертификат самостоятельно. Некоторые сценарии установки Apache SSL создают сертификат автоматически. Если в процедуре инсталляции не предусмотрено формирование сертификата, вы можете использовать следующую команду;

```
# openssl req $@ -new -x509 -nodes \
 -config /usr/share/doc/apache-ssl/examples/ssleay.cnf \
 -out /etc/apache-ssl/apache.pem \
 -keyout /etc/apache-ssl/apache.pem
```

В данном примере предполагается, что настройка средств поддержки SSL осуществляется посредством конфигурационного файла `/etc/apache-ssl`, а в составе пакета поставляется образец конфигурационного файла `/usr/share/doc/apache-ssl/examples/ssleay.cnf`. При необходимости вы можете изменить имена файлов или каталогов. Обратная косая черта указывает на то, что продолжение команды находится на следующей строке. Если вся команда помещается в одной строке, символ \ можно не использовать.

В процессе выполнения утилита `openssl` запросит дополнительную информацию, например имя компьютера. Эта информация включается в состав сертификата, который содержится в файле `/etc/apache-ssl/apache.pem`.

Впоследствии сгенерированный вами сертификат придется заменить сертификатом, который предоставит вам сертифицирующая организация. Если при использовании сертификата, созданного самостоятельно, пользователь, обратившийся к Web-узлу, увидит предупреждающее сообщение, то при наличии сертификата, выданного CA, такое сообщение не выводится. Предупреждающее сообщение, отображаемое броузером Opera в системе Linux, показано на рис. 20.2. В других броузерах формат сообщения будет отличаться от приведенного на рисунке.

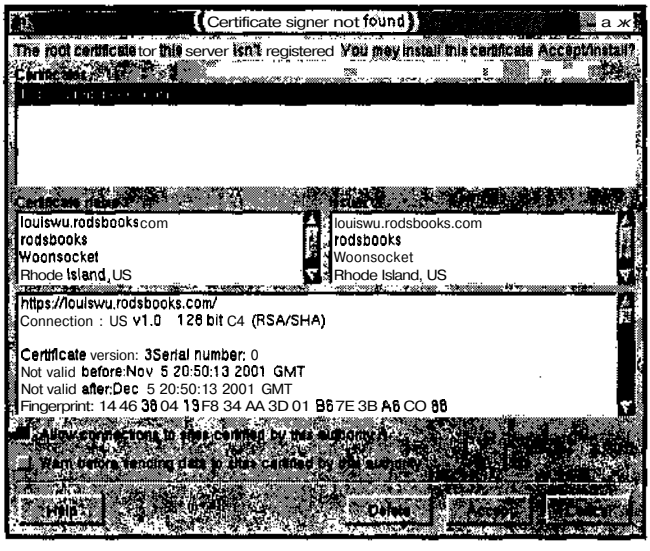


Рис. 20.2. При использовании сертификата, сгенерированного самостоятельно, пользователи, взаимодействующие с узлом, увидят предупреждающее сообщение о том, что сертификат не распознан или срок его действия истек

Установка компонентов Apache, предназначенных для поддержки SSL

Считается, что поддержка SSL в сервере Apache осуществляется за счет дополнительных модулей. На практике для установки SSL-модулей необходимо внести некоторые изменения в структуру сервера и повторно скомпилировать Apache. В некоторых инсталляционных пакетах SSL-модули включены по умолчанию, и код сервера скомпилирован с учетом использования SSL-компонентов. Если вы попытаетесь объединить компоненты обычного сервера Apache и пакета, сформированного для обеспечения поддержки SSL, такой сервер скорее всего работать не будет.

Во многих случаях для управления сервером, созданным с учетом поддержки SSL, используется конфигурационный файл, отличный от файла, применяемого для настройки обычного сервера Apache. Например, в системе Debian сервер Apache, настроенный для поддержки SSL, использует конфигурационный файл **/etc/apache-ssl**, в то время как для стандартной конфигурации Apache в этой системе применяется файл /etc/apache. Конфигурационные файлы для SSL-серверов во многом совпадают с файлами для Apache без поддержки SSL, за исключением некоторых директив, значения которых вам, возможно, придется изменить. Часть этих директив описана ниже.

- **ServerType**. Сервер с поддержкой SSL не может запускаться посредством суперсервера, поэтому для директивы ServerType должно быть установлено значение standalone.
- Использование портов. Для взаимодействия по протоколу SSL используется порт 443. При этом необходимо учитывать, что директива Listen позволяет связать сервер с определенным номером порта.
- Загрузка модулей. В качестве значений директив **LoadModule** и **AddModule** могут быть указаны один или несколько модулей, имеющих отношение к поддержке SSL. Как правило, в конфигурационном файле, сформированном по умолчанию, значения этих директив установлены корректно.
- **SSLRequireSSL**. Включив данную директиву в состав <Directory>, вы запретите доступ к каталогу для клиентов, не поддерживающих SSL. (Значения данной директивы не указываются.) Использование SSLRequireSSL позволяет предотвратить передачу важных данных по незащищенному каналу. Очевидно, что, помимо данной опции, следует применять и другие средства, ограничивающие доступ к каталогу.
- SSLEnable. Директива SSLEnable разрешает использование протокола SSL при обмене данными. Подобно SSLRequireSSL, значения для данной директивы не предусмотрены.
- **SSLCACertificatePath**. Эта директива указывает на каталог, содержащий сертификат. *Например*, в качестве значения SSLCACertificatePath может быть указано /etc/apache-ssl.
- **SSLCertificateFile**. В качестве значения данной директивы указывается файл, содержащий сертификат (например, /etc/apache-SSI/apache **.pem**).

Помимо указанных выше, для управления SSL-взаимодействием могут использоваться и другие директивы. Информацию о них можно получить, просмотрев комментарии в составе конфигурационного файла либо обратившись к документации на сервер Apache или к книгам по данной теме.

Установив конфигурацию сервера, вы можете запускать его для поддержки SSL-взаимодействия. Чтобы обратиться к серверу, надо ввести в поле адреса броузера URL, начинающийся символами `https://`. Если вы самостоятельно сгенерировали сертификат, броузер отобразит предупреждающее сообщение, подобное тому, которое показано на рис. 20.2. Чтобы протестировать создаваемый вами узел, вы можете принять этот сертификат (некоторые броузеры позволяют задать условия дальнейшего использования сертификата).

ВНИМАНИЕ | Работая в Internet, не следует принимать сертификаты, созданные администраторами Web-узлов. Если сервер использует сертификат, выданный сертифицирующей организацией, броузер не отображает предупреждающее сообщение. Если же подобное сообщение появилось на экране, это означает, что при использовании сертификата была допущена ошибка (например, сервер продолжает работать с сертификатом, срок действия которого истек) либо администратор вовсе не обращался к сертифицирующей организации. Появление предупреждающего сообщения также может означать, что злоумышленник пытается представить свой узел как узел официальной организации и собрать с его помощью важные данные.

Организация виртуальных доменов

Ранее в данной главе рассматривалось применение сервера Apache для работы с Web-страницами, принадлежащими одному Web-узлу. Возможно ли разместить на одном компьютере несколько Web-узлов? Положительный ответ на данный вопрос очевиден, так как именно это делают администраторы тех организаций, которые предоставляют в аренду дисковое пространство на своих Web-серверах. Размещение нескольких Web-узлов на одном компьютере обеспечивается посредством механизма *виртуальных доменов,* или *виртуальных узлов.* Конфигурация сервера, предназначенного для поддержки виртуальных доменов, отличается от стандартной конфигурации Apache лишь в деталях.

Использование виртуальных доменов

Наличие виртуальных доменов позволяет Web-серверу по-разному обрабатывать запросы, в зависимости от имен, указанных в них. (Чтобы к Web-серверу можно было обращаться по разным именам, необходимо создать несколько записей в конфигурационном файле DNS-сервера.) Примеры использования виртуальных доменов описаны ниже.

- Если имя сервера изменилось (сервер был перенесен на другой компьютер в составе того же домена), вы можете настроить Apache так, чтобы при обращении по старому имени отображалось сообщение об ошибке и запрос перенаправлялся по новому имени. Со временем, когда большинство пользователей изменят закладки на своих броузерах, от применения виртуальных доменов можно будет отказаться.
- Если две сотрудничающие компании или два отдела одной компании хотят создать свои Web-узлы, они могут разместить их на одном Web-сервере, сконфигурировав

сервер для поддержки виртуальных доменов. В некоторых случаях (особенно если Web-узлы создаются для отделов одной организации) использовать подкаталоги удобнее, чем организовывать обращение к одному и тому же серверу по разным именам, но часто решение о создании виртуальных доменов оправдано.

- Виртуальные узлы могут использовать соседи по студенческому общежитию. Такой подход применим только для тех пользователей, компьютеры которых постоянно соединены с Internet.

- Возможно, вы захотите предоставить свой сервер для размещения Web-узлов других организаций или частных лиц. Очевидно, что заниматься такой деятельностью можно только тогда, когда вы приобретете достаточный опыт поддержки сетей и Web-серверов.

Серверы, поддерживающие виртуальные домены, в основном устанавливаются в больших организациях или на компьютерах провайдеров. В небольших компаниях серверы чаще всего применяются для поддержки одного Web-узла.

Конфигурация виртуальных доменов

Существуют два способа организации работы с виртуальными доменами. Один из них состоит в том, что, в зависимости от *имени*, указанного в запросе, в качестве корневого каталога документов выбираются различные каталоги. Второй способ позволяет устанавливать для каждого виртуального домена разные наборы опций.

Использование VirtualDocumentRoot

`VirtualDocumentRoot` — одна из основных директив, используемых для настройки виртуальных доменов. Эта директива позволяет указать имя каталога, которое будет выполнять роль корневого каталога документов при указании в составе запроса определенного имени. В качестве значения VirtualDocumentRoot указывается имя каталога, которое может содержать различные переменные. (Назначение этих переменных описано в табл. 20.1.)

Рассмотрим в качестве примера следующую запись:

```
VirtualDocumentRoot /home/httpd/%0
```

Таблица 20.1. Переменные, используемые для создания имен каталогов

Переменная	Описание
%%	Символ % в имени каталога
%p	Номер порта, используемый сервером
%N.M	Часть имени, отделенная от других частей точками. N — это число, ссылающееся на компонент имени. 0 означает все доменное *имя*, 1 — первый компонент, 2 — второй компонент и т. д. Значение N также может быть отрицательным: −1 определяет последний компонент имени, −2 — предпоследний компонент и т. д. M принимает такие же значения, как и N, но ссылается не на компонент имени, а на символ в составе компонента. Если вы хотите использовать весь компонент имени, точку и M можно не указывать.

Она сообщает серверу о том, что он должен использовать подкаталог каталога /home/httpd, имя которого соответствует полному имени сервера, указанному в составе запроса. Например, если в запросе задан URL http://www.threeroomco.com/index.html, сервер будет искать файл /home/httpd/www.threeroomco.com/index.html. Такой способ очень удобен, но если вам необходимо поддерживать большое количество Web-узлов, то придется создавать много подкаталогов с достаточно длинными именами (в данном примере все подкаталоги должны присутствовать в каталоге /home/httpd). При необходимости вы можете использовать в качестве имени каталога часть доменного имени. Пример подобного подхода иллюстрирует приведенная ниже запись.

```
VirtualDocumentRoot /home/httpd/%-1/%-2
```

Если в конфигурационном файле содержится такое выражение, то, получив запрос, в котором указан URL http://www.threeroomco.com/index.html, Apache вернет клиенту файл /home/httpd/com/threeroomco/index.html (если он имеется на сервере). Если вы хотите использовать в имени каталога лишь один символ из доменного имени, вам надо включить в состав конфигурационного файла запись наподобие следующей:

```
VirtualDocumentRoot /home/httpd/%-2.1/%0
```

Теперь при получении URL http://www.threeroomco.com/index.html Apache вернет клиенту файл /home/httpd/t/www.threeroomco.com/index.html. Переменная %-2.1 определяет первый (.1) символ в составе имени домена (-2), предшествующего имени домена верхнего уровня.

Независимо от значения директивы VirtualDocumentRoot, вам надо задать значение Off для директивы UseCanonicalName.

```
UseCanonicalName Off
```

Если директива UseCanonicalName будет иметь значение On, устанавливаемое по умолчанию при инсталляции сервера, Apache будет использовать для обработки относительных ссылок доменное имя компьютера, на котором он выполняется. Например, если в документе index.html содержится ссылка на Web-страницу products.html, Apache будет стараться извлечь ее, основываясь на своем каноническом имени. При наличии виртуальных доменов такое поведение недопустимо. Если задать значение Off директивы UseCanonicalName, то для обработки относительных ссылок Apache будет применять имя, соответствующее виртуальному домену.

Использование <VirtualHost>

Альтернативный подход к созданию виртуальных доменов предполагает непосредственное описание каждого из них. Для этого в конфигурационном файле Apache предусмотрены две специальные директивы.

- NameVirtualHost. Данная директива указывается в главном конфигурационном файле Apache и информирует сервер о том, что вы собираетесь использовать виртуальные узлы. В качестве значения этой директивы чаще всего указывается символ *; при этом необходимо определять виртуальные домены для поддержки всех типов обращения к серверу. Кроме того, значением опции NameVirtualHost может быть IP-адрес, связанный с сетевым интерфейсом; в этом случае конфигурация основного сервера применяется ко всем запросам, за исключением запросов, пере-

данных через этот интерфейс, и запросов, соответствующих определению виртуального узла.

- **<VirtualHost>**. Данная директива указывает на начало блока, содержащего определение виртуального домена. Для этой директивы задаются те же значения, что и для директивы **NameVirtualHost**. Признаком окончания блока служит директива **</VirtualHost>**. В состав блока включаются директивы, определяющие конфигурацию виртуального домена; здесь вы можете указать многие из тех директив, которые используются для настройки сервера, не поддерживающего виртуальные узлы.

В составе блока, сформированного с помощью <VirtualHost>, обычно указываются директивы ServerName (она определяет **имя,** которому соответствует данный блок) и **DocumentRoot**. При необходимости вы также можете настроить другие характеристики сервера, например разрешить выполнение **CGI-сценариев.** В качестве примера рассмотрим следующий фрагмент конфигурационного файла, который описывает два виртуальных Web-узла:

```
NameVirtualHost *

<VirtualHost *>
    ServerName www.threeroomco.com
    DocumentRoot /home/httpd/threeroomco/html
    ScriptAlias /cgi-bin/ "/home/httpd/threeroomco/cgi-bin/"
</VirtualHost>

<VirtualHost *>
    ServerName www.pangaea.edu
    DocumentRoot /home/httpd/pangaea-u/html
</VirtualHost>
```

Если сервер настроен подобным образом, то при обращении к нему посредством имени www.**threeroomco.**com он будет предоставлять клиенту статические файлы, которые находятся в каталоге /home/httpd/threeroomco/html, или запускать на выполнение сценарии, содержащиеся в каталоге /home/httpd/threeroomco/cgi-bin. Если же в запросе указано имя **www.pangaea.edu,** то статические файлы будут извлекаться из каталога /home/httpd/pangaea-u/html, а выполнение CGI-сценариев будет запрещено.

В отличие от **VirtualDocumentRoot,** использование директивы <VirtualHost> позволяет настроить каждый виртуальный узел и размещать документы в произвольных позициях файловой системы. С другой стороны, VirtualDocumentRoot предельно упрощает добавление новых доменов; для этого достаточно создать новый каталог. В большинстве случаев администраторы предпочитают использовать директиву <VirtualHost>, однако вы можете выбрать любой из этих способов, исходя из особенностей поставленной перед вами задачи.

Создание содержимого Web-узла

Несмотря на то что данная глава в основном посвящена особенностям настройки и выполнения Web-сервера, администратору, осуществляющему поддержку Web-сервера, необходимо представлять себе, как создаются документы, которые размещаются на Web-узле. Некоторые типы Web-страниц (точнее, средства для их динамической генерации) рассматривались в предыдущих разделах, однако основную часть данных, расположенных на Web-узлах, составляют HTML-документы. Для создания HTML-документов, а также файлов, которые могут использоваться ими (например, файлов с графическими данными), часто применяются специальные инструментальные средства. Научившись работать с этими инструментами и зная особенности интерпретации HTML-кода клиентскими программами, вы сможете без труда создать Web-узел, пригодный для просмотра с помощью наиболее популярных современных броузеров.

Форматы данных, используемых при создании Web-узла

Несмотря на наличие специализированных инструментальных средств, необходимо знать форматы основных данных, применяемых при создании Web-узлов. Как правило, основное содержимое Web-узла составляют статические Web-страницы, включающие текстовую и графическую информацию.

Основу большинства Web-страниц составляет HTML-файл. Этот файл содержит текстовые данные, пригодные для редактирования с помощью обычного текстового редактора. Пример простого HTML-файла приведен в листинге 20.2. Данные, содержащиеся в HTML-файле, делятся на две категории: текст, предназначенный для отображения в окне броузера, и последовательности символов, помещенные в угловые скобки, называемые *дескрипторами*. Дескрипторы представляют собой элементы форматирования, а также выполняют некоторые другие функции. Большинство дескрипторов используются парами, каждая из которых состоит из открывающего и закрывающего дескрипторов. Открывающий и закрывающий дескрипторы имеют одно и то же имя, но перед именем закрывающего дескриптора указывается символ /. В состав открывающего дескриптора часто входят *атрибуты,* уточняющие действия дескриптора. Например, с помощью атрибутов могут задаваться размеры изображения и содержащий его файл, цвет фона и текста и т. д. Некоторые из дескрипторов формируют ссылки на документы, расположенные на том же сервере, либо на других Web-серверах.

Назначение некоторых из дескрипторов, приведенных в листинге 20.2, очевидно, другие требуют более подробного рассмотрения. Ниже представлено описание дескрипторов, наиболее часто встречающихся в HTML-документах.

- <HTML>. Данный дескриптор сообщает о том, что документ является HTML-документом. Большинство броузеров не требует наличия этого дескриптора, но желательно указывать его, так как он предусмотрен спецификацией языка.

- <HEAD>. HTML-документ делится на заголовок и тело документа. В заголовке в основном содержится информация, не предназначенная для отображения (за исключением содержимого элемента <TITLE>). Заголовок содержится между открывающим и закрывающим дескриптором <HEAD>.

Листинг 20.2. Пример HTML-файла

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTMLXHEAD>
<TITLE>Пример Web"=страницы</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#000000">
<CENTERXH1 ALIGN="CENTER">Пример Web"=страницы</H1></CENTER>
<IMG SRC="graphics/logo.jpg" ALT="Logo" WIDTH="197"
HEIGHT="279"> <P>Данная Web"=страница содержит <A
HREF="http://www.threeroomco.com/anotherpage.html">
гипертекстовую ссылку.</A></P>
</BODYX/HTML>
```

- <TITLE>. Строка, заданная с помощью этого дескриптора, выводится в заголовке окна броузера. Эта же строка отображается в списке закладок,

- <BODY>. С помощью данного дескриптора формируется тело HTML-документа. В состав дескриптора <BODY> часто включают атрибуты, определяющие цвет текста и фона, и другие характеристики документа.

- <H1>. Заголовки позволяют делить текст документа на разделы и, как правило, отображаются шрифтом большего размера, чем обычный текст. Создавая код Web-страницы, вы можете включать в него заголовки различных уровней. Наивысшим считается уровень 1 (<H1>), а самым низким — уровень 6 (<Нб>). В листинге 20.2 в дескрипторе <H1> содержится атрибут **ALIGN**, который сообщает Web-броузеру о том, что текст заголовка должен быть размещен по центру экрана. К сожалению, не все броузеры правильно обрабатывают атрибут выравнивания в составе заголовка, поэтому, чтобы обеспечить корректное отображение информации, его приходится дублировать дескриптором <CENTER>.

- <CENTER>. В листинге 20.2 заголовок, формируемый с помощью дескриптора <H1>, выравнивается по центру экрана не только посредством атрибута ALIGN, но и с помощью дескриптора <CENTER>. Во многих современных броузерах такая избыточность не нужна, но если вы хотите, чтобы документ корректно отображался в старых броузерах, вам следует задавать как дескриптор <CENTER>, так и атрибут ALIGN.

- <IMG>. Данный дескриптор позволяет включать на Web-страницу графические изображения. Пример использования дескриптора <IMG> приведен в листинге 20.2. Обычно в дескриптор <IMG> включают различные атрибуты. Атрибут SRC указывает на файл, содержащий изображение; если изображение хранится на том же сервере, значением атрибута является имя файла, а если файл с изображением находится на другом сервере, то в качестве значения SRC задается абсолютный URL этого файла. Атрибут ALT задает текст, описывающий изображение. Этот текст отображается броузерами, в которых запрещен вывод **изображений,** а также выводится на экран при помещении на изображение курсора мыши. Атрибуты WIDTH и HEIGHT

задают ширину и высоту изображения, что позволяет броузеру отображать текст документа еще до того, как загрузка изображения закончится.

- <P>. Данный дескриптор определяет начало абзаца. Web-броузер автоматически переносит текст, достигший правого края окна, на новую строку.

- <A HREF>. С помощью дескриптора <A> создается гипертекстовая ссылка (при этом URL документа, на который указывается ссылка, задается в качестве значения атрибута HREF). Текст ссылки выделяется в окне броузера цветом и подчеркиванием. После щелчка мышью на ссылке в окне броузера отображается документ, URL которого задан посредством атрибута HREF.

Пользуясь этими дескрипторами, можно создать простейшую Web-страницу. Кроме них, в языке HTML определены многие другие дескрипторы и атрибуты, позволяющие форматировать таблицы, задавать шрифты, формировать маркированные и нумерованные списки, разбивать окно броузера на части, называемые фреймами, и отображать во фреймах различные документы, а также выполнять многие другие действия. Проблема с использованием расширенных средств HTML состоит в том, что некоторые дескрипторы по-разному интерпретируются различными броузерами. Этот вопрос будет более подробно рассмотрен позже в данной главе.

Помимо HTML-файлов, Web-серверы могут предоставлять клиентским программам и другие типы документов. Так, например, в листинге 20.2 был приведен пример изображения, включаемого на Web-страницу с помощью дескриптора <IMG>. В документе можно создавать гипертекстовые ссылки, указывающие на текстовые и графические файлы, исполняемые программы, сценарии и другие типы данных. Необходимо лишь, чтобы сервер мог определить MIME-тип каждого из документов. Для этого используется файл `mime.types`, который рассматривался ранее в этой главе. Если сервер Apache не может определить MIME-тип файла, он передает данные как неформатированный текст. Это становится источником проблем при работе некоторых операционных систем, так как специальные символы, находящиеся в составе файла, могут разрушить изображение на экране.

Поскольку многие Web-страницы содержат графические изображения, необходимо рассмотреть графические форматы, используемые в Web. Эти форматы описаны ниже.

- GIF. Graphics Interchange Format (формат обмена графическими данными) приобрел популярность в 1980-х. В данном формате используется схема сжатия без потери информации. Это означает, что изображение, полученное после распаковки, будет в точности совпадать с исходным изображением. Для представления цвета в GIF-изображениях используется до 8 битов, т. е. такое изображение может содержать максимум 256 цветов.

- PNG. Portable Network Graphic (переносимые сетевые графические данные) также использует схему сжатия без потери информации. В отличие от GIF, PNG позволяет представлять цвет посредством большего количества битов (обычно применяется 24-битовое представление, но PNG дает возможность использовать для этой цели до 64 битов). Недостатком PNG является тот факт, что данный формат поддерживается не всеми броузерами. Более подробную информацию о PNG можно получить по адресу `http://www.libpng.org/pub/png/`.

- JPEG. В формате Joint Photographic Expert Group (объединенная группа экспертов по обработке фотоснимков) используется сжатие с потерей информации. В результате достигается большая степень сжатия по сравнению с форматами GIF и PNG, но распакованное изображение отличается от исходного. Для представления цвета в JPEG может применяться до 24 битов.

Как правило, форматы, использующие сжатие без потери информации, лучше подходят для представления чертежей, рисунков и других подобных изображений. При переводе в формат JPEG такие изображения искажаются. Оцифрованные фотоснимки лучше представлять в форматах, позволяющих кодировать цвет посредством большого числа битов (например, PNG или JPEG). Потеря информации при JPEG-сжатии практически не влияет на качество фотоснимка.

При создании JPEG-изображений графический пакет позволяет выбрать степень сжатия. Низкая степень сжатия приводит к тому, что размер файла оказывается очень большим. Высокая степень сжатия позволяет получить файл небольшого размера, но качество изображения становится недопустимо низким. Выбирая степень сжатия графических файлов, предназначенных для представления в Web, необходимо, с одной стороны, обеспечить приемлемое качество изображения, а с другой стороны, добиться, чтобы время его загрузки было не слишком большим.

Инструментальные средства создания Web-страниц

Несмотря на то что HTML-документы можно создавать с помощью обычных текстовых редакторов, многие Web-дизайнеры предпочитают использовать для этой цели специализированные инструменты с графическим пользовательским интерфейсом. Данные инструменты позволяют редактировать текст документа подобно текстовым процессорам WYSIWYG (what you see is what you get); при этом для выравнивания, создания абзацев, отображения текста полужирным шрифтом и выполнения других подобных действий используются кнопки, расположенные на панели инструментов. Поскольку при работе сервера Apache не имеет значения, каким способом были сформированы Web-страницы, нет никаких оснований отказываться от использования специализированных инструментальных средств. Единственным исключением является редактор Microsoft Front Page. Этот инструмент создает Web-страницы, ориентированные на конкретный сервер, поэтому при работе с Apache лучше отказаться от его использования.

Некоторые инструменты, предназначенные для создания HTML-документов, предоставляют средства автоматического копирования созданных файлов на сервер. При использовании сервера Apache эти средства могут работать некорректно, поэтому желательно сохранять HTML-файлы на локальном диске, а затем копировать их на сервер с помощью FTP.

Ниже описаны некоторые инструменты, которые могут быть использованы при создании Web-страниц.

- Текстовые процессоры. Многие современные текстовые процессоры предоставляют возможность экспортировать документы в формате HTML. (Поскольку средства форматирования текстовых процессоров мощнее, чем соответствующие средства, предусмотренные в языке HTML, при сохранении документа в виде HTML-файла внешний вид текста может несколько измениться.) Для тех, кто привык работать

с текстовыми процессорами, они могут стать удобными инструментами подготовки Web-страниц. В системе Linux возможность экспортирования в HTML-формат предоставляют программы Applix Words, StarOffice и WordPerfect.

- **Web-броузеры.** Многие популярные Web-броузеры, выполняющиеся в системе Linux, в частности Netscape, содержат средства для подготовки HTML-документов. Такие средства лучше подходят для создания Web-страниц, чем текстовые процессоры.

- **Независимые программы подготовки Web-страниц.** Единственным назначением этих инструментов является создание HTML-документов. В качестве примеров подобных программ, работающих в системе Linux, можно привести ASHE (`http://www.cs.rpi.edu/pub/puninj/ASHE/`), August (`http://www.lls.se/~johanb/august/`), Bluefish (`http://bluefish.openoffice.nl`)и WebSphere (`http://www-4.ibm.com/software/webservers/hpbuilder/`). Некоторые из этих инструментов очень просты, другие позволяют выполнять достаточно сложные действия.

Используя специализированные инструменты подготовки Web-страниц, необходимо учитывать, что каждый из них ориентирован на определенный Web-броузер. Поскольку один и тот же документ по-разному выглядит в различных броузерах, Web-страницы, созданные с помощью некоторого инструмента, могут некорректно отображаться посредством ряда клиентских программ.

Особенности создания Web-страниц

Некоторые Web-дизайнеры стараются в полной мере использовать возможности, предусмотренные спецификацией HTML, и создают HTML-документы, напоминающие страницы печатных изданий. Однако использование расширенных средств HTML может стать источником проблем. Дело в том, что предсказать, как тот или иной броузер будет обрабатывать некоторый фрагмент HTML-кода, практически невозможно. Даже предельно простая Web-страница, код которой показан в листинге 20.2, может по-разному выглядеть в различных броузерах. Как было замечено ранее, некоторые броузеры не обрабатывают дескриптор <CENTER>, другие игнорируют атрибут ALIGN в составе заголовка. Шрифт, указанный в документе, будет отображаться только в том случае, если он присутствует на компьютере, на котором выполняется Web-броузер, в противном случае содержимое документа будет воспроизведено с использованием шрифта по умолчанию. Цвет, заданный для отображения Web-страницы, должен сочетаться с цветами, выбранными пользователем. (Одна из ошибок, часто допускаемых Web-дизайнерами, связана с использованием цветов. Если задать цвет фона, не указав при этом цвет для отображения текста, могут возникнуть проблемы при выводе содержимого документа на экран. Предположим, например, что вы задали в документе белый цвет фона. Если при этом пользователь установил по умолчанию отображение белого текста на черном фоне, то прочитать текст документа в окне броузера будет невозможно. В листинге 20.2 заданы цвет фона и переднего плана, но не указан цвет для отображения гипертекстовых ссылок. Это также может стать источником проблем при выводе документа.)

Поскольку броузеры по-разному интерпретируют HTML-код, желательно проверять созданные вами Web-страницы в различных броузерах. Как минимум вы должны выяснить, как отображаются ваши документы в наиболее популярных на сегодняшний день

клиентских Web-программах: Netscape Navigator и Microsoft Internet Explorer. По возможности следует проверить качество воспроизведения документа в различных версиях этих броузеров. Следует также учитывать, что в последнее время среди пользователей Linux стали популярны броузеры Mozilla (`http://www.mozilla.org`, вариант Netscape Navigator, распространяемый в исходных кодах), Opera (`http://www.opera.com`) и Konqueror (созданный в рамках проекта KDE). Особого внимания заслуживает броузер Lynx (`http://lynx.browser.org`), отображающий лишь текстовую информацию. Если вы хотите, чтобы ваши Web-страницы были доступны всем пользователям, необходимо протестировать их с помощью данного броузера. Несмотря на то что Lynx в настоящее время практически не используется, он позволяет выявить большинство проблем, которые не очевидны при работе с другими броузерами. Обеспечив воспроизведение Web-страницы посредством клиентской программы, отображающей лишь текст, вы упростите работу с Web-документами пользователей с нарушениями зрения, применяющих синтезаторы речи. Необходимо также учитывать интересы пользователей, работающих с различными операционными системами. В системе Windows наиболее популярным является броузер Internet Explorer, в других системах, например MacOS, BeOS и OS/2, используются и другие клиентские программы. Некоторые из них работают на различных платформах, другие ориентированы на выполнение в конкретной операционной системе.

Просматривая файлы протоколов, вы сможете определить, какие типы броузеров чаще всего применяют пользователи, обращающиеся на ваш Web-узел.

Анализ файлов протоколов

Файлы протоколов — важный источник сведений о работе Web-сервера. Информация, содержащаяся в этих файлах, поможет вам администрировать Web-узел. Файлы протоколов включают информацию о клиентских программах, обращающихся к серверу, о документах, запрашиваемых клиентами, о времени обращений и другие сведения. Для анализа содержимого файлов протоколов вручную требуется много времени и усилий, поэтому большинство администраторов делают это с помощью специализированных инструментов, которые упрощают обработку данных. Наиболее часто для этой цели применяются программы Analog и Webalizer.

В данном разделе рассматриваются стандартные файлы протоколов Apache, записью информации в которые управляет директива `CustomLog`. Сервер Apache также поддерживает дополнительные файлы протоколов, в которые помещается информация об ошибках, сообщения, генерируемые в процессе загрузки, и другие данные.

Формат файла протокола Apache

Данные могут записываться в файл протокола Apache в различных форматах; конкретный формат задается с помощью директивы CustomLog. В данном разделе описывается формат combined, который объединяет в одном файле различные данные. Запись в формате combined выглядит следующим образом:

```
192.168.1.1 - - [06/Nov/2002:16:45:49 -0500] "GET /index.html \
HTTP/1.0" 200 8597 "-" "Mozilla (X11; I; Linux 2.0.32 i586)"
```

Эта запись включает следующие компоненты.

- **Доменное** имя или IP-адрес клиента. Первое поле записи содержит адрес или имя клиента, от которого был получен запрос.
- Идентификатор пользователя. Следующие два поля содержат имя пользователя, инициировавшего запрос. В первом из этих полей содержится имя для сервера `identd`, а во втором — имя для HTTP-аутентификации. (В данном примере в этих полях указаны дефисы, указывающие на то, что сведения о пользователе не доступны.)
- Дата и время. Apache записывает дату и время передачи запроса. В этой записи содержится локальное время с указанием временного пояса (в данном примере `-0500`).
- HTTP-запрос. HTTP-запрос включает команду, переданную клиентом (`GET`), запрашиваемый документ (`/index.html`) и версию протокола HTML (`1.0`). С помощью этой информации можно определить, какие из документов, расположенных на вашем Web-узле, наиболее популярны среди пользователей.
- Код ответа. В ответ Apache включает цифровой код, информирующий клиентскую программу о результатах обработки запроса. В данном примере указан код ответа 200, это означает, что обработка запроса окончилась успешно. Коды, начинающиеся с цифры 3, означают перенаправление запроса, а коды, начинающиеся с цифры 4 или 5, свидетельствуют об ошибке.
- **Размер объекта.** Число 8597 соответствует размеру ресурса, который Apache передал клиенту в ответ на запрос. При вычислении размера объекта заголовок ответа не учитывается.
- **Документ, ссылающийся на текущий ресурс.** Если пользователь запросил новый ресурс щелчком на гипертекстовой ссылке в составе HTML-документа, броузер передает серверу в составе запроса URL этого документа. Полученные сведения Apache записывает в файл протокола. В приведенном выше примере в данном поле содержится дефис, означающий, что документ, ссылающийся на запрашиваемый ресурс, не определен. Это может быть в случае, если пользователь непосредственно ввел URL ресурса в поле адреса броузера.
- **Клиентская программа.** Последнее поле записи содержит информацию о броузере, а также сведения об операционной системе, в которой он выполняется. (Заметьте, что броузер Netscape сообщает о себе с помощью идентификатора `Mozilla`.) На сведения, указанные в этом поле, нельзя полагаться, поскольку клиентскую программу можно настроить так, чтобы она сообщала о себе неверные данные. Кроме того, сведения о броузере могут быть заменены proxy-сервером.

Информация, содержащаяся в файле протокола, позволяет сделать вывод о популярности ваших документов среди пользователей, о том, из каких сетей пользователи наиболее часто обращаются к вашему серверу, и получить другие необходимые сведения. Как уже было сказано ранее, анализ данных в файле протокола представляет собой достаточно

сложную задачу, для решения которой часто используются специализированные инструменты.

В большинстве версий Linux инструмент **cron** по умолчанию настраивается так, чтобы через определенные промежутки времени осуществлялась *ротация* файлов протоколов (переименование файлов протоколов и удаление с диска старых файлов). Соответствующая задача для **cron** обычно описывается в каталоге **/etc/cron.d** или **/etc/cron.interval**. Если в вашей системе ротация файлов не выполняется, вам надо создать соответствующую задачу **cron**, в противном случае размеры файлов станут слишком большими, а это может привести к переполнению диска.

Использование Analog

Analog (**http://www.analog.cx**) является наиболее популярным из инструментов, предназначенных для анализа файлов протоколов. Этот инструмент в основном отображает результаты анализа в текстовом виде, но может также представлять их в виде диаграмм. С примером отчета, сгенерированным Analog, можно ознакомиться, обратившись по адресу **http://www.statslab.cam.ac.uk/~sret1/stats/stats.html**. Инструмент Analog входит в состав некоторых дистрибутивных пакетов. Если в вашей системе Analog отсутствует, вы можете скопировать его с Web-узла.

Настройка программы Analog

Работой программы Analog управляет конфигурационный файл **analog.cfg**, который обычно размещается в каталоге /etc. Этот файл содержит опции, задавая значения которых вы можете представлять данные, генерируемые Analog, в удобном для вас виде. Например, опция **SEARCHENGINE** задает поисковые серверы, которые могут ссылаться на ваши документы. С помощью этой опции Analog может учитывать ссылки на содержимое Web-узла, находящиеся на поисковых серверах. При настройке программы Analog вам придется задать следующие опции:

```
LOGFILE   путь_к_файлу_протокола
OUTFILE   путь_к_файлу_содержащему_выходные_данные
HOSTNAME  "имя_организации"
```

Первые две из приведенных выше опций особенно важны. Если вы не укажете их, Analog не сможет найти файл протокола, а выходная информация будет непосредственно передаваться в стандартный выходной поток. Analog генерирует выходные данные в формате HTML и включает в созданный им файл графические изображения. Таким образом, вы можете просмотреть результаты обработки файла протокола с помощью Web-броузера. (При настройке Analog необходимо указать лишь имя основного HTML-файла, например **httpd/html/analog/index.html**; графические данные будут размещены в том же каталоге.) Опция HOSTNAME не оказывает существенного влияния на работу Analog. Ее значение лишь отображается в начале отчета.

К сожалению, некоторые пакеты Analog не являются полнофункциональными, в частности, в них принимаются специфические и часто противоречащие друг другу предположения о размещении файлов. Для того чтобы разрешить эту проблему, необходимо создать несколько символьных ссылок.

- **Конфигурационный файл.** При создании некоторых пакетов Analog считается, что файл `analog.cfg` должен находиться в том же каталоге, что и исполняемый файл Analog (т. е. в каталоге `/usr/bin`), однако чаще всего конфигурационный файл размещается в каталоге `/etc`. Очевидно, что каталог `/usr/bin` — не самое подходящее место для конфигурационного файла, поэтому, чтобы обеспечить работу Analog с файлом, находящимся в каталоге /etc, необходимо выполнить команду `ln -s /etc/analog.cfg /usr/bin`.

- **Файлы поддержки языка.** Для того чтобы программа Analog выполнялась корректно, она должна иметь доступ к файлам поддержки языка. Некоторые пакеты размещают эти файлы в `/var/lib/analog/lang`, но Analog ищет их в каталоге `/usr/bin/lang`. Чтобы разрешить это противоречие, надо выполнить команду `ln -s /var/lib/analog/lang /usr/bin`.

- **Поддержка графики.** При обработке содержимого файлов протоколов Analog генерирует графические изображения, в частности диаграммы. Графические данные создаются для каждого узла, но Analog использует для записи информации файлы с фиксированными именами. В некоторых пакетах по умолчанию предусмотрено размещение этих файлов в каталоге `/var/www/html/images`, но в документах, сгенерированных при выполнении Analog, содержатся ссылки, которые указывают на файлы, находящиеся в подкаталоге `images` текущего каталога. Чтобы обеспечить доступ к графическим файлам, необходимо создать еще одну символьную ссылку, выполнив для этого команду `ln -s /var/www/html/ images`.

Указанные здесь изменения нужны лишь для некоторых пакетов. В частности, их необходимо выполнить при использовании пакета `analog-5.01-1mdk` в системе Mandrake.

Запуск программы Analog

Для запуска программы Analog на выполнение необходимо ввести команду `analog`. Пользователь, вызывающий эту команду, должен иметь право читать содержимое файла протокола и иметь право записи в тот каталог, в который Analog помещает свои выходные данные. Таким образом, при наличии необходимых полномочий запускать Analog можно от имени обычного пользователя.

В некоторых случаях возникает необходимость в периодическом запуске Analog (раз в неделю, раз в месяц или даже раз в день). Сделать это можно с помощью инструмента `cron`. При этом необходимо помнить, что Analog потребляет не очень большие, но все же значительные ресурсы, поэтому если запускать данную программу слишком часто (например, каждую минуту), это непременно скажется на производительности системы.

Интерпретация выходных данных Analog

Выходные данные Analog представляют собой сочетание различных отчетов. Каждый из них содержит информацию, которая была создана в результате некоторой операции по обработке файла протокола, и помещается в отдельном разделе. Назначение основных разделов выходного файла описано ниже.

- **Обобщенная сводка.** В этом разделе представлена общая информация, используемая для оценки состояния Web-сервера: среднее количество запросов, обрабатываемых в течение дня, среднее число запросов, при обработке которых возникли

ошибки, общий объем переданных данных и средний объем данных, переданных в течение дня.

- **Ежемесячный отчет.** В ежемесячном отчете указывается число документов, обработанных в течение месяца. Увеличение числа обращений в течение месяца и снижение производительности системы указывает на то, что вам необходимо перенести сервер на более мощный компьютер или увеличить пропускную способность соединения.

- **Ежедневный отчет.** В этом разделе указывается число документов, обработанных в течение определенного дня недели (понедельник, вторник и т. д.).

- **Почасовые отчеты.** В данном разделе приводится информация о работе сервера в течение каждого часа текущего дня. Если вам необходимо выполнять какие-либо действия, потребляющие ресурсы сервера, желательно выбрать для этого время, в течение которого сервер наименее загружен. В этом вам поможет информация, приведенная в данном разделе.

- **Отчет о работе с доменами.** Если ваш сервер поддерживает несколько доменов, просмотрев данный раздел, вы ознакомитесь с трафиком, связанным с каждым доменом.

- **Отчет об использовании серверов различных организаций.** Если ваш сервер поддерживает виртуальные узлы для различных организаций, этот отчет предоставляет сведения о трафике, связанном с работой сервера каждой организации.

- **Отчет о работе с операционными системами.** Если в вашем файле протокола содержится информация об операционных системах, в которых работают клиенты, обращающиеся к серверу, эта информация учитывается в данном отчете. Следует заметить, что из-за наличия proxy-серверов приведенные здесь данные не всегда отражают реальную ситуацию.

- **Отчет о кодах состояния.** В данный раздел Analog включает диаграмму, представляющую соотношение различных кодов состояния, передаваемых Web-сервером в составе ответов клиентам. Если коды 4.xx и 5.xx появляются во многих ответах, необходимо найти и устранить причину подобного поведения сервера.

- **Отчет о размерах файлов.** Данный раздел содержит сведения о числе файлов разных размеров, которые Web-сервер предоставляет пользователям. Информация, приведенная в этом отчете, может быть использована для управления трафиком. Если вы обнаружите, что средний размер передаваемых файлов увеличивается, вам надо принять соответствующие меры, например увеличить степень сжатия графических изображений.

- **Отчет о типах файлов.** В данном отчете сообщается о типах файлов (JPEG, HTML и т. д.), предоставляемых Web-сервером. Эта информация может быть использована для тех же целей, что и сведения, содержащиеся в отчете о размерах файлов.

- **Отчет о каталогах.** На большинстве Web-узлов информация хранится в различных каталогах. Данный отчет предоставляет сведения о том, в каких каталогах находится

информация, пользующаяся наибольшим успехом (решение о популярности того или иного каталога принимается на основании объема переданных данных).

- **Отчет о запросах.** Данный отчет содержит сведения об использовании файлов, находящихся в корневом каталоге Web-узла.

Информация, приведенная в различных отчетах, позволяет составить представление о работе Web-узла. Еще более полные сведения вы можете получить, собрав данные, сгенерированные Analog в течение определенного периода времени. Для этого надо организовать ротацию файлов протоколов Apache, причем в процессе ротации необходимо копировать выходные данные Analog в специальный подкаталог. Ротацию и копирование файлов Analog можно реализовать с помощью инструмента `cron`. Кроме того, вам придется создать главный HTML-документ, ссылающийся на данные Analog, скопированные в процессе ротации в выбранные вами каталоги. В результате вы получите информацию, собранную Analog в течение нескольких недель или месяцев.

Несмотря на то что Analog является чрезвычайно полезным инструментом, следует все же признать, что обработка данных, сгенерированных в процессе выполнения этой программы, — трудоемкое занятие, требующее ненамного меньше усилий, чем непосредственный анализ файлов протоколов Apache. Для дальнейшей обработки данных, сгенерированных Analog, и представления их в форме, удобной для восприятия, используются дополнительные инструменты. В качестве примера такого инструмента можно привести Report Magic (`http://www.reportmagic.com`).

Использование Webalizer

Инструмент Webalizer (`http://www.webalizer.org`) предоставляет администратору приблизительно такие же возможности, как и Analog. Подобно Analog, Webalizer читает содержимое конфигурационных файлов и создает выходной HTML-файл, представляя сведения о Web-узле в удобном для восприятия виде. Webalizer поставляется в составе некоторых дистрибутивных пакетов. Если же в вашей системе этот инструмент отсутствует, вы можете скопировать его с Web-узла, содержащего сведения о данном инструменте и его код. Пример выходного файла, сгенерированного Webalizer, расположен по адресу `http://www.webalizer.org/sample/`.

Настройка Webalizer

Работой Webalizer управляет конфигурационный файл `webalizer.conf`, который обычно располагается в каталоге /etc. Как и при работе с Analog, вы должны сообщить Webalizer о том, где находится файл протокола Web-сервера и в какой каталог следует записывать выходные данные. Для этого используются следующие опции:

```
LogFile  путь_к_файлу_протокола
OutputDir  путь_к_каталогу_содержащему_выходные_данные
```

Одно из отличий между Analog и Webalizer состоит в том, что в конфигурационном файле Analog задается имя выходного файла, а для Webalizer указывается каталог, в который данная программа будет записывать сгенерированные в процессе работы файлы. Если этот каталог доступен Web-серверу, то вы можете просматривать информацию, созданную Webalizer, с помощью Web-броузера. Если же вы разместите каталог за пределами области, к которой может обращаться Web-сервер, то для просмотра результатов работы

Webalizer вам придется задавать URL, начинающийся с `file://`. Ниже перечислены некоторые опции Webalizer, значения которых вы, возможно, захотите изменить.

- Incremental. Если для данной опции задано значение yes, Webalizer сохраняет от запуска к запуску информацию о своем состоянии. Это позволяет обрабатывать файл протокола по частям. Предположим, например, что вы запускаете Webalizer один раз в день. При установленном значении yes опции Incremental он будет помнить, какую запись он уже обработал, и сможет учитывать при этом ротацию файлов. Если значение данной опции равно no, Webalizer будет при каждом запуске обрабатывать весь файл протокола.

- HostName. Эта опция позволяет задать имя узла, отображаемое в заголовке отчета (содержание заголовка отчета определяется опцией ReportTitle).

- `GroupDomains`. Данная опция позволяет объединять доменные имена в группы. Значение опции указывает, какое количество компонентов доменного имени должно идентифицировать группу. Предположим, что значение GroupDomains равно 2. В этом случае доменные имена `gingko.pangaea.`edu и `birch.pangaea.edu` будут объединены в группу `pangaea.edu`. Данная опция позволяет упорядочивать данные, генерируемые Webalizer.

- GroupSite. Данная опция также предназначена для поддержки групп. Например, значение GroupSite `*.abigisp.`net объединяет в одну группу все узлы, принадлежащие домену `abigisp.net`.

- HideSite. Эта опция исключает узлы из группы, созданной с помощью GroupSite. Опции GroupSite и HideSite используются совместно.

Конфигурационные файлы Webalizer имеют больший объем и более сложную структуру по сравнению с соответствующими файлами Analog. Выше была перечислена лишь незначительная часть опций, предназначенных для настройки Webalizer. Большинство опций снабжено подробными комментариями, прочитав которые вы можете легко составить представление о назначении этих опций.

Запуск Webalizer

Для запуска Webalizer надо ввести команду webalizer. Как и при работе с программой Analog, чтобы запустить Webalizer, не надо обладать полномочиями пользователя root. Достаточно иметь право читать содержимое файла протокола и записывать информацию в каталог, предназначенный для выходных данных Webalizer. Возможно, вам потребуется организовать запуск Webalizer по расписанию с помощью инструмента `cron`. В этом случае целесообразно установить значение yes опции Incremental.

Часто при инсталляции Apache, формируется задача для инструмента `cron` на выполнение ротации файлов. Если ротация не предусмотрена, вам надо организовать ее самостоятельно. Чтобы обеспечить максимальное использование входных данных, желательно запускать Webalizer не только по графику, но и непосредственно перед выполнением ротации.

Интерпретация выходных данных Webalizer

Выходные данные, генерируемые в процессе работы Webalizer, представляют собой двухуровневую структуру. На первом уровне выводится информация об активности серве-

ра за прошедший год. (Для вновь инсталлированного сервера большинство месяцев будут пустыми.) Информация, соответствующая первому уровню, отображается как в виде таблиц, так и в виде диаграмм и представляет сведения о переданных Web-страницах, объеме скопированной информации и другие данные для каждого месяца. После щелчка на названии месяца отобразится Web-страница, соответствующая второму уровню данных, на котором отображаются более подробные сведения для выбранного месяца. На этой странице присутствуют следующие разделы.

- **Статистика работы сервера за месяц.** В этом разделе отображается та же информация, что и на первом уровне, дополненная сведениями о кодах, содержащихся в ответах клиентам.

- **Статистика работы сервера за день.** Во втором разделе Web-страницы содержатся диаграмма и таблица, представляющие сведения о Web-трафике в течение каждого дня месяца. Здесь выводятся данные о количестве обращений, числе скопированных файлов и объеме переданных данных в килобайтах.

- **Почасовая статистика.** Здесь представлены те же данные, что и в предыдущем разделе, но они разбиты по часам суток. С помощью этой информации вы можете выяснить, в какие часы нагрузка на сервер была максимальна, и принять меры для более равномерного ее распределения.

- **URL наиболее популярных документов.** Webalizer отображает две таблицы, в которых содержится информация о числе обращений и объеме скопированных данных для различных URL. (При необходимости вы можете объединять URL в группы, используя для этого рассмотренные ранее опции.) Одна таблица отображает URL документов, к которым было максимальное количество обращений, другая — URL, в результате обращения к которым были скопированы данные наибольшего объема.

- **Входная и выходная страницы.** Следующие две таблицы, генерируемые Webalizer, содержат сведения о наиболее популярной входной и выходной страницах. *Входной страницей* (entry page) называется документ, к которому обращается пользователь в начале работы с узлом, а *выходной страницей* (exit page) — документ, который пользователь просматривает непосредственно перед завершением работы.

- **Наиболее популярные узлы.** Webalizer записывает сведения о взаимодействии клиентов с Web-узлами, поддерживаемыми сервером, учитывая как число обращений, так и объем скопированной информации. При необходимости вы можете объединять узлы в группы с помощью опции GroupSite, расположенной в конфигурационном файле Webalizer.

- **Документы, наиболее часто ссылающиеся на содержимое сервера.** Анализируя информацию в файле протокола, Webalizer предоставляет сведения об узлах, наиболее часто ссылающихся на документы, расположенные на сервере.

- **Наиболее часто используемые ключевые слова.** Некоторые поисковые серверы, в базе данных которых учтены документы, расположенные на вашем Web-узле, включают в состав URL строку поиска. Webalizer выделяет эту информацию и представляет в отчете.

- **Наиболее популярные клиентские программы.** Webalizer учитывает типы Web-броузеров, наиболее часто обращающихся к вашему узлу.
- **Наиболее популярные страны.** В последнем разделе Webalizer предоставляет информацию об обращениях к документам из разных стран. Страна определяется на основании домена верхнего уровня, поэтому наряду с реальными странами на Web-странице встречаются US Commercial, Network и другие подобные имена.

На основании этой информации вы можете сделать вывод об особенностях обращений к вашему Web-узлу из Internet. Сравнивая соответствующие данные за различные месяцы, вы увидите, как интересующие вас характеристики изменяются со временем.

Резюме

Web-сервер — чрезвычайно важный компонент многих сетей, используемых как для внутреннего взаимодействия, так и для предоставления данных внешним пользователям. Чаще всего Web-серверы передают клиентам статическую информацию, но, используя формы и сценарии, можно организовать двунаправленный обмен данными и динамическую генерацию Web-страниц. Применение SSL позволяет защитить данные, передаваемые по сети, а механизм виртуальных доменов дает возможность разместить на одном компьютере несколько Web-узлов. Помимо настройки и запуска Web-сервера, поддержка Web-узла предполагает решение еще двух важных задач: создание Web-страниц и интерпретацию файлов протоколов. Web-страница может представлять собой как чрезвычайно простой документ, создаваемый с помощью текстового редактора, так и очень сложный набор файлов, для формирования которого приходится применять специализированные инструментальные средства. Файлы протоколов содержат информацию о работе Web-сервера. С помощью этих файлов можно определить, насколько популярен Web-узел среди пользователей, выявить и решить проблемы и узнать, какие из внешних документов ссылаются на содержимое вашего узла. На основании анализа файлов протоколов строятся планы по дальнейшему совершенствованию Web-узла.

Глава 21

FTP-серверы

Протокол FTP (File Transfer Protocol — протокол передачи файлов) существует давно и пользуется большой популярностью в Internet. Он обеспечивает обмен файлами между компьютерами, подключенными к сети. **FTP-клиенты** могут копировать файлы с сервера и, если позволяет конфигурация сервера, передавать информацию на сервер. Иногда FTP-сервер может заменить Web-сервер или файловый сервер, но в большинстве случаев его можно рассматривать как дополнение к этим средствам. В ряде ситуаций наличие FTP-сервера на компьютере не оправдано. Если вы приняли решение установить FTP-сервер, можете воспользоваться для этой цели различными программами, ориентированными на работу в системе Linux. По умолчанию программы, реализующие FTP-серверы, настроены для выполнения определенного круга задач. Возможно, вам придется изменить конфигурацию сервера. В некоторых случаях возникает необходимость в *анонимном FTP-сервере,* который позволяет каждому желающему копировать файлы на свой компьютер.

Использование FTP-сервера

FTP-сервер имеет некоторое сходство с Web-сервером, рассмотренным в главе 20, а также с серверами Samba и NFS, предназначенными для разделения файлов (о них шла речь в главах 7 и 8). Все эти серверы позволяют передавать файлы с одного компьютера на другой и в некоторых ситуациях взаимозаменяемы. Однако каждый из этих протоколов имеет свои особенности, определяющие выбор сервера для решения определенного круга задач. Ниже описаны основные отличия FTP-сервера от серверов HTTP, Samba и NFS.

- Аутентификация. Для того чтобы пользователь мог работать с FTP-сервером, он должен зарегистрироваться на нем, указав пользовательское имя и пароль. (Исключением является анонимный FTP-сервер, который будет описан ниже.) При работе с Web-сервером аутентификация обычно не требуется, хотя существуют средства, позволяющие настроить Web-сервер для проверки имени и пароля. Что же касается серверов, предназначенных для разделения файлов, то некоторые из них проверяют имя пользователя и пароль, другие выполняют аутентификацию, анализируя лишь IP-адрес.

- **Использование учетных записей.** Особенности аутентификации, выполняемой FTP-сервером, позволяют применять его для предоставления пользователям доступа к принадлежащим им файлам с удаленных компьютеров. Подобным образом могут использоваться серверы NFS и Samba. Web-сервер также позволяет пользователям работать с их файлами, но тот же уровень доступа он предоставляет всем желающим. Для обеспечения защиты информации при работе с Web-сервером приходится принимать дополнительные меры.

- **Шифрование.** Стандартные FTP-серверы не выполняют шифрование данных, в том числе имени пользователя и пароля. Это создает опасность при передаче информации через Internet. Исключением в данном случае являются анонимные FTP-серверы, при использовании которых шифрование информации не имеет смысла. Существуют защищенные версии FTP, поддерживающие кодирование передаваемых данных. Защиту информации, которая передается средствами FTP, обеспечивают также средства Kerberos, которые рассматривались в главе 6. В обычных условиях Web-серверы не выполняют шифрование данных, хотя при необходимости такая возможность реализуется достаточно просто. Сервер Samba может быть настроен для шифрования паролей и прочей информации. NFS не использует пароли, поэтому о кодировании паролей речь не идет. Что же касается данных, передаваемых средствами NFS, при необходимости шифрование их может осуществляться. Программы scp и sftp, входящие в состав пакета SSH, кодируют все данные, поэтому их можно использовать вместо средств FTP в том случае, когда вопросы безопасности имеют большое значение.

СОВЕТ ✔ Если используемый вами пароль передается по Internet в незакодированном виде, вы должны регулярно изменять его. При этом злоумышленник, получивший пароль незаконным способом, будет иметь меньше возможностей нанести вред системе.

- **Поддержка соединения.** Подобно Samba и NFS, при обмене данными по протоколу FTP поддерживается постоянное соединение. Пользователь может зарегистрироваться на FTP-сервере, бездействовать в течение длительного времени (столько, сколько позволяет установленное значение тайм-аута), а затем скопировать файл с сервера. Web-серверы действуют совершенно по-другому; в рамках соединения может быть передан один, в крайнем случае несколько файлов. FTP коренным образом отличается от других протоколов подобного назначения тем, что при взаимодействии клиента и сервера используются два порта: один для передачи команд, а другой для передачи данных. Клиент обращается к серверу через порт 21. Затем в зависимости от режима (режим может быть *активным* или *пассивным)* передача данных осуществляется либо по инициативе клиента, либо по инициативе сервера. Такая особенность взаимодействия усложняет настройку брандмауэров, однако в большинстве программ, реализующих брандмауэры, предусмотрены специальные средства, позволяющие решить эту задачу.

- **Непосредственная обработка файлов.** Серверы, предназначенные для разделения файлов, например NFS и Samba, предоставляют возможность обрабатывать удаленные файлы так, как будто они расположены на локальном компьютере. Например, пользователь может загрузить файл с удаленной машины в текстовый редактор,

внести в него необходимые изменения и снова сохранить на удаленном компьютере. При этом файл на локальный диск не копируется. Ни FTP, ни HTTP не обеспечивают подобной возможности. Если доступ к удаленному компьютеру может осуществляться только посредством протокола FTP или HTTP, то, для того, чтобы отредактировать файл, вы должны сначала скопировать его на локальную машину, внести и сохранить изменения, а затем снова передать файл на сервер. Существуют средства, позволяющие организовать разделение файлов с помощью протокола FTP. В системе Linux такую возможность предоставляет продукт Linux FTP Filesystem (`http://ftpfs.sourceforge.net`). Однако подобное использование протокола FTP можно рассматривать скорее как исключение из общего правила.

- **Двунаправленная передача данных.** Серверы, предназначенные для разделения файлов, и FTP-сервер позволяют передавать данные с сервера на клиентскую машину и наоборот. При необходимости системный администратор имеет возможность запретить запись файлов на диск сервера. Web-серверы в основном применяются для передачи данных с сервера на клиентский компьютер, но в протоколе HTTP предусмотрена также возможность передавать информацию на сервер.

- **Работа клиентов на различных платформах.** FTP- и Web-серверы взаимодействуют с клиентами, выполняющимися в любой операционной системе, поддерживающей TCP/IP; не является исключением даже система DOS. В отличие от FTP и HTTP, протоколы разделения файлов ориентированы на конкретную платформу: сервер NFS работает с клиентами UNIX и Linux, а Samba — с клиентами DOS, Windows и OS/2. Пытаясь обеспечить межплатформенную совместимость протоколов разделения файлов, вы неизбежно столкнетесь с проблемами представления прав доступа, атрибутов файлов и другими ограничениями. При этом вам также приходится обеспечивать взаимодействие с клиентскими программами, многие из которых распространяются на коммерческой основе.

- **Трудозатраты при настройке сервера.** По умолчанию в системе Linux устанавливается конфигурация FTP-сервера, предоставляющая пользователям такие же права чтения и записи файлов, которые они получили бы, зарегистрировавшись на компьютере. Если вас устраивает подобное поведение сервера, его настройка займет крайне мало времени. Чтобы обеспечить подобное взаимодействие с серверами NFS, Samba и HTTP, вам придется внести существенные изменения в их конфигурационные файлы. С другой стороны, чтобы установить анонимный FTP-сервер, надо приложить определенные усилия для его настройки, в то время как в других серверах подобный принцип взаимодействия с пользователями реализован по умолчанию.

Таким образом, средства FTP удобно использовать для решения следующих двух задач.

- **Обслуживание локальных пользователей.** Если на компьютере под управлением Linux существуют учетные записи для некоторых пользователей, FTP-сервер позволит этим пользователям копировать свои файлы с сервера на удаленный компьютер либо с удаленной машины на сервер. Подобный тип доступа к данным менее удобен по сравнению с разделением файлов, но для настройки FTP-сервера от вас,

как от системного администратора, потребуется меньше времени и усилий. Межплатформенная совместимость также является существенным преимуществом FTP-сервера.

- **Анонимный доступ к данным.** Установив в сети анонимный FTP-сервер, вы дадите возможность внешним пользователям копировать на их компьютеры файлы, предоставленные вами для всеобщего доступа, а в некоторых случаях и передавать свои файлы на сервер. В качестве альтернативы анонимному FTP-серверу (особенно, если вам не требуется копирование данных с клиентских компьютеров на сервер) может выступать Web-сервер. Если же вы не хотите тратить время на подготовку Web-страниц, установка FTP-сервера может стать более приемлемым решением.

В обоих случаях необходимо учитывать вопросы безопасности. Если вы хотите предоставлять пользователям локальной сети их файлы, следует принять меры для того, чтобы обращения к серверу могли осуществляться только из локальной сети. Риск перехвата пароля в Internet настолько велик, что решение использовать FTP-сервер для передачи важных данных по глобальной сети было бы опрометчивым. Даже если FTP-сервер доступен лишь в пределах локальной сети, необходимо периодически менять пароли на случай, если кто-либо из пользователей поддастся соблазну и займется "подслушиванием" паролей. При использовании анонимного FTP-сервера сохранять пароль в секрете бессмысленно, но в этом случае вам необходимо принять меры для того, чтобы злоумышленник, обращающийся к серверу извне, не смог воспользоваться недостатками в его защите для получения несанкционированного доступа к данным в сети. Так, например, предоставление права записывать информацию на сервер связано с большим риском, в особенности если возможности записи информации не ограничены несколькими каталогами, которые полностью контролируются вами. Если вы разрешаете запись данных на анонимный сервер, вы должны принять меры для того, чтобы эти файлы становились доступными другим пользователям лишь после того, как вы ознакомитесь с ними и одобрите их. В противном случае ваш сервер может превратиться в инструмент для обмена данными между хакерами.

Программы, реализующие FTP-сервер в системе Linux

В настоящее время существует большое количество программ, реализующих FTP-сервер в системе Linux. Три из них, пользующиеся наибольшей популярностью, описаны ниже.

- **BSD FTPD.** Версия BSD Unix поставляется в комплекте с FTP-сервером, который адаптирован для переноса в систему Linux. Так, например, на выполнение в Linux ориентирован пакет OpenBSD FTPD. Варианты BSD FTPD поставляются с системами Debian и SuSE. От других FTP-серверов BSD FTPD отличается более надежными средствами защиты, но, тем не менее, он не получил большой популярности среди пользователей Linux.

- **ProFTPd.** Пакет ProFTPd, Web-сервер которого находится по адресу `http: //www. proftpd.` org, поставляется с системами Debian, Mandrake, Slackware, SuSE и Tur-

boLinux. Популярность данного сервера резко возросла в 2002 г. При создании ProFTPd были использованы некоторые подходы, характерные для сервера Apache.

- **WU-FTPD.** Washington University FTP Daemon (WU-FTPD) — наиболее популярный из современных FTP-серверов для Unix. Web-узел WU-FTPD расположен по адресу `http://www.wu-ftpd.org`. Данный продукт поставляется с Caldera, Debian, Mandrake, Red Hat, SuSE и TurboLinux. В процессе его эксплуатации были выявлены и устранены многочисленные недостатки в защите.

Каждая из этих программ поддерживает основные функции и некоторые из расширенных функций FTP-сервера. В данной главе описываются только продукты ProFTPd и WU-FTPD, так как именно они пользуются наибольшей популярностью и поставляются в составе различных дистрибутивных пакетов Linux. Программа ProFTPd обладает большей гибкостью и с точки зрения безопасности превосходит WU-FTPD. Однако если в составе вашей системы поставляется только WU-FTPD либо если вы хорошо знакомы с данным продуктом, имеет смысл остановить свой выбор на нем. Если же вы склонны экспериментировать, попробуйте установить в своей системе BSD FTPD.

Настройка основных функций FTP-сервера

Инсталлировав FTP-сервер, надо обеспечить его выполнение. Как правило, в дистрибутивных пакетах, комплектуемых WU-FTPD, запуск данной программы осуществляется посредством суперсервера, а если в составе системы содержится ProFTPd, для его запуска обычно используют сценарий SysV. Если вас не устраивает подход, использованный в вашей системе, вы можете реализовать альтернативное решение. При работе со многими дистрибутивными пакетами обеспечение запуска FTP-сервера является единственным действием, необходимым для его настройки, так как конфигурация, установленная по умолчанию, позволяет использовать сервер для решения многих задач. Так, например, по умолчанию пользователь, для которого в системе существует учетная запись, имеет возможность регистрироваться на FTP-сервере и копировать файлы из своего рабочего каталога. Если в вашей системе сервер должен действовать по-другому, следует изменить его настройку. Наиболее часто используемый вариант настройки — анонимный FTP-сервер — будет рассмотрен далее в этой главе.

Запуск FTP-сервера

Варианты запуска серверов в системе Linux рассматривались в главе 4. Если пакет, реализующий FTP-сервер, поставляется в составе системы, для обеспечения запуска сервера вам потребуется приложить лишь минимальные усилия. Не исключено также, что запуск сервера предусмотрен по умолчанию при установке пакета. Однако вам необходимо не упускать из виду следующие детали.

- В некоторых дистрибутивных пакетах, использующих `inetd`, в файл `/etc/inetd.conf` включается несколько записей для различных FTP-серверов. При желании вы можете инсталлировать различные FTP-серверы и запускать один из них по выбору. Для этого в файле `inetd.conf` надо удалить символ комментариев в начале записи, соответствующей выбранному серверу, закомментировать остальные записи и перезапустить `inetd`. Если в вашей системе установлен только один FTP-сервер,

необходимо убедиться, что в файле `inetd.conf` символ комментариев отсутствует лишь в записи, соответствующей этому серверу, в противном случае FTP-сервер работать не будет.

- В большинстве систем, использующих `xinetd` для запуска FTP-сервера, в каталог `/etc/xinetd.d` помещается специальный файл. Этот файл является частью пакета сервера. Если в нем содержится строка `disable = yes`, это означает, что запуск FTP-сервера запрещен. Подобная запись включается из соображений безопасности. Если вы хотите, чтобы FTP-сервер выполнялся в системе, задайте значение по опции `disable`. (Чтобы суперсервер учел внесенные изменения, его надо перезапустить.)

- Независимо от того, запускается ли FTP-сервер посредством `inetd` или `xinetd`, при запуске ему передаются некоторые параметры. По умолчанию в конфигурационном файле суперсервера указываются параметры для FTP-сервера, поставляемого вместе с системой. Если же вы хотите заменить FTP-сервер, вам надо изменить в конфигурационном файле суперсервера не только параметры, но и имя программы, реализующей FTP-сервер.

Если ваш FTP-сервер часто посещают пользователи, имеет смысл запускать его с помощью сценария SysV или локального сценария запуска. В этом случае сервер будет быстрее отвечать на запросы, но, учитывая небольшие размеры программы, реализующей сервер, увеличение быстродействия будет минимальным. В некоторых версиях Linux, например в Debian и Mandrake, данный подход применяется для запуска ProFTPd. Если же ProFTPd постоянно присутствует в памяти, поддержка анонимного FTP-сервера упрощается.

Перед тем как продолжить настройку, необходимо убедиться в том, что FTP-сервер работает и выполняет аутентификацию пользователей (при регистрации указываются пользовательское имя и пароль). При обращении FTP-клиента с удаленного узла сервер должен отобразить приглашение для ввода пользовательского имени и пароля. Рассмотрим следующий пример, в котором к серверу обращается клиентская программа `ftp`, выполняющаяся в системе Linux:

```
$ ftp harding.threeroomco.com
ftp: connect: Connection refused
```

Данное сообщение клиентской программы означает, что FTP-сервер не выполняется. Если вы получите подобный ответ, вам следует просмотреть файлы протоколов и выяснить причину возникновения проблемы. Если сервер был недавно установлен, причина, возможно, заключается в том, что вы забыли перезапустить суперсервер. Убедившись в том, что сервер работоспособен, можно продолжать его настройку.

Настройка WU-FTPD

При настройке WU-FTPD надо внести изменения в несколько конфигурационных файлов. Эти файлы определяют, кто из пользователей имеет право обращаться к FTP-серверу и какие действия доступны им. В некоторых файлах также содержатся дополнительные опции, посредством которых можно разрешить WU-FTPD выполнять специальные действия по обработке файлов или расширенные команды.

Конфигурационные файлы WU-FTPD

В большинстве случаев конфигурационные файлы, управляющие работой WU-FTPD, находятся в каталоге /etc. Имена этих файлов начинаются символами ftp.

- ftpaccess. Из всех конфигурационных файлов WU-FTPD наиболее сложным является файл ftpaccess. В нем содержатся опции, которые управляют регистрацией пользователей и правами доступа, определяют особенности TCP/IP-взаимодействия, используются для организации анонимного FTP-сервера, а также другие самые разнообразные средства.

- ftpconversions. Помимо прочих действий, файл ftpaccess управляет сжатием файлом или архивированием каталогов перед передачей их клиентам. Для того чтобы воспользоваться этой возможностью, необходимо определить типы файлов архивов.

- ftphosts. Данный файл позволяет ограничить набор узлов, с которых может осуществляться обращение к FTP-серверу, и даже запретить доступ для отдельных пользователей. Записи, начинающиеся с ключевого слова allow, разрешают, а записи, начинающиеся с deny, запрещают обращение для указанных узлов или указанных пользователей. Например, запись deny sjones означает, что попытки пользователя sjones зарегистрироваться на FTP-сервере будут блокированы, а запись deny badsite.pangaea.edu запрещает обращение к серверу для всех пользователей, работающих на узле badsite.pangaea.edu.

- ftpusers. Данный файл содержит список локальных пользователей, которым запрещено обращаться к серверу WU-FTPD. Этот файл не является частью WU-FTPD; он действует посредством модулей PAM (Pluggable Authentication Module). Тем не менее он представляет собой удобное средство противодействия попыткам незаконного использования FTP-сервера. По умолчанию в данный файл включаются имена, соответствующие учетным записям root, nobody и daemon.

- ftpservers. В обычных условиях один и тот же набор опций используется для работы с любыми клиентами. С помощью данного файла вы можете задать конфигурацию, которая будет применяться только при взаимодействии с определенными узлами. Каждая строка в этом файле содержит IP-адрес или имя узла, за которым следует имя каталога. Если клиент, обратившийся к FTP-серверу, выполняется на одном из указанных компьютеров, WU-FTPD использует для взаимодействия с ним конфигурационные файлы, находящиеся в соответствующем каталоге. Например, запись 192.168.21.8 /etc/ftpd/trusted означает, что при обращении клиента 192.168.21.8 будут использоваться конфигурационные файлы из каталога /etc/ftpd/trusted. Этот файл позволят контролировать обращения с внешних узлов и в то же время снимать ограничения для пользователей, работающих в локальной сети.

Из перечисленных выше файлов наиболее важным является ftpaccess. Файлы ftphosts, ftpusers и ftpservers также имеют большое значение для обеспечения безопасности сервера. Если вы хотите задать обработку файлов перед передачей их клиенту, вам, помимо конфигурационого файла ftpaccess, потребуется также файл ftpconversions.

Опции общего назначения для сервера WU-FTPD

Действие многих из опций, определяющих конфигурацию WU-FTPD, базируется на понятии *класса*. Класс — это группа пользователей, подобная группе Linux. Для определения класса в файле `ftpaccess` предусмотрена опция `class`, которая записывается в следующем формате:

```
class   имя_класса   список_типов   список_адресов
```

Назначение компонентов этой записи описано ниже.

- Имя класса. По умолчанию при инсталляции сервера часто создается универсальный класс с именем all. При необходимости вы можете определить свои классы.
- Список типов. В этом поле указывается список типов учетных записей, соответствующих данному классу. Ключевое слово `real` задает локальных пользователей, `guest` — пользователей, обращающихся с удаленных узлов, а `anonymous` описывает учетную запись для анонимного FTP-узла.
- Список адресов. Данный список содержит IP-адреса, имена узлов и имена доменов, принадлежащих классу. Символ ! означает, что указанный пункт должен отсутствовать в составе класса. Символ * определяет всех клиентов. Если список содержит несколько пунктов, над ними выполняется логическая операция OR. Например, значение `threeroomco.com,pangaea.edu` определяет всех клиентов, входящих в состав каждого домена.

Стандартный файл `ftpaccess` обычно содержит следующее определение:

```
class   all   real,guest,anonymous   *
```

Эта запись определяет универсальный класс, применимый ко всем типам доступа и включающий все узлы. Настраивая сервер, вы можете создать несколько классов, например, класс описывающий только локальных пользователей, и класс, соответствующий удаленным пользователям. Даже если классы различаются только списком адресов, вы можете использовать их независимо друг от друга.

В файл `ftpaccess` также включаются перечисленные ниже опции.

- deny *список_адресов файл_с_сообщением*. Эта опция указывает WU-FTPD на то, что все попытки доступа, предпринимаемые с указанных адресов, должны отвергаться. Данная опция похожа на запись deny в файле `ftphosts`, но в ней указывается файл, содержащий сообщение. Это сообщение, объясняющее причины отказа в соединении, передается пользователю, обратившемуся к серверу.
- autogroup *имя_группы класс*[*,класс*. . .]. Данная опция указывает, что при получении обращений одного из указанных классов WU-FTPD должен выполнить операцию `setgid` с именем группы. С помощью этой опции вы можете разрешить анонимным пользователям, принадлежащим некоторому классу, читать те файлы, которые имеют право читать члены указанной группы, но которые не доступны для чтения остальным пользователям.
- defumask *umask класс* [*, класс*]. Данная опция сообщает WU-FTPD, что если пользователь, принадлежащий указанному классу, передает файл на сервер, то при создании файла должна использоваться маска *umask*.

- timeout *ключевое_слово время_в_секундах*. Данная опция позволяет задает значения тайм-аута. В качестве ключевого слова может использоваться accept, connect, data, idle, maxidle или rfc931.

- noretrieve [relative|absolute] [class=*имя_класса*] *имена_файлов*. Эта опция запрещает передачу указанных файлов. Если вместо имени файла задано имя каталога, все содержимое каталога становится недоступным. При необходимости вы можете применять эту опцию только к указанному классу. Значения relative и absolute указывают, должно ли имя файла интерпретироваться как *относительное* (определяемое относительно корня *поддерева* chroot) или как *абсолютное* (определяемое относительно корневого каталога файловой системы). По умолчанию абсолютным считается имя, начинающееся с символа /. Например, если задана опция noretrieve /etc /usr, это означает, что копирование файлов из каталогов /etc и /usr запрещено.

СОВЕТ ✔ Опцию noretrieve часто используют для того, чтобы запретить доступ к /etc/passwd, /etc/shadow, /etc/ftpaccess, файлу core (находящемуся в любом каталоге) и другим важным файлам.

- allowretrieve [relative|absolute] [class=*имя_класса*] *имена_файлов*. Данная опция выполняет действия, противоположные опции noretrieve. С ее помощью определяются исключения из правила, заданного посредством noretrieve. Синтаксис данной опции полностью совпадает с синтаксисом noretrieve.

- message *имя_файла [событие]* [*класс*]. Опция message задает файл, содержимое которого должно быть передано FTP-клиенту при наступлении некоторого события. Так, например, если в качестве события указано ключевое слово login, сообщение будет отображаться при регистрации пользователя. Если событие описано как cwd=*каталог*, то сообщение будет передано при выборе этого каталога в качестве текущего. При необходимости вы можете ограничить действия данной опции определенным классом пользователей. Например, если задана опция message .message cwd=*, то при переходе в любой каталог пользователю будет передано сообщение из файла .message, содержащегося в этом каталоге. Таким образом, вы можете предоставлять пользователям описание содержимого каталогов и сообщать о назначении всего FTP-сервера.

- compress [yes|no] *класс [,класс]*. Данная опция разрешает сжатие данных. Если пользователь запрашивает один из существующих файлов, но указывает дополнительное расширение, означающее сжатие, этот файл будет передан в сжатом виде. (Например, для получения файла с именем file пользователь может указать имя file.gz.) Расширения, означающие сжатие, приведены в файле ftpconversions.

- tar [yes|no] *класс*[,*класс*]. Эта опция действует подобно опции compress, но применяется для объединения содержимого каталога в tar-архив. Данная опция предоставляет удобные средства для копирования каталогов.

- chmod, delete, overwrite, rename и umask. Эти опции принимают значение yes или no, кроме того, в них указывается такой же список типов, как и в определении класса. Каждая из этих опций разрешает или запрещает использование клиентом соответствующей команды. Например, запись delete no guest, anonymous запрещает пользователям типа guest и anonymous удалять файлы.

- dns refuse_mismatch *имя_файла*. Данная опция сообщает серверу WU-FTPD о том, что тот должен выполнить обратное преобразование IP-адреса клиента, а затем осуществить прямое DNS-преобразование. Если адрес, полученный в результате прямого преобразования, не соответствует адресу клиента, соединение должно быть разорвано. Однако перед разрывом соединения сервер передает клиенту содержимое указанного файла.

- dns refuse_no_reverse *имя_файла*. Данная опция указывает на то, что если выполнить обратное DNS-преобразование не удается, сервер не должен продолжать взаимодействие с клиентом. Перед завершением работы клиенту передается содержимое указанного файла.

Здесь приведены лишь некоторые из рпций WU-FTPD. Дополнительную информацию о настройке данного сервера можно получить, обратившись к страницам справочной системы, посвященным ftpaccess. Далее в этой главе будут также рассмотрены опции, применяемые для организации работы анонимного FTP-сервера.

Настройка ProFTPd

При создании конфигурационных файлов ProFTPd разработчики ориентировались на соответствующие средства сервера Apache, поэтому, если вам приходилось настраивать Apache, многие опции ProFTPd будут знакомы вам.

Конфигурационные файлы ProFTPd

Главный конфигурационный файл ProFTPd называется proftpd.conf; как правило, он располагается в каталоге /etc. В этом файле содержится большинство опций, используемых для настройки ProFTPd. Строки, содержащие комментарии, начинаются с символа #. Остальные записи представляются в следующем формате:

Директива [Значение]

Существуют директивы, для которых может быть задано несколько значений. Некоторые директивы формируют блок, включающий другие опции. Эти директивы помещаются в угловые скобки. Признаком окончания блока является директива, перед именем которой указан символ /. Пример блока, сформированного с помощью директивы Limit, приведен ниже.

```
<Limit WRITE>
 DenyAll
 Allow from 172.21.33.
</Limit>
```

Помимо главного конфигурационного файла, для настройки ProFTPd используется также файл ftpusers. Этот файл выполняет те же функции, что и одноименный файл сервера WU-FTPD. Пользователям, указанным в этом файле, запрещена регистрация на

FTP-сервере. (Строго говоря, ProFTPd применяет для аутентификации модули PAM, которые, в свою очередь, используют файл ftpusers.) По умолчанию при инсталляции ProFTPd создается файл ftpusers, в котором указываются такие имена пользователей, как nobody, daemon и root. Вы можете включить в данный файл учетные записи, созданные вами для специальных целей и не предполагающие регистрацию пользователей. Кроме того, в файле ftpusers можно задать имена обычных пользователей, которым по каким-либо причинам следует запретить доступ к FTP-серверу.

Опции общего назначения для сервера ProFTPd

В сервере ProFTPd предусмотрено большое количество директив, используемых для настройки этого сервера. Подробную информацию о них вы можете получить из документации на ProFTPd, представленной по адресу http://www.proftpd.org/docs/. Вероятнее всего, что, настраивая сервер, вы примете для большинства директив значения по умолчанию.

Поскольку значения многих директив применяются только в пределах *контекстного блока,* вам, прежде всего, надо составить представление о директивах, используемых для определения контекстных блоков. Эти директивы перечислены ниже.

- <Anonymous *имя_каталога*>. С помощью данной директивы вы можете создать анонимный FTP-сервер. В блоке, созданной посредством этой директивы, задаются другие директивы, используемые для обеспечения анонимного FTP-доступа. Анонимным пользователям разрешен доступ только к файлам, содержащимся в определенном каталоге, имя которого задается в качестве значения данной опции. Этот каталог ProFTPd указывает в качестве корневого каталога поддерева chroot (использование системной функции chroot() рассматривается в главе 23).

- <Directory *имя_каталога*>. С помощью данной директивы указывается каталог, к которому применяются другие директивы. Значением директивы является абсолютное имя каталога, начинающееся с символа /. Конфигурационный файл ProFTPd, создаваемый по умолчанию, обычно содержит блок, сформированный посредством директивы <Directory /*>. В этот блок помещаются директивы, с помощью которых задаются характеристики всех каталогов.

- <Global>. Директива <Global> формирует блок, содержимое которого применяется ко всему серверу и всем виртуальным узлам, формируемым посредством <VirtualHost>.

- <Limit *группа_команд*>. Данная опция задает набор команд FTP-клиента, действия которых ограничены директивами, содержащимися в составе блока. В группу команд входят одна или несколько команд из следующего набора: CWD, CDUP, MKD, RNFR, RNTO, DELE, RMD, RETR и STOR. В качестве значения данной директивы также могут быть указаны специальные идентификаторы, обозначающие категории команд. К ним относятся READ (все команды чтения), WRITE (все команды записи), DIRS (все команды для работы с каталогами) и ALL (все команды). Кроме того, для введения ограничений при регистрации вы можете использовать идентификатор LOGIN.

- <VirtualHost *адрес*>. ProFTPd позволяет поставить использование директив в зависимость от адреса клиента. В качестве значения данной директивы указыва-

ется IP-адрес или имя узла, и при обработке запроса с этого адреса применяются директивы, содержащиеся в блоке.

Большинство директив может присутствовать в одном или нескольких блоках. Кроме того, ряд директив указывается за пределами всех блоков; они рассматриваются как глобальные опции. Некоторые директивы могут встречаться как за пределами, так и в составе блока. Возможно также присутствие директивы в двух блоках, один из которого входит в состав другого. В этом случае директива с наибольшей степенью вложенности имеет наивысший приоритет. При настройке FTP-сервера чаще всего приходится изменять значения приведенных ниже директив.

- Allow [from] *идентификаторы_узлов*. Данная директива используется в составе блока <Limit> и указывает, какие клиенты имеют право доступа к ресурсу. В качестве идентификаторов узлов задается список IP-адресов, имен узлов, имен доменов (имя домена должно начинаться с точки) или блоков IP-адресов. Пункты списка отделяются друг от друга запятыми. Для идентификации узлов сети могут также использоваться ключевые слова all и none. После Allow может стоять ключевое слово from, но оно никак не изменяет действия, выполняемые данной директивой.

СОВЕТ ✔ Вместо имен узлов и доменов рекомендуется использовать блоки IP-адресов. Это уменьшает зависимость FTP-сервера от работы DNS-сервера.

- AllowAll. По умолчанию ProFTPd разрешает доступ к каталогам, но существуют различные способы ограничить доступ. Поместив директиву AllowAll в состав блока <Directory>, <Limit> или <Anonymous>, вы можете восстановить соглашения, принятые по умолчанию.

- AllowGroup *список_групп*. Данная опция разрешает доступ для пользователей, которым в блоке <Limit> запрещено обращаться к ресурсу. В качестве значения этой опции указываются имена групп, разделенные запятыми. Чтобы получить доступ к ресурсу, пользователь должен принадлежать всем указанным группам. Если имени группы предшествует символ !, учитываются лишь пользователи, не принадлежащие данной группе. Эта опция чаще всего используется для того, чтобы отменить для некоторых пользователей ограничения, наложенные такими опциями, как DenyAll.

- AllowOverwrite [on | off]. Эта опция определяет, может ли пользователь заменять файлы на сервере. По умолчанию принимается значение off, которое запрещает замену файлов.

- AllowUser *список_пользователей*. Директива AllowUser предоставляет пользователю или группе пользователей доступ к ресурсу, закрытому для остальных. Если имени пользователя предшествует символ !, право доступа получают все пользователи, кроме указанного.

- DefaultRoot *имя_каталога* [*список_групп*]. С помощью данной опции вы можете ограничить сферу деятельности пользователей подкаталогами определенного каталога. Для этого надо указать в качестве значения имя соответствующего

каталога. Имя каталога может начинаться с символа /, в этом случае предполагается абсолютное имя. Символ ~ определяет рабочий каталог пользователя. Если директива DefaultRoot должна иметь отношение лишь к некоторым из пользователей, надо указать их в списке групп. Список групп составляется по тем же правилам, что и для директивы AllowGroup.

СОВЕТ ✔ Указав глобальную опцию DefaultRoot ~, вы запретите пользователям доступ к системным каталогам, а также к рабочим каталогам других пользователей. Настроенный подобным образом, сервер ProFTPd разрешает пользователям обращаться только к своим рабочим каталогам.

- DefaultTransferMode [ascii|binary]. FTP-сервер поддерживает два режима передачи файлов. В двоичном режиме (binary) содержимое файла передается без изменений, а в символьном режиме (ascii) выполняется преобразование некоторых символов. Символьный режим удобен для передачи текстовых файлов, но недопустим для работы с двоичными файлами, например, файлами, содержащими код программ. Директива DefaultTransferMode устанавливает режим передачи, используемый по умолчанию. При инсталляции сервера задается значение ascii данной директивы.
- Deny [from] *идентификаторы_узлов*. Данная директива выполняет действия, противоположные директиве Allow. Она используется в составе блока <Limit> и запрещает клиентам доступ к ресурсу.
- DenyAll. Данную директиву можно использовать в составе блоков <Limit>, <Anonymous> или <Directory>. Она запрещает всем пользователям доступ к ресурсу. При необходимости вы можете указать в том же блоке одну из директив, отменяющих для некоторых пользователей ограничения, наложенные посредством DenyAll.
- DenyGroup *список_групп*. Эта директива позволяет определять группы, которым запрещен доступ к ресурсу в блоке <Limit>. Список групп формируется так же, как и для директивы AllowGroup.
- DenyUser *список_пользователей*. Данная директива противоположна AllowUser. Она запрещает доступ к ресурсу в блоке <Limit>.
- DisplayConnect *имя_файла*. Если в конфигурационном файле указана данная опция, ProFTPd передает клиенту текст, содержащийся в указанном файле. Это происходит после установления соединения, но до завершения процедуры регистрации.
- DisplayFirstChdir *имя_файла*. Данная директива указывает ProFTPd на то, что содержимое заданного файла должно быть передано пользователю в тот момент, когда он впервые сделает каталог текущим. Чаще всего для данной директивы задается файл .message, в результате чего пользователю передается файл .message, содержащийся в том каталоге, к которому он обращается в первый раз.
- DisplayLogin *имя_файла*. Эта директива действует подобно директиве DisplayConnect, но сообщение передается пользователю после успешного завершения процедуры регистрации.

- Group *идентификатор_группы*. Сервер ProFTPd запускается от имени пользователя root, но сразу после запуска переходит на выполнение с ограниченными полномочиями. Это снижает риск незаконного проникновения в систему извне. С помощью данной директивы указывается группа, полномочия которой получает ProFTPd. При инсталляции сервера в качестве значения данной директивы обычно задается nogroup, ftp или другая подобная группа.

- MaxClients *число* \ попе. Данная директива позволяет ограничить количество клиентов, которые могут работать с сервером. Числовое значение (например, 30) определяет максимальное количество клиентов, значение попе снимает данное ограничение.

- MaxInstances *число*. Эта директива действует подобно MaxClients, но если MaxClients задает максимальное количество успешно зарегистрировавшихся пользователей, то Maxlnstances ограничивает число соединений, устанавливаемых сервером. Директива Maxlnstances неэффективна при запуске ProFTPd посредством суперсервера, но подобные ограничения можно задать при настройке самого суперсервера.

- Order allow, deny | deny, allow. Если в блоке <Limit> присутствуют и запрещающие, и разрешающие директивы, ProFTPd сначала выполняет проверку на соответствие разрешающим, а лишь затем запрещающим директивам. В результате разрешающие директивы имеют более высокий приоритет, чем запрещающие. Кроме того, если отсутствует запрещающая директива, доступ по умолчанию разрешается. Поведение сервера можно изменить с помощью опции Order deny, allow. В этом случае запрещающие директивы получают более высокий приоритет, чем разрешающие, а доступ, не разрешенный явно, не предоставляется.

- RootLogin on | off. По умолчанию ProFTPd отказывает в регистрации пользователю root. Если в конфигурационном файле присутствует опция RootLogin on, root получает право работать на FTP-сервере. (Для этого вам, возможно, придется предпринять и другие действия, например, удалить пользователя root из файла /etc/ftpusers.)

- ServerIdent on | off ["*строка-идентификатор*"]. Данная директива определяет, должен ли ProFTPd при установлении соединения предоставлять клиенту сведения о себе. Задавая значение on данной директивы, вы можете также указать строку-идентификатор. По умолчанию сервер сообщает пользователю, что для реализации функций FTP-сервера используется продукт ProFTPd. Если вы не собираетесь объявлять тип программы, вам надо явно указать строку, которая должна передаваться клиенту.

- ServerName "строка-идентификатор". С помощью данной директивы вы можете задать имя сервера, которое будет включаться в состав строки, используемой Serverldent. Директива Serverldent позволяет переопределить все сообщение, но если вы хотите изменить лишь имя сервера, вы можете использовать для этого опцию ServerName.

- ServerType inetd | standalone. Если вы запускаете ProFTPd посредством суперсервера, вы должны задать значение inetd данной опции, если же для запуска

используется сценарий SysV или локальный сценарий, надо установить значение standalone. С помощью данной опции ProFTPd получает сведения о том, запущен ли он от имени обычного пользователя и должен ли непосредственно обрабатывать запрос (inetd), или запуск осуществляется от имени пользователя root и для обработки запросов следует порождать новые процессы (standalone).

- SyslogLevel emerg|alert|crit|error|warn|notice|info|debug. Данная директива определяет, насколько подробные сведения должны записываться в файл протокола. Значения расположены по мере возрастания объема записываемой информации: emerg соответствует самым общим, а debug — наиболее подробным сведениям.
- TransferLog *имя_файла* \NONE. С помощью данной директивы вы можете указать файл протокола для помещения в него сведений о переданных файлах или запретить запись подобной информации (для этого надо задать значение NONE). Данная директива позволяет создавать различные файлы протоколов, предназначенные для разных целей. Она может независимо использоваться в блоках <Anonymous>, <VirtualHost> и <Global>, а также указываться за пределами всех блоков.
- Umask *маска_файла* [*маска_каталога*]. Данная директива позволяет задать маску umask, которая будет использоваться при создании новых файлов (и, возможно, новых каталогов). По умолчанию принимается значение 022, приемлемое для многих систем.
- UseFtpUsers on|off. Задавая значение off директивы UseFtpUsers, вы можете запретить использование файла /etc/ftpusers. По умолчанию для данной директивы устанавливается значение on.
- UserAlias *псевдоним имя_пользователя*. В обычных условиях ProFTPd использует для аутентификации пользовательское имя, указанное при регистрации. Посредством данной директивы вы можете задать псевдоним, который будет отображаться в имя конкретного пользователя. Например, если в конфигурационном файле указана опция UserAlias rjones ronald, то при указании имени rjones аутентификация будет производиться с помощью учетной записи ronald. (Такая конфигурация часто используется в анонимных FTP-серверах, где для аутентификации всех пользователей применяется учетная запись ftp.)

Директивы, приведенные выше, а также другие директивы, не рассмотренные в данной главе, позволяют настроить ProFTP для выполнения самых разнообразных задач. Стандартная конфигурация позволят пользователям регистрироваться на сервере и работать с файлами, расположенными в их рабочих каталогах. В конфигурационном файле можно создать специальный блок <Anonymous> для поддержки анонимных обращений. Для того чтобы обеспечить доступ всех пользователей к серверу, надо внести в его конфигурацию дополнительные изменения. Кроме того, файлы, предназначенные для всеобщего доступа, необходимо разместить в специально предназначенных для этого каталогах.

Установка анонимного FTP-сервера

FTP-серверы очень часто используются для предоставления анонимного FTP-доступа. Как было сказано ранее в этой главе, вместо анонимного FTP-сервера может быть

установлен Web-сервер, однако в ряде ситуаций FTP-сервер целесообразно использовать параллельно с Web-сервером и даже вместо него. Так, например, вам может потребоваться анонимный FTP-сервер для предоставления файлов всем желающим и обычный FTP-сервер, осуществляющий аутентификацию посредством анализа пользовательского имени и пароля. Установив серверы, поддерживающие протоколы HTTP и FTP, вы упростите работу пользователей, привыкших работать лишь с одной из служб.

Перед тем как устанавливать анонимный FTP-сервер, вам надо тщательно изучить вопросы обеспечения безопасности системы. Если вы решили настроить FTP-сервер для предоставления файлов всем желающим, вам надо изучить назначение соответствующих опций, находящихся в конфигурационных файлах. Кроме настройки сервера, вам также придется изменить конфигурацию системы, в частности, установить необходимые права доступа к каталогам.

Особенности работы анонимного FTP-сервера

Основное назначение анонимного FTP-сервера — обеспечивать передачу файлов с сервера на клиентский компьютер. На анонимном FTP-сервере можно разместить программы, документацию и другие сведения, которые вы собираетесь предоставить всем желающим. В состав HTML-документов можно включать ссылки на файлы, находящиеся на анонимном FTP-сервере; такие ссылки должны начинаться последовательностью символов `ftp://` (например, `ftp://ftp.threeroomco.com/pub/manual.pdf`). Некоторые особенности работы подобных FTP-серверов требуют специального рассмотрения.

- Файлы, расположенные на анонимном FTP-сервере, копируются в одном направлении: с сервера на клиентский компьютер. Подобным образом настроено большинство Web-серверов. Существуют исключения из данного правила, но файлы, скопированные на анонимный FTP-сервер, обычно становятся недоступными остальным пользователям. К подобным мерам администраторы прибегают для того, чтобы их компьютеры не стали пунктом для обмена незаконной информацией. Если вы хотите получать файлы от удаленных пользователей, лучше всего настроить сервер так, чтобы перед передачей данных пользователь должен был зарегистрироваться на сервере. В качестве альтернативы FTP-серверу можно рассмотреть обмен документами по электронной почте.

- Файлы, расположенные на анонимном FTP-сервере, доступны всем желающим. Это означает, что на том компьютере, на котором размещен анонимный FTP-сервер, нельзя размещать важные данные. Для того чтобы организовать защиту системных файлов, пользователю, обратившемуся на анонимный FTP-сервер, предоставляется лишь ограниченное подмножество файловой системы. Вся информация, находящаяся за пределами выделенной области, скрыта от него. На большинстве FTP-серверов для обеспечения защиты создается поддерево `chroot`, которое будет рассматриваться в главе 23.

ВНИМАНИЕ | Несмотря на то что поддерево `chroot` повышает уровень защиты системы, оно не гарантирует ее безопасность. Лучше всего удалить всю секретную информацию с компьютера, на котором работает анонимный FTP-сервер. В результате, даже если взломщику удастся проникнуть в систему, он не сможет похитить важные данные.

Если FTP-сервер использует поддерево chroot, вам придется скопировать в соответствующие каталоги некоторые системные конфигурационные файлы. Во многих пакетах, реализующих FTP-серверы для системы Linux, такие копии файлов создаются по умолчанию. Некоторые серверы, в том числе ProFTPd, перед тем как ограничить сферу своего доступа поддеревом chroot, имеют возможность прочитать свои конфигурационные файлы, поэтому объем данных, которые необходимо скопировать в каталоги поддерева, остается минимальным.

Некоторые варианты конфигурации FTP-сервера (в особенности это касается ProFTPd) лучше работают при запуске сервера посредством сценария SysV. В других случаях нормальная работа обеспечивается и при использовании суперсервера (это справедливо для WU-FTPD). Одна из особенностей настройки FTP-сервера состоит в том, что системный вызов chroot() может использоваться только в том случае, если программа запускается от имени пользователя root. Если в конфигурационном файле суперсервера указан запуск FTP-сервера от имени другого пользователя, создать поддерево chroot невозможно. (Как было сказано ранее, вскоре после запуска FTP-сервер переходит к работе с ограниченными полномочиями, но происходит это после вызова системной функции chroot().)

Для работы анонимного FTP-сервера необходимо, чтобы некоторые файлы располагались в определенных каталогах. Этот вопрос будет подробнее рассмотрен далее в этой главе.

Обеспечение безопасности при работе анонимного FTP-сервера

Поскольку анонимный FTP-сервер доступен всем желающим, он может создавать серьезную угрозу для системы. Считается, что поддерживать на компьютере анонимный FTP-сервер не более опасно, чем Web-сервер или почтовый сервер. Однако, как показывает опыт многих администраторов, риск оказывается намного больше. Одна из причин состоит в том, что в защите многих программ, реализующих FTP-серверы (и особенно в WU-FTPD), были обнаружены многочисленные недостатки. Еще один источник опасности связан с тем, что FTP-сервер обеспечивает двунаправленную передачу данных, поэтому, если взломщику удастся воспользоваться недостатками защиты и выйти за пределы поддерева chroot, он сможет изменить конфигурацию системы в соответствии со своими потребностями. По сравнению с FTP-сервером почтовый сервер дает злоумышленнику гораздо меньше возможностей для воздействия на систему. Это связано с особенностями обработки почты.

Одно из положительных качеств анонимного FTP-сервера состоит в том, что он не позволяет перехватить пароль. Связано это с тем, что при обращении на анонимный FTP-сервер пользователь может указать любой пароль на свой выбор. (Большинство серверов предлагает пользователю ввести в качестве пароля свой почтовый адрес.) С этой точки зрения анонимный FTP-сервер является более защищенным, чем обычный FTP-сервер.

Если вы сконфигурируете одну программу для работы в качестве как анонимного, так и обычного FTP-сервера, вы тем самым объедините недостатки обоих типов серверов. Лучше всего, если анонимный FTP-сервер будет поддерживать только анонимные обращения (по крайней мере от внешних пользователей). Чтобы повысить уровень защиты, минимизируйте число учетных записей на компьютере, на котором работает анонимный сервер, и запретите запуск ненужных серверов.

Опции, используемые для настройки анонимного FTP-сервера

Большинство FTP-пакетов, поставляемых в составе различных версий Linux, полностью или частично сконфигурированы для функционирования в качестве анонимного FTP-сервера. Для того чтобы окончить настройку, вам надо приложить лишь минимальные усилия. В данном разделе описываются опции, позволяющие реализовать анонимный FTP-сервер как с помощью WU-FTPD, так и посредством ProFTPd. Вначале будет рассмотрено создание дерева каталогов (эта задача решается одинаково для обоих серверов), а затем — средства конфигурации для каждого из серверов.

Создание поддерева каталогов

Первый шаг по организации работы FTP-сервера — это создание поддерева каталогов. Как правило, корнем этого поддерева является каталог `/home/ftp`, но при необходимости вы можете разместить его в другой позиции файловой системы. В большинстве случаев в качестве владельца каталогов указывается `root`, либо пользователь, который должен непосредственно заниматься поддержкой FTP-узла, а права доступа к каталогам задаются равными 755 (`rwxr-xr-x`). Это позволяет администратору редактировать файлы, содержащиеся в каталоге, но другим пользователям запись данных запрещена. В отличие от каталогов, при определении прав доступа к файлам не устанавливается бит исполняемой программы.

В большинстве случаев в составе поддерева FTP задаются следующие подкаталоги.

- pub. В этом каталоге размещаются файлы, предназначенные для копирования на пользовательские компьютеры. Вы можете создавать в каталоге pub любую структуру подкаталогов и помещать в них любые файлы. Необходимо лишь обеспечить, чтобы пользователь `ftp` имел право чтения этих файлов.

- bin. Для выполнения некоторых действий FTP-сервер обращается к другим программам. Эти программы должны находиться в каталоге /bin (путь к каталогу определяется относительно корневого каталога, заданного с помощью функции `chroot()`). Чаще всего серверу требуется утилита `ls`, кроме того, в процессе работы ему могут понадобиться программы `tar`, `gzip` и `zcat` (последний файл представляет собой символьную ссылку на `gzip`). Установив FTP-сервер, вы, возможно, обнаружите, что в каталоге bin уже находятся некоторые программы, причем размер их превышает размер соответствующих программ, находящихся в каталоге /bin системы. Причина в том, что при установке FTP-сервера в состав исполняемого файла помещаются все необходимые коды, в результате чего исключается необходимость в библиотечных файлах. Убедитесь, что для файлов, находящихся в этом каталоге, установлен бит исполняемой программы.

- `lib`. В этом каталоге содержатся динамические библиотеки, используемые при работе программ в /bin. Если вы скопируете в каталог /bin поддерева FTP утилиты из каталога /bin операционной системы, вам надо выяснить, какие библиотеки требуются для работы каждой из них. Это позволяет сделать команда `ldd`. Так, например, чтобы определить, какие библиотеки нужны для программы `ls`, надо выполнить команду `ldd /bin/ls`.

- etc. Для работы FTP-сервера требуются два файла, содержащихся в каталоге /etc: passwd и group. Вам нет необходимости копировать соответствующие файлы из каталога /etc системы. Вам нужна лишь запись для пользователя ftp (либо другого пользователя, учетную запись которого вы используете для организации анонимного доступа).

После того как вы создадите перечисленные выше каталоги и поместите в них требуемые файлы, можете приступать к дальнейшей настройке анонимного FTP-сервера. Возможно, впоследствии вы внесете некоторые изменения в структуру поддерева FTP, например, включите новые файлы или модифицируете существующие. Например, не исключено, что вы захотите, чтобы перед передачей производилось сжатие файла посредством утилиты gzip. В этом случае вам придется скопировать соответствующий исполняемый файл в каталог /bin поддерева FTP.

Опции WU-FTPD, используемые при создании анонимного FTP-сервера

Наиболее важные опции, используемые для создания анонимного FTP-сервера на базе продукта WU-FTPD, находятся в файле /etc/ftpaccess. В процессе настройки вам может потребоваться изменить значения следующих опций.

- class. Возможно, для обеспечения анонимного доступа вам придется создать новый класс. Для его создания применяются те же средства, что и для формирования других классов.

- compress, tar, chmod, delete, overwrite и rename. Эти опции разрешают или запрещают пользователю выполнять соответствующие команды. Запретив анонимному пользователю вызывать команды, определяемые последними четырьмя опциями, вы лишите его возможности изменять файлы, расположенные на сервере. Эти опции могут показаться излишними, однако некоторая избыточность позволяет сохранить контроль над сервером, если при установке конфигурации была допущена ошибка или если некоторые функции сервера выполняются некорректно.

- anonymous-root. В качестве значения данной опции надо задать корневой каталог поддерева chroot, в пределах которого выполняется сервер WU-FTPD.

В большинстве систем WU-FTPD запускается посредством суперсервера с привилегиями root. Когда сервер принимает запрос от анонимного пользователя, он порождает процесс от имени пользователя ftp. Таким образом, WU-FTPD может реализовать анонимный FTP-сервер, даже при запуске посредством суперсервера.

Опции ProFTPd, используемые при создании анонимного FTP-сервера

Основные опции ProFTPd, используемые для настройки анонимного FTP-сервера, находятся в файле proftpd.conf. Фрагмент конфигурационного файла, реализующий простой анонимный сервер, приведен ниже.

```
<Anonymous /home/ftp>
  User                          ftp
  Group                         ftp
```

```
  # При регистрации пользователь может указывать имя anonymous
  # либо ftp
  UserAlias                    anonymous ftp
  # В пределах поддерева chroot запись данных запрещена
  <Limit WRITE>
    DenyAll
  </Limit>
</Anonymous>
```

- Директива **<Anonymous>** создает **блок,** в **который** помещаются остальные опции, применяемые для формирования конфигурации анонимного сервера. При наличии этой директивы **ProFTPd** изменяет процедуру регистрации, в данном примере сервер создает поддерево chroot, корневой каталог которого размещается в каталоге **/home** /ftp.

- Директивы User и Group сообщают ProFTPd о том, какое имя пользователя и группы должно использоваться для работы с анонимным сервером. ProFTPd порождает процесс с полномочиями указанного пользователя и группы. Необходимо убедиться в том, что каталоги поддерева FTP и расположенные в них файлы доступны для указанных пользователя и группы.

- Директива UserAlias обеспечивает обслуживание пользователей, которые указывают при регистрации имя anonymous.

- В блоке, созданном с помощью директивы **<Limit WRITE>,** содержится директива **DenyAll.** Этот блок запрещает всем пользователям запись в каталоги. Если вы корректно задали права доступа в поддереве FTP, этот блок можно считать излишним. Однако, как было сказано ранее, некоторая избыточность поможет в том случае, если при настройке сервера была допущена ошибка или если в его системе защиты есть недостатки.

- Если вы хотите создать *псевдоанонимный сервер,* который регистрирует пользователей посредством имени anonymous, но требует ввода пароля, вы должны **использовать** опцию **AnonRequiresPas**sword on. В этом случае вам также следует задать пароль в файле /etc/passwd или /etc/shadow. (Сервер ProFTPd выполняет аутентификацию пользователя перед тем, как ограничить сферу своих действий поддеревом chroot, поэтому соответствующий пароль задается в системном файле /etc/passwd или /etc/shadow.)

Если вы хотите, чтобы **FTP-сервер** работал только как анонимный сервер, вам надо принять меры для того, чтобы запретить доступ обычным пользователям. По возможности разместите FTP-сервер на том компьютере, на котором имеется как можно меньше учетных записей администраторов, и запретите доступ этим пользователям, включив их имена в файл **/etc/ftpusers.**

Резюме

Ранее FTP-серверы представляли собой чрезвычайно важный компонент Internet, но в настоящее время некоторые их функции взяли на себя Web-серверы. Однако и сейчас FTP-серверы широко используются как средства для обеспечения удаленного доступа пользователей к своим файлам (в этом случае поддерживается двунаправленный обмен информацией) и поддержки анонимных обращений (при этом чаще всего осуществляется передача данных с сервера на клиентские машины). Наибольшей популярностью в Linux пользуются FTP-серверы WU-FTPD и ProFTPd. Обе программы предоставляют широкий набор возможностей и обслуживают как пользователей, регистрирующихся на сервере, так и анонимных пользователей. Настраиваются эти программы по-разному. Структура конфигурационных файлов ProFTPd напоминает структуру файлов Apache. При установке большинство серверов настраивается для регистрации пользователей. Незначительно изменив содержимое конфигурационных файлов, вы можете настроить их для работы в качестве анонимных FTP-серверов.

ЧАСТЬ IV

Средства защиты и маршрутизации

Глава 22

Общие вопросы защиты системы

Linux — мощная система, способная поддерживать различные типы сетевого взаимодействия. Однако обилие средств обмена данными по сети неминуемо создает угрозу безопасности системы. В защите многих серверов, созданных для работы в системе Linux, неоднократно выявлялись недостатки, которые могли быть использованы для получения незаконного доступа к системе. Более того, сам принцип работы некоторых серверов не идеален с точки зрения защиты. Так, например, некоторые из них обмениваются данными с клиентом в незакодироваином виде, что может привести к перехвату пароля или секретной информации, передаваемой по сети. Выполняя администрирование системы, нельзя недооценивать вопросы ее защиты. Вам необходимо принимать все меры для повышения уровня безопасности компьютеров и следить за последними сообщениями, публикуемыми на Web-серверах, в списках рассылки и в группах новостей. Если окажется, что в сервере, находящемся на вашем компьютере, была обнаружена ошибка, следует позаботиться о ее устранении, в противном случае вы рискуете стать жертвой взломщика.

В данной главе рассматриваются способы, позволяющие выявить ненужные серверы и запретить их выполнение; средства контроля за использованием учетных записей и паролей; вопросы, связанные с обновлением программ, выполняющихся на вашем компьютере; методы распознавания попыток взлома, а также дополнительная информация о защите системы. Специальные средства, применяемые для обеспечения безопасности, будут более подробно рассмотрены в последующих главах. В частности, в главе 23 описывается способ ограничения сферы действия сервера поддеревом файловой системы; глава 25 посвящена вопросам настройки средств фильтрации пакетов, используемых для создания брандмауэров; в главе 26 рассматриваются средства расширения локальной сети на другие области Internet и обеспечение кодирования передаваемой информации.

Тем, кто собирается тщательно изучить вопросы безопасности системы, можно посоветовать дополнительную литературу, в частности, работы Манна (Mann) и Митчела (Mitchell) *Linux System Security: The Administrator's Guide to Open Source Security Tools* (Prentice Hall, 1999), а также Гарфинкела (Garfinkel) и Спеффорда (Spafford) *Practical UNIX & Internet Security, 2nd Edition* (O'Reilly, 1996). Из изданий, специально посвященных созданию брандмауэров, стоит обратить внимание на книгу Констейнтайна (Constain-

tine) и Зиглера (Ziegler) *Linux Firewalls* (New Riders, 2001). Если же в вашей сети имеются компьютеры, выполняющиеся под управлением систем, отличных от Linux, то, возможно, вам будет полезна книга Макклуе (McClure), Скембри (Scambray) и Курца (Kurtz) *Hacking Exposed, 3rd Edition* (McGraw-Hill, 2001).

Отключение ненужных серверов

Серверы обеспечивают доступ к ресурсам компьютеров, поэтому каждая серверная программа, выполняющаяся на компьютере, увеличивает опасность незаконного проникновения в систему. Взломщик может воспользоваться недостатками в защите сервера, ошибками в его настройке или перехватить пароль, передаваемый по сети. Чтобы снизить риск незаконного обращения к важным данным, следует запретить выполнение ненужных серверов, но для этого надо обнаружить их. Выявив ненужный сервер, следует найти способ отключить его. Обычно отключение серверов не вызывает проблем, но существуют способы, позволяющие решить эту задачу наиболее эффективно.

Выявление ненужных серверов

Задачу выявления ненужных серверов можно разбить на две подзадачи: идентификация серверов, присутствующих в системе, и принятие решения о том, какие из них могут быть отключены без вреда для системы. Обе эти подзадачи можно решить различными способами. Применяя для обнаружения серверов разные подходы, вы увеличиваете свои шансы на успех.

Обнаружение серверов, присутствующих в системе

В системе Linux отсутствует централизованный реестр выполняемых серверов, поэтому для обнаружения серверов необходимо объединить информацию из различных источников. Используя лишь один способ обнаружения, можно упустить из виду тот или иной сервер, поэтому лучше использовать для их выявления различные подходы.

Использование средств управления пакетами

Для обнаружения серверов, присутствующих в системе, может быть использована система управления пакетами. Если при установке программ вы пользовались исключительно диспетчером пакетов, то в базе данных диспетчера содержится информация о каждой программе, которая была инсталлирована на компьютере. Просмотрев описание пакета, содержащееся в базе, можно определить, является ли установленная программа сервером и насколько она необходима для работы системы. В качестве примеров инструментальных средств управления пакетами можно привести инструмент GNOME RPM, который используется в системе Red Hat, YaST — в SuSE, и Storm Package Manager (часть дистрибутивного пакета Storm, используемая в системе Debian). Окно GNOME RPM, предназначенное для просмотра инсталлированных пакетов, показано на рис. 22.1. Выбрав пакет, вы сможете прочитать его описание. Некоторые диспетчеры пакетов распределяют информацию, которая хранится в базе, по категориям, однако на практике, чтобы найти серверы, присутствующие на компьютере, надо просмотреть все категории. Диспетчеры пакетов не позволяют выявить серверы, для инсталляции которых были использованы tar-архивы или исходные коды. Кроме того, такой подход не дает возможности ответить на вопрос, выполняется ли сервер в системе. (Сервер, который был инсталлирован, но не запущен, создает гораздо меньшую угрозу безопасности системы, чем сервер, выполняющийся на

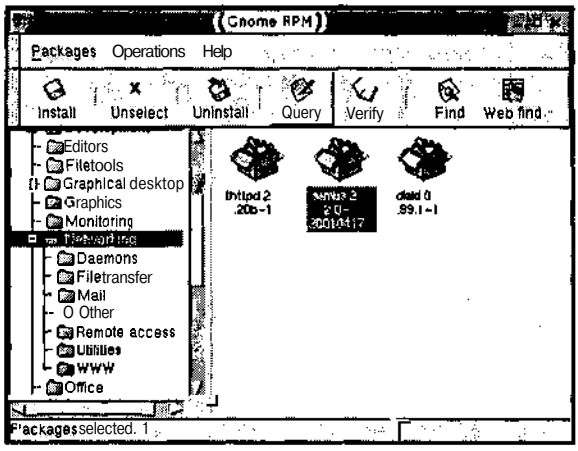


Рис. 22.1. Диспетчер пакетов предоставляет информацию о серверах, установленных в системе

компьютере. Реальная опасность возникнет лишь в том случае, если впоследствии будет установлена конфигурация системы, предполагающая запуск сервера.)

Проверка файлов запуска

Чтобы обнаружить серверы, присутствующие в системе, можно проверить следующие файлы.

- **Конфигурационный файл суперсервера.** При поиске серверов следует проверить конфигурационные файлы `/etc/inetd.conf` и `/etc/xinetd.conf`, а также файлы в каталоге `/etc/xinetd.d`. Таким образом, вы найдете ссылки на все серверы, запускаемые посредством суперсервера. В конфигурационном файле `inetd` строки, в начале которых стоит символ #, представляют собой комментарии, поэтому серверы, указанные в них, не активны. Для суперсервера `xinetd` запуск сервера запрещает запись `disable = yes`.

- **Сценарии запуска** SysV. Можно найти серверы, запускаемые посредством сценариев SysV, просмотрев каталоги, предназначенные для размещения таких сценариев (обычно это каталог `/etc/rc.d/rc?.d` или `/etc/rc?.d`, где символ ? означает уровень выполнения). Необходимую вам информацию предоставят имена файлов, находящихся в этих каталогах. Заметьте, что некоторые из программ, запускаемых посредством сценариев SysV, не являются серверами, поэтому, прежде чем запретить их выполнение, следует выяснить назначение этих программ.

- **Локальные сценарии запуска.** Во многих дистрибутивных пакетах для запуска локальных программ используются локальные сценарии. Файлы, содержащие их, обычно называются `rc.local` или `boot.local`. Локальными считаются такие программы, при инсталляции которых использовался нестандартный способ, отличающийся от подхода, принятого для данного дистрибутивного пакета. Чтобы обнаружить серверы, выполняющиеся в системе, надо просмотреть весь локальный сценарий.

Подробно вопросы запуска серверов, в том числе соглашения об именовании сценариев SysV, были рассмотрены в главе 4. Определить назначение сценариев запуска вам помогут такие инструменты, как ntsysv и tksysv. Кроме того, в некоторых системах (Caldera, Mandrake, Red Hat и TurboLinux) команда chkconfig --list, заданная в командной строке, отображает состояние сценариев SysV, а в ряде случаев и назначение записей в конфигурационном файле xinetd.

Проанализировав сценарии запуска и конфигурационный файл суперсервера, вы выясните, какие серверы выполняются в системе, однако ничего не узнаете об инсталлированных серверах. Как было сказано ранее, получить сведения о программах, которые были инсталлированы на компьютере, позволяет диспетчер пакетов. Кроме того, вы можете проверить все исполняемые файлы в системе, однако для этого необходимо затратить столько усилий, что данное решение нельзя считать приемлемым.

Анализ данных о процессах

Для обнаружения серверов может использоваться утилита ps. Она возвращает информацию о. процессах, выполняющихся в системе. При запуске ps можно указывать разные опции, но для выявления серверов достаточно ввести в командной строке ps ax. Объем информации, возвращаемой ps, достаточно велик, поэтому имеет смысл перенаправить вывод в файл или передать данные, сгенерированные программой ps, утилите more или less. Если вы ищете сведения о конкретном сервере, используйте утилиту grep. Например, чтобы получить информацию о сервере sendmail, надо ввести команду ps ax | grep sendmail. Следует помнить, что утилита ps предоставляет сведения как о серверах, так и о других программах. Ниже приведен фрагмент данных, сгенерированных программой ps.

```
$ ps ax
  PID TTY      STAT   TIME COMMAND
    1 ?        S      0:15 init [3]
  502 ?        S      0:05 named -u bind
  520 ?        S      0:01 cupsd
  535 ?        SW     0:00 [nfsd]
 1741 pts/4    S      0:00/bin/bash
 4168 ?        S      0:00 httpd
```

На самом деле в процессе выполнения программа ps генерирует десятки и даже сотни строк. В данном примере удалена почти вся информация, кроме нескольких строк, иллюстрирующих работу этой утилиты. В первой строке программа ps выводит сведения о процессе init, для идентификации которого всегда используется номер 1. Данный процесс является корнем дерева, представляющего иерархию процессов в системе. Все остальные процессы порождаются либо непосредственно init, либо его дочерними процессами. Процессы, имена которых помещаются в квадратные скобки, представляют собой процессы ядра. В данном примере процессом ядра является [nfsd]. Как видно из его имени, [nfsd] поддерживает функции сервера NFS, реализованного средствами ядра. Процессы named, cupsd и httpd представляют собой пользовательские процессы. О принадлежности их к серверам можно судить по двум признакам. Во-первых, имя каждого из них оканчивается буквой "d", а во-вторых, эти процессы не связаны с терминалами (в поле TTY отображается символ ?). В отличие от них, процесс /bin/bash не является процессом сервера, так как в поле TTY выводится значение pts/4, т. е. данный процесс связан с конкретным терминалом.

Определив процессы серверов с помощью ps, надо отыскать документацию на те серверы, назначение которых вам неизвестно. Для этого следует ввести команду man *имя,* указав в качестве параметра имя интересующего вас процесса. Кроме того, постарайтесь найти исполняемый файл с именем, совпадающим с именем процесса. Это позволит вас проследить, какой пакет использовался для его инсталляции. Для получения информации о пакете введите команду rpm -qf *путь_к_файлу.* (В системе Debian для этого используется команда dpkg -S *путь_к_файлу.*)

Используя ps, не забывайте, что эта утилита не отображает сведения о серверах, которые не выполнялись в момент ее запуска. Например, если сервер запускается с помощью суперсервера и во время вызова ps ни один из клиентов не работал с ним, вы не получите информацию об этом сервере. Данная утилита также не предоставит сведений о тех серверах, работа которых была временно завершена.

Использование netstat

При использовании ps для поиска серверов возникает проблема, состоящая в том, что данная утилита не сообщает, используется ли данный сервер для поддержки сетевого взаимодействия. Получить эту информацию вам поможет программа netstat. Данная программа возвращает данные о сетевых соединениях. Подобно ps, при запуске netstat могут указываться различные опции. Для поиска информации о серверах можно использовать команду netstat -lp. Опция -l сообщает утилите netstat о том, что она должна анализировать порты, через которые серверы ожидают поступление запросов, а опция -p задает вывод имен серверов, связанных с этими портами. Как и ps, утилита netstat генерирует большой объем данных, поэтому желательно перенаправить вывод в файл или передать входные данные программе less или more.

Несмотря на то что netstat является чрезвычайно полезным инструментом, при использовании этой программы необходимо помнить, что для серверов, запускаемых посредством суперсервера, netstat будет генерировать неверные данные. Сообщая о том, какой сервер ожидает обращение через определенный порт, она вместо имени сервера выведет имя суперсервера.

Использование программ сканирования

Мощными инструментами, которые можно использовать для поиска серверов, выполняющихся в системе, являются внешние программы сканирования. В качестве примеров подобных программ можно привести Nessus (http://www.nessus.org), SAINT (http://www.wwdsi.com/saint/) и Nmap (http://www.insecure.org/nmap/). Программа сканирования выполняется на любой машине, связанной по сети с компьютером, подлежащим проверке. Некоторые из таких инструментов, помимо информации о серверах, выводят также данные об используемой операционной системе, а также сведения о наличии недостатков в защите серверов. Для поиска серверов в большинстве случаев достаточно ввести имя сканирующей программы и указать имя компьютера, который следует проверить. Например, соответствующая команда может иметь вид nmap gingko.threeroomco.com. В результате вы получите список портов и имен серверов, связанных с ними.

Внешняя программа сканирования может оказаться полезной в том случае, если вы подозреваете, что сервер на вашем компьютере подвергся атаке. Установив собственный сервер в вашей системе, опытный хакер позаботится о том, чтобы скрыть следы своего вмешательства. Чтобы вы не смогли обнаружить изменения в системе, он постарается заменить средства диагностики (например, netstat) своими программами. Внешняя

программа сканирования, вероятнее всего, даст вам реальную информацию о серверах, выполняющихся в системе.

ВНИМАНИЕ Программы сканирования портов часто используются компьютерными взломщиками, которые пытаются обнаружить уязвимые места в защите системы. Эти же инструменты применяют администраторы систем для того, чтобы найти и устранить недостатки в защите компьютеров. Для того, чтобы исключить возможные недоразумения, планы по использованию программ сканирования следует согласовать с руководством.

В данной книге встречается слово хакер: здесь оно обозначает компьютерного взломщика. Однако этот термин имеет и другие значения. Хакерами часто называют специалистов высокой квалификации, имеющих большой опыт создания сложных программ, а также энтузиастов, чье увлечение программированием граничит с фанатизмом. Смысл, вкладываемый в слово хакер, обычно становится ясным из контекста.

Недостаток использования внешних программ сканирования состоит в том, что некоторые серверы могут оказаться недоступными для них. Предположим, например, что на компьютере установлены два сетевых интерфейса, а сервер, выполняющийся на этом компьютере, настроен для обработки обращений, поступающих лишь с одного из интерфейсов. Такой сервер программа сканирования не сможет обнаружить. Даже если компьютер имеет лишь один сетевой интерфейс, это не гарантирует успех, так как доступ к серверу с некоторых IP-адресов может быть запрещен с помощью брандмауэра.

Определение необходимости сервера

Получив список серверов, вам предстоит определить, какие из них необходимы для работы системы. Решить эту задачу не всегда просто. Если администратор не имеет большого опыта работы с Linux, он вполне может посчитать ненужным сервер, жизненно важный для функционирования системы. Выяснить, какие из серверов нужны, а какие нет, поможет материал, изложенный в предыдущих главах. Кроме того, вы можете обратиться к документации на сервер, а также попытаться найти нужные сведения в Internet.

Если вы не уверены, нужен ли конкретный сервер, временно отключите его и посмотрите, как отреагирует на это система. Если компьютер продолжает нормально работать, вполне вероятно, что сервер не выполняет важных функций в системе. Однако это еще не означает, что сервер не нужен. Возможно, что последствия его отключения проявятся не сразу. Предположим, например, что на вашем компьютере присутствует сервер шрифтов. Если на той же машине не установлена система X Window, то после отключения сервера шрифтов компьютер будет работать нормально. Последствия проявятся на других машинах, обслуживаемых этим сервером.

Необходимо соблюдать осторожность при отключении процессов, связанных с регистрацией пользователей. Даже если компьютер не выполняет функции сервера регистрации, удаление одного из серверов, описанных в главах 13 и 14, может стать причиной серьезной проблемы, для разрешения которой придется перезагрузить компьютер с гибкого диска. Особенно осторожно следует отключать процессы регистрации, запускаемые посредством сценариев SysV.

Отключение серверов, запускаемых с помощью суперсервера, не создает непосредственной опасности для системы. Даже если вы закомментируете в конфигурационном

файле все записи, предназначенные для запуска серверов, работа компьютера не нарушится. Но, как нетрудно догадаться, эти серверы могут выполнять важные функции по обслуживанию других компьютеров, находящихся в вашей сети.

Отключение серверов

Отключить сервер, который выполняется в системе, можно различными способами. На практике для этого применяются два основных подхода.

- Вы можете выполнить действия, противоположные тем, которые предпринимались для запуска сервера. Например, можно закомментировать запись в конфигурационном файле `/etc/inetd.conf` или переименовать сценарий запуска SysV. Подробно способы запуска серверов обсуждались в главе 4.
- Вы можете деинсталлировать сервер. Если программа, реализующая сервер, отсутствует, она не может быть запущена.

Первый способ гораздо безопаснее второго. Если впоследствии обнаружится, что сервер необходим для работы системы или других компьютеров, вы можете снова разрешить его выполнение.

СОВЕТ ✔ Запрещая работу серверов, следите за тем, чтобы информация из конфигурационных файлов не была утеряна. Например, если сервер запускается с помощью суперсервера, соответствующую запись не следует удалять, надо лишь включить в соответствующую строку символ комментариев. Аналогично, сценарий запуска SysV надо переименовать, а не удалять. Это позволит в случае необходимости снова запустить сервер.

Если вы убедитесь в том, что сервер не нужен и не понадобится в ближайшее время, удалите соответствующую программу с компьютера. Это позволит сэкономить место на диске и гарантирует, что сервер не будет случайно запущен при изменении конфигурации системы. Если же вы предполагаете, что впоследствии данный сервер может потребоваться, либо откажитесь от его деинсталляции, либо перед удалением сохраните конфигурационные файлы. Имея конфигурационные файлы, вы сэкономите время при повторной установке программы.

Использование учетных записей и паролей

Сервер, выполняющийся на компьютере, представляет собой "дверь" в систему. К сожалению, эта дверь иногда бывает открыта не только для пользователей системы, но и для "непрошеных гостей". Существуют различные способы, позволяющие закрыть эту дверь. Один из них состоит в применении брандмауэров, для создания которых применяются такие инструменты, как `iptables` (этот инструмент будет обсуждаться в главе 25). Другой способ — это тщательный контроль за использованием учетных записей и паролей. Если на компьютере имеются неиспользуемые учетные записи, это увеличивает вероятность незаконного проникновения в систему. Пароль, попавший в руки злоумышленника, предельно упрощает его задачу. В отличие от других задач администрирования, обеспечение контроля за учетными записями и паролями возможно только в том случае, если ваши пользователи помогут вам в этом. Расскажите им о том, какому риску они подвергают систему и свои данные, небрежно обращаясь со своими паролями.

Политика использования учетных записей

Для того чтобы обеспечить безопасность при работе с учетными записями, надо прежде всего разработать политику их использования. Продолжая аналогию между сервером и дверью в систему, учетную запись можно сравнить с ключом к одной, а возможно, и к нескольким дверям. Уменьшая число ключей, вы тем самым снижаете вероятность утери одного из них. Очевидно, что для многих серверов необходимы учетные записи пользователей. При отсутствии учетных записей появляются дополнительные ограничения на использование файловых серверов, а FTP-сервер может предоставлять лишь анонимный доступ. Таким образом, необходимо уметь определить, в каких случаях возникает реальная необходимость в создании учетных записей.

Для некоторых серверов решить данную проблему очень просто: набор учетных записей надо ограничить записями для сотрудников, выполняющих администрирование системы (а еще лучше, для одного администратора). Так можно поступить, если на компьютере выполняется сервер шрифтов, DHCP или временной сервер. При обращении к таким серверам учетные записи не нужны, поэтому не обязательно, чтобы они присутствовали на компьютере, на котором выполняется сервер. Другие серверы, например Web- и FTP-сервер, в зависимости от их конфигурации могут требовать, а могут не требовать учетные записи. Наконец, сервер регистрации обычно располагается на компьютере, на котором создано большое количество учетных записей пользователей.

Если на некотором компьютере учетные записи необходимы, вам следует разработать политику, регламентирующую их создание. Например, компьютером, находящимся на физическом факультете университета, имеют право пользоваться сотрудники факультета и студенты, изучающие физику. Они имеют право требовать от администратора создания учетных записей, и этот факт необходимо отразить при составлении политики. Наличие формальной политики позволяет избежать неоправданного увеличения числа пользователей компьютера. Если впоследствии окажется, что ограничения, накладываемые политикой, слишком строги, вы можете разработать новую политику, однако она также не должна допускать неоднозначной трактовки. В особенности это важно, если компьютер обслуживает большое количество пользователей. Неформальная политика неминуемо приведет к появлению лишних учетных записей.

Вам также следует написать сценарий, управляющий созданием новых учетных записей, или, по крайней мере, составить соответствующую инструкцию. Особенно внимательно следует подойти к разработке правил выбора пароля. Принципы использования паролей будут подробно рассмотрены далее в этой главе. Не следует упускать из виду и вопросы определения прав доступа, устанавливаемых по умолчанию. Например, если в организации ведется работа над несколькими проектами, имеет смысл создать для каждого проекта группу, поместив в нее пользователей, участвующих в работе над этим проектом. При этом устанавливать права доступа к рабочим каталогам следует так, чтобы к содержащимся в них файлам могли обращаться только пользователи, работающие над тем же проектом. Правила, задаваемые в составе политики, могут изменяться в зависимости от характера деятельности организации. Так, например, для открытой среды можно указать, что по умолчанию должны устанавливаться права 0755 или даже 0775 и соответствующее значение umask. Если же пользователи работают с секретными данными, желательно задавать права 0700.

Контроль над учетными записями

Необходимо строго контролировать применение учетных записей. Действия по контролю сводятся к выявлению неиспользуемых записей и проверке правильности использования активных записей.

Выявление неиспользуемых записей

Не все пользовательские записи должны постоянно присутствовать на компьютере. После того как студенты заканчивают университет, а сотрудники организации переходят на другую работу, их учетные записи становятся ненужными. Для того чтобы снизить опасность несанкционированного доступа к системе, от таких записей необходимо избавиться. Если вы получили сообщение о том, что пользователь покинул организацию, вы должны сделать неактивной или удалить его запись. Однако далеко не всегда руководство оповещает системного администратора об увольнении сотрудников. В этом случае вы можете указать, что по прошествии некоторого времени запись должна стать недействительной. Сделать это можно с помощью команды

```
# usermod -e 2003-07-04 george
```

Данная команда сообщает системе о том, что срок действия учетной записи `george` заканчивается 4 июля 2003 года. (Ограничить срок действия записи можно также при ее создании, указав при вызове команды `useradd` опцию `-e`.) Такой подход очень удобен в тех случаях, когда вы наперед знаете время, начиная с которого учетная запись не будет использоваться. Например, администраторы знают об окончании студентами курса обучения и о завершении работы сотрудников, принятых на контрактной основе.

Менее радикальный подход состоит в ограничении срока действия пароля. При этом пользователь вынужден регулярно, например раз в месяц, задавать новый пароль. Указать, что пароль действителен лишь в течение ограниченного времени, можно с помощью следующей команды;

```
# chage -M 30 -W 5 george
```

Эта команда сообщает системе о том, что пользователь `george` должен задавать новый пароль каждые 30 дней и что за 5 дней до истечения срока действия текущего пароля он должен получать соответствующее сообщение. Если `george` забудет обновить пароль, учетная запись станет недоступной и для возобновления работы с ней необходимо будет обратиться к системному администратору.

Описанные подходы позволяют решить проблему неиспользуемых учетных записей, но они применимы не во всех ситуациях. Некоторые учетные записи не предполагают регистрацию пользователей, а применяются, например, для организации работы с файловым сервером или для доставки почты. В данном случае, чтобы пользователь увидел сообщение об окончании срока действия пароля, надо создать задание для инструмента `cron`, предусмотрев в нем определение времени, в течение которого пароль остается действительным, и оповестить пользователя по электронной почте. Существуют средства, позволяющие получить информацию об использовании учетных записей. Например, утилита `last` возвращает сведения о нескольких последних регистрациях пользователя в системе, а в некоторых дистрибутивных пакетах поддерживается файл протокола `/var/log/auth`, в который записывается информация о выполнении аутентификации. Приложив определенные усилия, вы можете написать сценарий, который проверял бы файл протокола и оповещал вас о том, что некоторые записи не используются в тече-

ние длительного времени. Периодический запуск такого сценария можно организовать с помощью инструмента `cron`. Если вы узнаете, что некоторая учетная запись давно не использовалась для регистрации в системе, выясните причины этого и, если запись больше не нужна, удалите ее.

Выполняя администрирование системы, надо также следить за учетными записями, срок действия которых истек. Ненужные записи следует удалить. Для поиска недействительных записей можно написать специальный сценарий. (Выявлять недействительные учетные записи можно по следующему признаку: по истечении срока действия значение третьего поля записи в файле `/etc/shadow` становится меньше, чем значение, содержащееся в восьмом поле.)

Выявление случаев незаконного использования учетных записей

Время от времени системные администраторы сталкиваются со случаями незаконного использования учетных записей. Случается это по разным причинам. Иногда встречаются недобросовестные пользователи, применяющие компьютеры для организации атаки на удаленные системы. В других случаях злоумышленникам, работающим на удаленных компьютерах, удается заполучить локальные учетные записи.

Одним из признаков незаконного использования учетной записи могут быть необычные действия пользователей, зарегистрированные в файлах протоколов `/var/log/messages` и `/var/log/secure`. (Имя файла протокола и характер записываемой в него информации различается для разных дистрибутивных пакетов.) Поскольку в файлы протоколов записывается информация о работе серверов, а не клиентов, обнаружить случаи незаконного использования учетных записей удается не всегда. Так, например, если локальный пользователь воспользуется клиентской программой `telnet` для атаки удаленного компьютера, информация об этом в файле протокола будет отсутствовать, однако нужные сведения можно получить, анализируя файлы протоколов брандмауэра. Содержимое файлов протоколов на локальной машине чаще всего позволяет обнаружить попытки взлома, предпринимаемые извне.

Анализ файлов протоколов — утомительная процедура, отнимающая много времени. Для того чтобы упростить работу администраторов, были созданы специальные инструменты. Примером инструментального средства, предназначенного для обработки файлов протоколов, является Simple Watcher (SWATCH, `http://oit.ucsb.edu/~eta/swatch/`). Данная программа позволяет искать записи в файлах протоколов по ключевым словам.

Для выявления случаев незаконного использования учетных записей можно установить в системе сервер аутентификации `auth` (в некоторых версиях Linux он называется `identd`). Когда клиентская программа, выполняемая на вашем компьютере, обращается к удаленному серверу, последний может установить связь с вашим компьютером и идентифицировать пользователя, инициировавшего обращение. Если кто-то из ваших пользователей нанес вред удаленной системе, информация о нем записывается в файл протокола и администратор уделенной системы может связаться с вами и сообщить имя пользователя, выполнявшего незаконные действия. Чтобы это стало возможно, необходимо, чтобы на вашем компьютере выполнялся сервер аутентификации, а злоумышленник не имел возможности заменить этот сервер своей программой. Сервер аутентификации входит в состав многих дистрибутивных пакетов и требует минимальной настройки. Обычно он запускается посредством суперсервера.

К сожалению, ваши возможности по выявлению случаев незаконного использования учетных записей ограничены, так как контроль над процессами даже на одном компьютере занимает слишком много времени и не под силу одному администратору.

Выбор паролей

Для того чтобы пароль можно было использовать, он должен находиться на диске компьютера. В системе Linux пароль располагается в файле `/etc/shadow` (в старых версиях Linux он находился в файле `/etc/passwd`). Пароль хранится в зашифрованном виде; для его кодирования обычно используется *алгоритм одностороннего кодирования*. Поскольку алгоритм декодирования отсутствует, может показаться, что содержимое файла паролей бесполезно для злоумышленника, однако при наличии компьютера с мощным процессором возможно подобрать пароль. Взломщики используют для этой цели словари, включающие слова многих языков, имена собственные и слова, в которых символы следуют в обратном порядке. Имея файл паролей, хакер может закодировать каждое слово из словаря и сравнить его с зашифрованными паролями. Если зашифрованное слово из словаря совпадает с записью в файле, это слово является паролем для регистрации в системе.

Таким образом, становится ясно, что наилучшим паролем является случайный набор букв, цифр и знаков пунктуации. Вероятность того, что он встретится в словаре, используемом хакером, предельно мала. Однако подобные случайные наборы трудны для запоминания, поэтому некоторые пользователи выбирают в качестве пароля общеупотребительные слово. Несмотря на то что такой пароль легко запоминается, его также легко подобрать. Поэтому вы, как системный администратор, должны научить своих пользователей правильно выбирать пароли. Процедура создания пароля состоит из двух этапов: генерации базовой последовательности символов и ее модификации.

Для создания базовой последовательности надо выбрать два несвязанные между собой слова и объединить их. В результате получится слово, отсутствующее в любом словаре, например bunpen. Можно также придумать легко запоминающуюся фразу и использовать в качестве базовой последовательность, составленную из первых букв слов, входящих в фразу. Например, из фразы "yesterday I went to the dentist" можно составить слово `yiwttd`. Такой набор букв легко запомнить, и в то же время он отсутствует в словарях. (В данном примере я использовал последовательность из шести символов. Дело в том, что в результате модификации длина пароля увеличится, а в большинстве систем длина пароля не должна превышать восемь символов.) Несмотря на то, что сгенерированные последовательности символов отсутствуют в словарях, не исключено, что хакер использует описанный алгоритм для расширения своего словаря. Поэтому созданная вами базовая последовательность должна быть модифицирована. Ниже приведены возможные варианты такой модификации.

- **Изменение регистра некоторых символов.** Если в вашей системе при распознавании пароля учитывается регистр символов, измените базовую последовательность так, чтобы в ней присутствовали как прописные, так и строчные буквы. Например, составленные выше последовательности можно представить в виде BUnPeN и `YiWTtd`. Если же система при проверке пароля не учитывает регистр символов, то подобная модификация не скажется на уровне защиты.

- **Добавление цифр и знаков пунктуации.** Добавив выбранные случайным образом цифры или знаки пунктуации в случайные позиции базовой последовательности, вы получите пароль наподобие BU3nP&eN или `Y+iWTtd2`.

- **Изменение порядка следования символов.** Если в базовой последовательности вы объединили два слова, измените порядок следования символов в одном из них на обратный. На основе одного из приведенных выше примеров таким способом получим пароль `BU3nNe&P`.

Вы можете модифицировать базовую последовательность и другими способами. Несмотря на то, что полученный в результате пароль выглядит как случайный набор символов, он легко запоминается. Записывать пароль на бумаге или в файле нельзя. Если ваша записка или файл, содержащий пароль в незакодированном виде, попадет в чужие руки, важные данные, содержащиеся на вашем компьютере, могут стать достоянием посторонних лиц.

Для того чтобы проверить, насколько хорош выбранный вами пароль, воспользуйтесь одной из программ, применяемых взломщиками для подбора паролей, например Crack (`http://www.users.dircon.co.uk/~crypto/`). Если эта программа сможет подобрать ваш пароль, значит, процедуру выбора пароля надо повторить.

ВНИМАНИЕ Используя программу подбора паролей, запускайте ее на компьютере, не подключенном к сети, иначе результатами вашей работы может воспользоваться хакер. Следует заметить, что администрации многих организаций возражают против использования программ подбора паролей, поэтому планы, связанные с применением подобных программ, следует согласовать с руководством.

Выбрав пароль, надо принять меры для того, чтобы сохранить его в секрете. Как было сказано выше, записывать пароль нельзя. Недопустимо также сообщать его кому-нибудь (даже членам семьи или сотрудникам). Вы должны объяснить пользователям, что их пароли вам не понадобятся. Иногда бывает, что взломщик звонит пользователю, представляется системным администратором и просит сообщить пароль. Ваши пользователи не должны поддаваться на подобную уловку.

Даже если пользователь удачно выберет пароль и никому не сообщит его, существуют способы узнать его. Злоумышленник может незаметно подсмотреть, какие клавиши нажимает пользователь при вводе пароля. Проще всего реализовать этот способ в компьютерном классе. На рабочем месте сотрудника сделать это достаточно сложно. Пароль также может быть похищен в результате "подслушивания". Многие сетевые карты можно перевести в режим сбора поступающих на них пакетов. В результате пароль, передаваемый от клиента серверу, будет перехвачен. Это можно сделать как в локальной сети, так и в Internet. Чтобы уменьшить риск подслушивания пароля, в сетях Ethernet надо вместо концентраторов применять коммутаторы. В отличие от концентраторов, коммутаторы передают пакеты только тем компьютерам, которым они предназначены. При использовании концентраторов пароль может быть перехвачен лишь в том случае, когда злоумышленнику удастся разместить программу подслушивания на компьютере, на котором выполняется сервер, либо на компьютере, на котором работает клиент. Еще лучше использовать программные средства, выполняющие шифрование пароля. В этом случае подслушивание становится бессмысленным.

Своевременное обновление системы

Многие системы, которые считаются уязвимыми для атак извне, приобрели подобную репутацию лишь из-за отсутствия должной поддержки. Считанные минуты, потраченные на получение и установку дополнительных модулей, предназначенных для устранения недостатков в защите системы, могут сэкономить многие часы, в течение которых приходится бороться с последствиями вторжения взломщика. Если вы достаточно быстро установите дополнительный модуль, злоумышленник, вероятнее всего, не успеет воспользоваться ошибкой, найденной в системе, для своих целей.

Влияние ошибок на выполнение программ

В программном обеспечении встречаются различные ошибки, которые могут проявляться по-разному. Наличие ошибок может привести к повреждению данных или программного кода, а может изменить поведение программы. Некоторые ошибки влияют на защиту системы. Встречаются такие, которые предоставляют пользователю возможность вносить изменения в любые файлы, расположенные в различных каталогах (в том числе и в файлы, определяющие конфигурацию системы), либо запускать программы от имени других пользователей. По сути, подобные ошибки наделяют обычного пользователя привилегиями суперпользователя.

В серверах, как и в любых других программах, также могут встречаться ошибки. Однако, в отличие от обычных программ, ошибки в серверах могут повлечь за собой более тяжкие последствия для системы, так как серверы доступны всем пользователям сети. Предположим, что обычная программа, например man, содержит ошибку, в результате которой обычный пользователь может получить неограниченные полномочия. Если локальные пользователи заслуживают доверия, а у внешних пользователей нет доступа к ресурсам этого компьютера, система не пострадает. (Такое предположение не всегда оказывается верным, поэтому замеченные ошибки надо устранить как можно скорее.) Если же Web-сервер, доступный из Internet, содержит ошибку, позволяющую проникнуть в систему, ею сможет воспользоваться любой желающий.

Проблема усугубляется тем, что многие серверы запускаются от имени пользователя root. Если программа (не обязательно сервер) запускается с полномочиями обычного пользователя, возможности злоумышленника, воспользовавшегося недостатками в защите, ограничены. Например, такая программа не позволит взломщику изменить содержимое файла /etc/passwd. Если же программа запускается от имени суперпользователя, возможности взломщика резко возрастают. В частности, он может создать новую учетную запись и пользоваться ею в дальнейшем. Привилегии root необходимы многим серверам для нормальной работы. Например, права пользователя root нужны серверам регистрации. Более того, чтобы принимать обращения через порт с номером ниже 1024, программа должна быть запущена от имени суперпользователя. (Полномочия root имеет суперсервер, но программы, запускаемые с его помощью, обычно выполняются с более низкими полномочиями.)

В результате становится ясно, насколько важно вовремя обновить программы, реализующие серверы. Не все дополнительные модули предназначены для исправления ошибок, некоторые из них призваны расширить возможности программы. Если эти возможности вам не нужны, устанавливать такой модуль не обязательно. Если же вы узнали о появ-

лении дополнения к системе, устраняющего недостаток в ее защите, установите его как можно скорее.

Источники информации о дополнениях к системе

Информацию о появлении дополнений к программным продуктам можно получить из следующих источников.

- **Web-узлы и списки рассылки, посвященные программным пакетам.** Для поддержки большинства пакетов, в том числе серверных программ, организуются официальные Web-узлы, списки рассылки и группы новостей. Просматривая материалы, опубликованные на Web-узлах, а также сообщения в списках и группах новостей, вы сможете своевременно получить информацию о появлении дополнительных модулей. Для получения данных таким способом требуется много времени, так как в системе Linux используются десятки серверов. Подобный подход приемлем в том случае, если необходимо выяснять положение дел с одним-двумя редко используемыми программными пакетами.

- **Web-узел конкретной версии Linux.** Для каждого дистрибутивного пакета Linux поддерживается свой Web-узел, на котором публикуется новая информация о системе, в том числе сведения о дополнениях к программам. В состав подобного Web-узла включаются Web-страницы с данными о серверах, предназначенных для выполнения в системе. В частности, на этих Web-страницах размещаются сведения о недостатках в защите серверов и способах их устранения. Преимущество подобного подхода состоит в том, что вам нет необходимости обращаться к различным Web-серверам; все нужные сведения вы можете получить на одном узле. Однако такой подход имеет свой недостаток. Для того чтобы данные о дополнениях к программам попали с узлов компаний-разработчиков на узел дистрибутивного пакета Linux, необходимо определенное время. Иногда это время составляет несколько минут, но чаще всего информация появляется на Web-узле с задержкой в несколько часов и даже несколько дней.

- **Универсальные источники информации о защите.** В конце данной главы приведены сведения о Web-узлах, списках рассылки и группах новостей, посвященных вопросам безопасности при работе в Internet, в частности обеспечению защиты системы Linux. Эти источники информации чрезвычайно важны и могут оказать существенную пользу каждому администратору. В них приводятся советы, позволяющие противодействовать злоумышленникам, пытающимся использовать недостатки в защите программ. Последовав этим советам, вы можете уберечь свою систему на время, пока не будут выпущены дополнительные модули, предназначенные для устранения ошибок в программах. Там же содержатся ссылки на Web-узлы разработчиков программ, где вы можете получить более подробные сведения о замеченных ошибках и способах их устранения.

В большинстве случаев последние два способа позволят вам находиться в курсе последних новостей, имеющих отношение к используемым вами программам. Большую помощь в работе также могут оказать Web-узлы, посвященные отдельным серверам. Просматривая две-три Web-страницы в день, вы сможете избежать участи многих администраторов, сети которых подверглись атаке.

Автоматическое обновление программ

Поиск дополнительных модулей вручную — утомительное занятие, занимающее много времени. Для автоматизации этого процесса были разработаны специализированные инструментальные средства. Некоторые из них описаны ниже.

- apt-get. Данная программа является стандартным компонентом Debian и систем, созданных на ее основе. Программа apt-get используется для инсталляции пакетов, а также автоматически обновляет установленное программное обеспечение. По команде apt-get update, за которой следует apt-get dist-upgrade, программа извлекает информацию в наличии дополнений и обновляет все пакеты, для которых были выпущены новые версии. Если вторую команду заменить на apt-get -s -u upgrade, программа apt-get будет предоставлять отчет об обновлениях, не инсталлируя их. Для того чтобы apt-get работала, в файле /etc/apt/sources.list необходимо указать хотя бы один узел, предназначенный для распространения дистрибутивных пакетов Debian. Существуют средства для переноса apt-get в другие системы.

- Red Hat Update Agent. Для обновления компонентов системы Red Hat используется программа Update Agent. Ее необходимо зарегистрировать, после чего она будет передавать информацию об аппаратных и программных средствах вашего компьютера на сервер Red Hat. После этого обновление системы осуществляется автоматически. Процесс настройки Update Agent достаточно сложен. Дополнительные сведения об этой программе вы можете получить по адресу http://www.redhat.com/docs/manuals/RHNetwork/ref-guide/.

Автоматические средства обновления программ позволяют эффективно устранять ошибки, снижающие уровень защиты системы, однако они имеют свои недостатки. Полагаясь на такие средства, вы принимаете на себя дополнительную ответственность. Автоматическое обновление иногда осуществляется некорректно. Например, обновленный пакет может содержать новые ошибки либо конфликтовать с другими программами (последнее чаще всего происходит в тех случаях, когда пакет, подлежащий обновлению, работает совместно с программами, написанными вами или установленными из tar-архивов). Не исключено также, что узел, управляющий автоматическим обновлением, подвергнется атаке и хакеры будут использовать его для распространения программ типа "троянский конь". Злоумышленники также могут захватить контроль над сервером DNS и перенаправлять обращения программ автоматического обновления на свой сервер. Пакеты, обновляющие программное обеспечение в системе Debian, обычно вызывают инсталляционные сценарии, требующие участия пользователя в процессе установки систем. По этой причине программу apt-get нельзя запускать с помощью cron. Даже если вы планируете регулярно вызывать данную программу, делать это необходимо вручную. (Посредством cron может вызываться лишь команда apt-get -s -u upgrade.) Средства автоматического обновления, как правило, не делают различий между дополнительными модулями, предназначенным для устранения недостатков в системе защиты, и дополнениями, которые были созданы для других целей и могут оказаться причиной возникновения проблем при работе других программ.

Несмотря на то что средства автоматического обновления очень удобны и экономят время администратора, использовать их необходимо очень осторожно. В настоящее вре-

мя разрабатываются средства авторизации дополнений, которые существенно повысят надежность инструментов, предназначенных для обновления программ.

Выявление случаев незаконного доступа к системе

Как известно, иногда злоумышленники взламывают даже системы, для которых была разработана формальная политика, в которых учетные записи создавались с соблюдением всех правил и в которых отсутствуют ненужные серверы. Вовремя выявив факт незаконного проникновения в систему, вы можете минимизировать урон от атаки. Существуют специальные инструменты, предназначенные для выявления попыток взлома и выполняющиеся в системе Linux. Кроме того, вам необходимо распознать признаки, свидетельствующие о незаконном доступе к системе.

Инструменты, выявляющие попытки вторжения

Если взломщик проникает в систему, он изменяет ее конфигурацию в соответствии со своими потребностями. В зависимости от характера вторжения изменяется внешний вид Web-страниц, в файлах протоколов появляются новые записи, упрощающие обращение злоумышленников к системе, изменяются коды программ, а также появляются другие "сюрпризы". К сожалению, предсказать, какие именно действия предпримет взломщик, невозможно. Именно поэтому бороться с последствиями вторжения крайне сложно; если вы не знаете, что предпринял взломщик и каковы были его цели, нельзя доверять ни одной из системных программ. Радикальное решение — удалить с дисков компьютера всю информацию и повторно установить систему или восстановить ее с помощью резервной копии, сделанной еще до атаки.

Поскольку взломщик модифицирует системные файлы, наличие измененных файлов может служить признаком атаки. Обнаружить факт проникновения в систему можно лишь в том случае, если администратор заранее сохранил информацию о состоянии основных системных файлов, например, файла `/etc/passwd` и исполняемых программ в каталоге `/bin`. Эта информация должна храниться в закодированном виде либо ее следует записать на сменный носитель. Эти данные необходимо периодически использовать для проверки целостности файлов. Если файл, который не должен был подвергаться изменениям, окажется модифицированным, есть все основания полагать, что система была взломана. (Необходимо учитывать, что некоторые файлы мог изменить сам администратор. Например, при создании новой учетной записи данные записываются в файл `/etc/passwd`.)

Использование базы данных пакетов

Во многих версиях Linux есть инструмент, который можно использовать для контроля целостности файлов. Речь идет о базе данных пакетов. Система управления пакетами Debian и система RPM сохраняют в базе данных информацию об инсталлированных программах. Для сравнения программы на диске с исходным содержимым пакета надо указать опцию `--verify` (или -V) программы `rpm`. Ниже приведен пример вызова данной команды.

```
# rpm -V postfix
S.5.... T c /etc/postfix/aliases
S.5.... T c /etc/postfix/main.cf
```

В результате выполнения программы выводится информация о файлах, состояние которых не соответствует исходному. В начале каждой строки выходных данных содержится набор признаков, сообщающих о характере несоответствия файлов. Например, буква "S" указывает на то, что размер файла изменился, цифра "5" свидетельствует о несоответствии сумм MD5, а буква "T" означает, что изменилось время модификации файла. Сообщения, отображаемые в данном примере, не являются признаком атаки, так как файлы, указанные программой, могут периодически изменяться при настройке пакета. Если же вы выясните, например, что был изменен исполняемый файл Postfix, вам необходимо начать поиски других признаков вторжения, а впоследствии предпринять меры для устранения последствий атаки.

В системе Debian аналогичные функции выполняет утилита dlocate, однако она не входит в состав Debian 2.2. Установив данную программу, вы сможете выполнить команду наподобие следующей:

```
# dlocate -md5check postfix
```

При выполнении данной команды проверяются суммы MD5 для содержимого пакета postfix и генерируется отчет о том, совпадают ли эти суммы для каждого файла.

Вместо проверки каждой программы вы можете проверить целостность всех пакетов, используя команду rpm -Va. Выходные данные будут насчитывать сотни строк, большинство из которых сообщают об изменениях конфигурационных файлов и файлов данных. Такие сообщения можно не принимать во внимание. Учитывая большой объем данных, желательно перенаправить выход в файл либо передать сгенерированную программой информацию утилите more или less.

Программы rpm и dlocate имеют существенные недостатки. Один из них состоит в том, что после инсталляции пакета нельзя выяснить, кто внес изменения в конфигурационный файл: системный администратор или взломщик. Кроме того, при желании взломщик может легко скрыть следы своего вмешательства. Для этого ему надо лишь использовать для установки модифицированных программ диспетчер пакетов. Например, если злоумышленник хочет заменить оболочку /bin/bash, ему достаточно установить новый RPM-пакет bash. В результате вызов rpm -Va не выявит изменений. По этой причине не следует полностью полагаться на диспетчер пакетов; желательно использовать наряду с ним дополнительные инструменты, предназначенные для выявления вмешательства в работу системы. Общее правило можно сформулировать так. Если диспетчер пакетов обнаружил изменения файлов, полученное сообщение следует рассматривать как признак того, что система подверглась атаке. Если же диспетчер пакетов не смог выявить изменения, этот факт не может быть гарантией целостности системы.

Использование Tripwire

Для выявления случаев несанкционированного доступа к системе разработан инструмент Tripwire (http://www.tripwire.org). Эта программа поставляется со многими версиями Linux. Если же в вашем дистрибутивном пакете она отсутствует, скопируйте ее с Web-узла. Версию Tripwire, входящую в состав дистрибутивного пакета, инсталлировать гораздо проще, чем пакет, скопированный с Web-узла, так как в ней заранее учтены набор файлов, используемых в системы, и их расположение. Tripwire сохраняет информа-

цию о файлах в базе данных, этим данный инструмент напоминает диспетчеры пакетов, однако в нем реализованы специальные функции, превращающие его в специализированное средство обеспечения защиты. Tripwire может быть сконфигурирован для хранения информации о произвольном наборе файлов, причем сведения о файлах записываются в базу данных после инсталляции. Вы можете создать базу данных после того, как внесете необходимые изменения в конфигурационные файлы. В процессе работы Tripwire шифрует информацию, что не дает возможности взломщику изменить базу данных. Для обеспечения сохранности базы Tripwire поместите ее на сменный носитель, запретив запись данных.

Tripwire может работать в одном из следующих режимов.

- **Генерация базы данных.** При первом запуске Tripwire необходимо инициализировать базу данных. Для этого после редактирования конфигурационного файла вызовите команду `tripwire -initialize`. Выполнение этой процедуры может продлиться достаточно долго, так как программа Tripwire должна создать контрольные суммы всех файлов, контроль над которыми был предусмотрен при настройке данного инструмента. Сформированная база данных помещается в подкаталог `databases` текущего каталога, но желательно переместить ее в каталог `/usr/lib/tripwire/databases`. Запись данных в этот каталог следует запретить.

- **Обновление базы данных.** Если вы внесли изменения в систему, можете обновить базу данных Tripwire. Для этого вызовите команду `tripwire -update` *`путь_к_файлу`*, указав файл, который необходимо учесть в базе данных.

- **Интерактивное обновление базы данных.** Если внесенные вами изменения затрагивают несколько компонентов системы или если вы установили большой пакет, запустите Tripwire в интерактивном режиме. Для этого вызовите команду `tripwire -interactive`. В этом случае программа будет отыскивать файлы, подвергшиеся изменениям, и осведомляться у вас, следует ли учитывать эти изменения в базе данных.

- **Проверка целостности системы.** Этот режим используется по умолчанию. Для того чтобы запустить Tripwire в таком режиме, достаточно ввести в командной строке `tripwire`. Проверку целостности системы желательно выполнять каждый день. Периодический вызов Tripwire можно организовать с помощью `cron`.

Работой Tripwire управляет конфигурационный файл `/etc/tripwire/tw.config`. Подобно многим другим конфигурационным файлам, строки, начинающиеся с символа #, содержат комментарии. В остальных строках указываются каталоги, предназначенные для проверки. Соответствующие записи имеют следующий формат:

`[!|=]` *`объект`* `[`*`флаги_выбора \ шаблон`*`]`

Назначение компонентов записи приведено ниже.

- `!`. Если данный символ предшествует имени объекта, то указанный файл или каталог не подлежит проверке. Если объектом является каталог, содержащиеся в нем подкаталоги также не проверяются.

- =. Данный символ указывает на то, что каталог подлежит проверке, а файлы и подкаталоги, содержащиеся в нем, не должны проверяться. Обычно этот символ указывается для рабочих каталогов пользователей. Если символ = предшествует имени каталога, то Tripwire лишь сообщает о том, что файлы или подкаталоги были созданы или удалены, но не приводит подробную информацию об этих файлах.

- *объект*. Объект представляет собой имя файла или каталога, предназначенного для проверки, например /etc или /usr. Если в качестве объекта указан каталог, выполняется проверка всех его подкаталогов. Не проверяются лишь подкаталоги, представляющие собой отдельные файловые системы. Например, если содержимое каталогов /usr и /usr/local находится в разных разделах, то, чтобы проверить все дерева подкаталогов, вы должны создать записи как для /usr, так и для /usr/local.

- *флаги_выбора*. Данный компонент записи указывает Tripwire на то, какие типы изменений должны быть отражены в отчете. Флаги задаются в формате [+|-] [pinugsamcl23456789] . . .. Символ + или - разрешает или запрещает включать сведения в отчет. Остальные символы определяют типы проверки. Например, p задает поверку прав доступа, i — проверку индексных дескрипторов (mode), n соответствует числу связей, и — идентификатору владельца файла, g — идентификатору группы, s — размеру файла, a — времени доступа, m — времени модификации, c — времени создания индексного дескриптора, а числа 0-9 задают особенности контрольного суммирования.

- *шаблон*. Вместо флагов выбора вы можете задать шаблон. По умолчанию принимается шаблон R, соответствующий +pinugsm12-ac3456789. В качестве примеров других шаблонов можно привести L (+pinugsacm123456789), используемый для проверки файлов протоколов, N (+pinugsamc123456789), который выполняет подробную проверку, но проверка эта занимает много времени, и E (-pinugsamc123456789), игнорирующий все типы проверки.

Сформировав конфигурационный файл Tripwire, вы должны запустить программу в режиме генерации базы данных. В результате файл базы данных будет создан в каталоге databases. При последующих запусках Tripwire будет отыскивать файл базы данных в каталоге /usr/lib/tripwire/databases. Этот файл очень важен, поэтому вы должны обеспечить его сохранность. Способы сохранения файла базы данных описаны ниже.

- Файл базы данных может храниться на сменном носителе, защищенном от записи, например на дискете или компакт-диске. Если вы собираетесь выполнять проверку, периодически запуская Tripwire с помощью cron, носитель может быть постоянно смонтирован (такой подход создает неудобства при работе с системой). Если вы предполагаете запускать Tripwire вручную, носитель можно монтировать непосредственно перед проверкой.

- Для сохранения файла базы данных можно создать отдельный небольшой раздел и монтировать его только для чтения. При этом целесообразно предпринять дополнительные меры по обеспечению сохранности базы данных. Например, вы можете отформатировать раздел по соглашениям операционной системы, отличной от

Linux. Это затруднит действия взломщика по поиску пароля, но не гарантирует целостность файла. Опытный хакер способен преодолеть подобные преграды.

- Вы можете записать на резервный носитель копию файла базы данных и перед началом проверки сравнить файлы. Если файл базы данных будет отличаться от резервной копии, это само по себе уже свидетельствует о факте вмешательства в работу системы.

- Tripwire поддерживает кодирование данных, поэтому файл базы данных можно сохранить в зашифрованном виде. Чтобы изменить содержимое файла базы данных, взломщик должен знать пароль, использовавшийся при кодировании.

Предприняв необходимые меры для сохранения файла базы данных, вы сможете контролировать целостность файлов на вашем компьютере. Необходимо лишь помнить, что база данных Tripwire должна создаваться непосредственно после установки Linux. Если от момента инсталляции системы до момента создания базы данных пройдет хотя бы один-два дня, то вполне возможно, что за это время система подвергнется атаке и в базу данных будут записаны сведения об измененных файлах.

Способы, позволяющие выявить вторжение в систему

Помимо применения специализированных инструментов, таких как Tripwire, обнаружить факт вторжения можно путем анализа различных признаков. Некоторые из этих признаков перечислены ниже.

- **Записи в файлах протоколов.** Файлы протоколов, располагающиеся в каталоге `/var/log`, содержат информацию о работе серверов. Как было сказано ранее, для обработки файлов вручную требуется слишком много времени, и не каждый администратор может позволить себе заняться этим. Однако с помощью специализированных инструментов, например SWATCH, вы можете выявить факты, свидетельствующие о взломе. Например, если вы обнаружите, что пользователь, находящийся в отпуске в другой стране, на днях зарегистрировался на сервере, это означает, что его учетной записью воспользовался кто-то другой.

- **Некорректная работа сервера.** Если сервер без видимых причин начинает работать некорректно, это может быть признаком вторжения, так как взломщики, пытаясь изменить конфигурацию сервера, часто портят конфигурационные файлы. Очевидно, что неправильная работа сервера может быть вызвана целым рядом других причин, но версию вторжения извне также нельзя сразу отбрасывать.

- **Жалобы пользователей.** Для регистрации в системе взломщики часто используют учетные записи других пользователей. Настраивая окружение пользователя в соответствии со своими потребностями, они нередко портят конфигурационные файлы, отвечающие за формирование среды. Поэтому жалобы пользователей на то, что система внезапно начала отвергать пароль или что параметры оболочки по непонятным причинам изменились, нельзя оставлять без внимания.

- **Появление "странных" файлов.** Если вы обнаружили, что в системе появился файл, который вы не создавали и который не мог создать никто из пользователей, это может быть программа, используемая для проникновения в систему. Чтобы скрыть

свои действия по взлому, злоумышленники часто удаляют конфигурационные файлы, поэтому отсутствие такого файла также является тревожным симптомом.

- **Неожиданное увеличение сетевого трафика.** Если трафик неожиданно увеличился, возможно, что причиной этого является деятельность злоумышленника, связанная со взломом системы. Не исключено, конечно, что трафик увеличился в результате роста популярности вашего узла. (Причиной внезапного роста популярности может быть появление ссылки на ваш узел на одном из часто посещаемых узлов.) Убедившись в увеличении трафика, вам следует проверить характер соединений, устанавливаемых с этого компьютера. Например, если с компьютера, на котором выполняется только Web-сервер, устанавливаются Telnet-соединения с другими машинами вашей сети, это свидетельствует о том, что компьютер используется взломщиками для дальнейшего проникновения в вашу сеть.

Существуют также другие признаки вторжения в систему. Дело в том, что большинство атак предпринимается не опытными хакерами, а малограмотными взломщиками, которые пользуются чужими сценариями. Такие сценарии неминуемо оставляют на компьютере следы своей работы. К сожалению, каждый сценарий, написанный с целью взлома системы, проявляет себя по-своему, поэтому составить перечень признаков, по которым можно было бы выявить вторжение в систему, крайне сложно. Материалы на эту тему публикуются на многих Web-узлах, посвященных вопросам защиты.

Встретившись с необычным поведением программ, не следует пытаться объяснить все подобные случаи действиями злоумышленников. Причинами этого могут быть сбои в работе аппаратуры, некорректное содержимое конфигурационных файлов, ошибки в программах и т. д.

Действия при обнаружении факта взлома системы

При выявлении факта вторжения в систему рекомендуется выполнить следующие действия.

- **Отключить компьютер от сети.** Компьютер, который подвергся атаке, может представлять собой угрозу для других узлов. Если на этом компьютере содержатся важные данные, то чем дольше он будет работать в сети, тем больше вероятность того, что эти данные попадут в руки злоумышленника.

- **Выяснить причины, в результате которых взломщик смог получить доступ к системе.** Необходимо убедиться, что попытка взлома имела место и была успешной. Как было замечено ранее, многие проблемы, связанные с недостатками в аппаратных и программных средствах, могут быть ошибочно приняты за взлом системы. Следует также попытаться выяснить путь, по какому взломщик смог проникнуть в вашу систему. Если вы устраните последствия взлома, но не выявите причины, которые сделали это возможным, злоумышленник сможет снова воспользоваться теми же средствами. К сожалению, установить конкретные недостатки в защите системы чрезвычайно сложно, поэтому в большинстве случаев незаконного проникновения извне системные администраторы ограничиваются обновлением версии системы и принимают меры общего характера, направленные на повышение уровня защиты.

- **Создать резервные копии важных данных.** Если резервные копии данных создавались слишком давно, вам следует скопировать важную информацию на резервный

носитель. Имеет также смысл создать копию всей системы, подвергшейся атаке. Копия может пригодиться вам впоследствии для анализа особенностей проникновения в систему.

- **Устранить последствия атаки.** К сожалению, действия по устранению последствий взлома не ограничиваются восстановлением файлов, которые были изменены злоумышленником. Вы можете пропустить какой-либо из файлов и предоставить тем самым возможность взломщику повторить атаку. Вам необходимо полностью очистить жесткий диск или, по крайней мере, те разделы, в которых содержались компоненты системы Linux. При необходимости раздел /home можно оставить нетронутым. Затем следует повторно инсталлировать систему с нуля либо восстановить ее с резервной копии, которая была создана до атаки. На этом этапе на следует подключать компьютер к сети.

- **Восстановить файлы с данными.** Если на предыдущем этапе вы удалили всю информацию с жесткого диска, вам надо восстановить данные, резервные копии которых были созданы на третьем этапе. Если вы инсталлировали систему с нуля, следует также восстановить содержимое конфигурационных файлов.

- **Устранить недостатки в защите.** Если на втором этапе вам удалось выяснить причины, позволившие злоумышленнику проникнуть в систему, следует устранить их. Целесообразно также принять меры общего характера для повышения уровня защиты, например, инсталлировать в системе инструмент Tripwire (если он не был инсталлирован ранее) или установить и настроить брандмауэр.

- **Подключить компьютер к сети.** После устранения всех проблем надо подключить компьютер к сети и возобновить его работу.

Помимо описанных выше действий, желательно также принять дополнительные меры. Вы можете, например, проследить маршрут к удаленному узлу, с которого была предпринята попытка атаки, и передать сообщение системному администратору. Хакеры часто взламывают систему, зарегистрировавшись на компьютере, который был взломан ранее. Если вы сообщите администратору о том, что с его машины была предпринята атака на ваш компьютер, он, вероятнее всего, займется проверкой системы. Если взломщик нанес вам большой урон, вы можете также обратиться за помощью к органам правопорядка.

Источники информации о защите систем

Поскольку подготовка книги к публикации занимает достаточно длительное время, я не могу сообщить последние сведения о методах взлома и борьбе с ними, информацию о недостатках в системе защиты и другие данные. К тому моменту, как книга попадет в руки читателя, они, безусловно, устареют. Поэтому в данной главе были приведены лишь общие сведения о некоторых способах, позволяющих выявить факты взлома, об инструментах, с помощью которых можно контролировать текущее состояние системы, и о действиях, которые необходимо предпринять в случае, если ваш компьютер подвергся атаке. Однако, чтобы успешно осуществлять администрирование системы, вам необходимо быть в курсе последних новостей, связанных с защитой. Существует целый ряд Web-серверов, списков рассылки и групп новостей, содержащих сведения по этим вопросам; ссылки на эти источники приводятся в данном разделе.

Web-узлы, посвященные вопросам защиты

В Internet поддерживаются Web-узлы, информирующие практически обо всех вопросах, связанных с использованием компьютеров, и защита системы не является исключением. Многие Web-узлы предоставляют самую новую информацию по этой теме. Ниже перечислены Web-узлы, содержимое которых должен периодически просматривать каждый системный администратор.

- Web-узел, посвященный используемой версии Linux. Практически для каждого дистрибутивного пакета Linux в Internet создан Web-узел, содержащий информацию о данной версии системы. Немалую часть этой информации составляют вопросы защиты. Как правило, на таких узлах публикуются сведения о вновь обнаруженных недостатках в защите программ и ссылки на версии программ, в которых эти проблемы устранены.

- Web-узел CERT/CC. Computer Emergency Response Team Coordination Center (CERT/CC) — одна из ведущих организаций, занимающаяся проблемами защиты. Web-узел CERT/CC находится по адресу `http://www.cert.org`.

- Web-узел **CIAC**. Организация Computer Incident Advisory Capability (CIAC) поддерживает Web-узел, расположенный по адресу `http://www.ciac.org/ciac/`. На нем публикуются сведения такого же характера, как и на узле CERT/CC.

- Раздел Linux Weekly News, **посвященный** защите системы. Linux Weekly News (`http://lwn.net`) — сетевая газета, посвященная использованию Linux. В одном из ее разделов публикуются сведения о способах повышения безопасности различных дистрибутивных пакетов Linux. (URL раздела часто изменяется. Для того, чтобы попасть на соответствующую Web-страницу, надо активизировать ссылку Security на главной странице Linux Weekly News.)

- Web-узел **SecurityFocus**. На этом узле (`http://www.securityfocus.com`) публикуются новости о вопросах защиты. Информация на этом сервере представляет собой своеобразный дайджест, составленный на основе данных, представленных на узлах CERT/CC и CIAC.

На перечисленных выше узлах вы найдете сведения о средствах, используемых хакерами, и противодействии различным способам атаки, о выявленных недостатках в защите различных программных продуктов, о дополнительных модулях для различных программ, информацию о вирусах и борьбе с ними, а также другие данные подобного рода. Вам, как системному администратору, следует периодически просматривать содержимое одного-двух из этих узлов (желательно делать это ежедневно или, по крайней мере, раз в неделю).

Списки рассылки и группы новостей, посвященные вопросам защиты

Чтобы получать информацию с Web-узлов, к ним надо периодически обращаться; иногда это не совсем удобно. В качестве альтернативы Web-узлам можно рассматривать списки рассылки и группы новостей. Списки рассылки позволяют передавать новые сведения со скоростью распространения электронных писем. Среди множества списков рассылки существуют и такие, которые посвящены вопросам защиты. Некоторые из списков не

позволяют подписчикам передавать сообщения; они представляют собой лишь информационную среду, применяемую для распространения новых сведений. Если вы привыкли регулярно просматривать поступающую почту, подпишитесь на один или несколько списков рассылки. В этом случае новые сведения, касающиеся защиты, будут приходить вам практически сразу же после того, как они поступят в список.

Если вы хотите просматривать сообщения, касающиеся защиты, как только они окажутся в вашем почтовом ящике, установите фильтр Procmail, настроив его для просмотра сообщений и запуска специальной программы, которая информировала бы вас о новых сообщениях по вопросам защиты. Для привлечения вашего внимания эта программа может отображать диалоговое окно с сообщением или воспроизводить звуковой сигнал.

Группы новостей во многом напоминают списки рассылки. Подобно спискам, они доставляют данные подписчикам. Подписавшись на группы, посвященные защите, вы можете периодически просматривать их и даже написать специальный сценарий, который информировал бы вас о поступлении сообщения, содержащего заданные ключевые слова.

Ниже перечислены списки рассылки и группы новостей, в которых публикуются сведения, имеющие отношение к защите системы Linux.

- Список рассылки CERT/CC. Помимо Web-узла, CERT/CC поддерживает также список рассылки, посвященный вопросам защиты. Для того чтобы подписаться на этот список, надо отправить почтовое сообщение по адресу `majordomo@cert.org`, включив в него строку `subscribe cert-advisory`.

- Список рассылки CIAC. Подобно CERT/CC, CIAC передает материалы, имеющие отношение к защите систем, с помощью списка рассылки. Для того чтобы подписаться на данный список, надо отправить почтовое сообщение по адресу `majordomo@tholia.llnl.gov`, включив в него строку `subscribe ciac-bulletin`.

- Список рассылки Bugtraq. Данный список создан не только для распространения новых материалов, но и для организации дискуссий. Участвуя в этом списке, вы можете получать от других администраторов полезные советы по организации защиты системы. Подписаться на материалы списка можно, направив письмо по адресу `listserv@netspace.org`. В тело письма надо включить строку `subscribe bugtraq`.

- Группы новостей comp.security. В иерархии `comp.security` существует несколько групп новостей (например, `comp.security.unix`). Среди них есть группы, посвященные отдельным (даже конкретным) типам продуктов. В качестве примера подобной группы можно привести `comp.security.firewalls`.

- Группа новостей **comp.os.linux.security**. Данная группа посвящена вопросам безопасности Linux.

Поскольку большинство серверов, выполняющихся в системе Linux, используются также в системах, подобных UNIX, вопросы безопасности Linux на самом деле относятся ко всем версиям UNIX. Поэтому многие группы новостей и списки рассылки, посвященные защите Linux, не ограничиваются рамками данной системы.

Резюме

Для того чтобы обеспечить безопасность при использовании Linux, надо хорошо знать особенности работы данной системы. Выполняя администрирование, вы должны отключить ненужные серверы, правильно создать учетные записи, рассказать пользователям о недопустимости небрежного обращения с паролями, вовремя установить дополнения, предназначенные для устранения ошибок в программах, и уметь распознавать случаи незаконного проникновения в систему. Поскольку новая информация, имеющая отношение к защите, публикуется практически ежедневно, вам необходимо постоянно пополнять свои знания, просматривая содержимое Web-узлов, посвященных вопросам безопасности, а также материалы списков рассылки и групп новостей. Если вы хорошо знакомы с принципами работы Linux, то, для того, чтобы быть в курсе последних новостей, связанных с защитой, вам потребуется лишь несколько минут в день. Время, потраченное на ознакомление с новыми решениями в области защиты, многократно окупится, если вам удастся сорвать планы злоумышленника по проникновению в вашу систему.

В главах 23, 25 и 26 рассматриваются специальные средства, предназначенные для обеспечения безопасности при использовании Linux.

Глава 23

Создание поддерева chroot

Каждый сервер в процессе работы читает файлы с диска компьютера, некоторые серверы также записывают файлы на локальный диск. Получив контроль над таким сервером, взломщик может изменить конфигурацию программ, обеспечивая себе базу для дальнейшего проникновения в систему. Уменьшить возможности злоумышленника можно, ограничив сферу действий сервера подмножеством файловой системы компьютера. Это подмножество файловой системы называется *поддеревом chroot.* Если сервер выполняется в пределах такого поддерева, то важные системные файлы будут не доступны для взломщика, даже если он сможет использовать сервер в своих целях.

Нормально функционировать в рамках поддерева chroot могут не все серверы, но некоторые из них специально предназначены для работы в таком окружении. Для сервера, поддерживающего chroot, необходимо не только задать соответствующие опции в конфигурационном файле, но и создать среду chroot, в которой он будет выполняться.

Что такое поддерево chroot

Корнем дерева файловой системы Linux является каталог /. Относительно этого каталога определяется путь к любому другому каталогу. При создании поддерева chroot корневой каталог переопределяется; вместо него назначается один из каталогов файловой системы. Принцип создания поддерева chroot показан на рис. 23.1. Если в качестве нового корневого каталога задать, например, каталог /opt/chroot, то путь к любому файлу или каталогу будет определяться не относительно каталога /, а относительно /opt/chroot. В результате, если сервер попадет под контроль взломщика и тот модифицирует файл /etc/passwd, файл /opt/chroot/etc/passwd будет изменен, а системный файл паролей останется в прежнем виде.

Для создания поддерева chroot используется системный вызов chroot(). Функцию chroot() может вызывать либо сам сервер, либо программа chroot, применяемая для запуска сервера. Подробно этот вопрос будет рассмотрен далее в данной главе.

При использовании поддерева chroot должны выполняться следующие условия.

- Если программа использует конфигурационные файлы или библиотеки, а также предоставляет клиенту или принимает от него некоторые файлы, все эти файлы должны размещаться в поддереве chroot. В результате для ряда серверов размеры

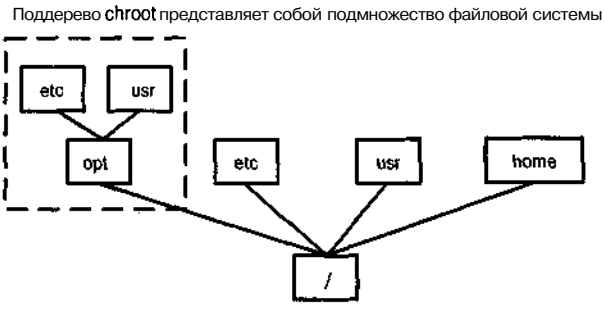


Рис. 23.1. Поддерево chroot представляет собой специальное окружение, содержащее лишь те файлы, которые необходимы для работы сервера

поддерева должны быть очень большими. Однако если сервер самостоятельно вызывает функцию chroot(), он сможет прочитать содержимое требуемых файлов до вызова chroot(), т. е. в тот момент, когда область его действий еще не будет ограничена поддеревом chroot. В этом случае часть файлов, с которыми работает сервер, может лежать за пределами поддерева chroot.

- Сервер может обращаться только к тем файлам, которые находятся в поддереве chroot. Сократив до необходимого минимума число файлов, содержащихся в каталогах поддерева, можно уменьшить риск повреждения системы в случае, если взломщик получит контроль над сервером.

- Если в поддереве chroot должно выполняться несколько серверов, для каждого из них необходимо создать отдельное поддерево. В этом случае злоумышленнику не удастся использовать один сервер для изменения конфигурации остальных.

- Поскольку поддерево chroot является подмножеством файловой системы Linux, программы, выполняющиеся за пределами данного поддерева, могут записывать файлы в каталоги, принадлежащие поддереву chroot. В зависимости от конкретных обстоятельств, этот факт можно рассматривать либо как преимущество, либо как возможный источник проблем. Вопросы доступа локальных программ к каталогам поддерева chroot будут рассматриваться далее в этой главе.

Несмотря на то что использование поддерева chroot позволяет существенно снизить опасность для компьютера, на котором выполняются серверы, данный подход имеет свои недостатки и ограничения. Одно из ограничений состоит в том, что не все серверы могут выполняться в рамках поддерева chroot. Для одних серверов подобный режим работы является вполне естественным (в качестве примера можно привести сервер FTP). Другим серверам, например Telnet, требуется более или менее полный доступ к файловой системе Linux. Таким образом, некоторые серверы неизбежно придется запускать за пределами поддерева chroot.

Поддерево chroot не устанавливается автоматически программой инсталляции пакета, его создает системный администратор. По этой причине затрудняется установка дополнений к серверу. Если вы забудете скопировать измененные файлы в нужный каталог, конфигурация сервера, выполняющегося в пределах chroot, останется неизменной.

Не следует также забывать, что процесс, выполняемый в рамках поддерева chroot с полномочиями root, может вызвать функцию chroot() и расширить область своих действий на всю файловую систему. (Организовать такой вызов достаточно сложно, но вполне возможно.) Поэтому предоставляя привилегии root серверу, выполняющемуся в пределах поддерева chroot, необходимо соблюдать осторожность. Как известно, в защите практически каждого сервера были найдены недостатки, и, несомненно, подобные недостатки будут обнаружены и в дальнейшем. Таким образом, несмотря на то, что поддерево chroot является чрезвычайно полезным инструментом, оно вовсе не гарантирует безопасность системы.

Поддерево chroot защищает компьютер от атаки извне, но оно не может помешать использовать сервер для взлома другого компьютера. Так, например, если в пределах поддерева chroot выполняется сервер DNS и злоумышленнику удалось получить контроль над ним, он не обязательно будет пытаться проникнуть с его помощью в вашу систему. Заменив записи в конфигурационном файле сервера, он может перенаправить запросы клиентов на свой компьютер. Если же, взломав сервер, работающий в рамках поддерева chroot, злоумышленник организует с его помощью атаку на удаленные узлы, то с точки зрения администратора удаленной сети это будет выглядеть так, как будто атака предпринимается пользователем, работающим на вашем компьютере.

И наконец, если вы запустили один сервер в рамках поддерева chroot, остальные серверы могут выполняться за пределами поддерева, создавая тем самым опасность для системы. Более того, если области файловой системы, доступные разным серверам, перекрываются, то не исключено, что один сервер можно будет использовать для изменения конфигурации другого, что создаст дополнительные возможности для незаконного проникновения в систему.

Формирование среды chroot

Для того чтобы сервер мог работать в рамках поддерева chroot, необходимо в первую очередь сформировать само поддерево. Надо создать требуемые каталоги и скопировать в них системные файлы и файлы сервера. Другими словами, вам следует сформировать в пределах поддерева усеченный вариант системы Linux, в котором отсутствовало бы большинство программ и конфигурационных файлов, имеющихся в полном варианте системы.

В данном разделе обсуждаются лишь общие вопросы создания поддерева chroot. Более подробно конфигурация серверов для работы в пределах поддерева будет рассмотрена в следующем разделе. Там же будет приведен пример подготовки сервера BIND для выполнения в рамках поддерева chroot.

Создание поддерева

Для создания поддерева chroot сначала необходимо сформировать само поддерево. Его можно разместить в любой позиции файловой системы, за исключением псевдосистем, таких как /proc. Если сервер должен иметь возможность записывать файлы, для подкаталогов необходимо задать соответствующие права доступа. В примере, рассмотренном выше, для создания поддерева chroot использовался каталог /opt/chroot,

но реально роль корневого каталога поддерева может выполнять практически любой каталог файловой системы.

В поддереве chroot надо создать некоторые из каталогов и подкаталогов, присутствующие в обычной файловой системе. Вероятнее всего, вам потребуется лишь ограниченное количество подкаталогов Linux. Чаще всего для выполнения сервера в поддереве chroot приходится создавать каталоги /bin, /sbin, /usr, /lib, /etc и /var. В эти каталоги не следует копировать файлы, присутствующие в соответствующих каталогах файловой системы Linux; нельзя забывать, что поддерево chroot создается именно для того, чтобы ограничить набор инструментов, доступных серверу.

Если в пределах поддерева chroot должно выполняться несколько серверов, надо для каждого из них создать отдельное поддерево. Например, если в таком режиме предполагается запустить серверы FTP и sendmail, вы можете использовать в качестве корневых каталогов поддеревьев каталоги /opt/chroot/ftp и /opt/chroot/sendmail.

Копирование файлов сервера

Сформировав поддерево chroot, надо скопировать в содержащиеся в нем каталоги требуемые файлы. Набор необходимых файлов зависит от особенностей сервера. Если сервер самостоятельно вызывает функцию chroot(), вам нет необходимости размещать в пределах поддерева chroot исполняемые файлы сервера. Вы можете запустить сервер за пределами поддерева chroot, указав ему расположение поддерева. После вызова chroot() сфера действий сервера будет ограничена сформированным вами поддеревом. Поскольку сервер, самостоятельно вызывающий chroot(), может читать конфигурационные файлы, размещенные за пределами поддерева, число файлов, которые необходимо поместить в каталоги поддерева chroot, сводится к минимуму. Вам придется скопировать лишь те файлы, которые требуются серверу для работы после вызова chroot(). Примером сервера, функционирующего подобным образом, является анонимный сервер FTP. (Подробно вопросы работа сервера FTP рассматривалась в главе 21.)

Если сервер не вызывает самостоятельно функцию chroot(), его следует запускать посредством утилиты chroot. При этом в каталоги поддерева chroot необходимо скопировать исполняемые и конфигурационные файлы сервера, а также все файлы, необходимые серверу в процессе работы. Кроме того, иногда приходится помещать в каталоги поддерева chroot некоторые системные файлы. Определить набор файлов, необходимых серверу, достаточно сложно. Для того чтобы выяснить, какие файлы требуются в процессе работы, надо прочитать документацию, а если нужные сведения там отсутствуют, вам придется проанализировать содержимое дистрибутивного пакета. Для получения информации о файлах, содержащихся в пакете, можно использовать инструменты tar, rpm или dpkg. Очевидно, что копировать в каталоги поддерева chroot надо не все файлы; например, для работы сервера не нужна документация. Чтобы выяснить, какие файлы необходимы серверу, можно использовать программу strace. Вызвав команду strace *имя_серверной_программы*, вы получите сведения о файлах, используемых в процессе работы сервера, в том числе имена файлов, которые сервер открывает самостоятельно.

Вместо копирования файлы можно переместить в каталоги поддерева chroot. При этом вы будете иметь гарантию того, что после запуска сервер будет выполняться в пределах поддерева.

Копирование системных файлов

После того как вы разместите в пределах поддерева chroot файлы сервера, вам следует скопировать в каталоги поддерева некоторые системные файлы. Для работы серверов часто требуются следующие типы файлов.

- **Библиотеки.** Во время работы многие серверы используют динамические библиотеки. Обычно они хранятся в каталоге /lib или /usr/lib. Выяснить, какие библиотеки нужны конкретному серверу, позволяет программа ldd. Например, чтобы определить, какие библиотеки использует сервер имен, надо выполнить команду ldd /usr/sbin/named. Файлы библиотек надо поместить в каталог поддерева chroot.

- **Программы поддержки.** Для работы некоторых серверов нужны дополнительные программы. Например, если Web-сервер поддерживает CGI-сценарии, ему могут понадобиться интерпретатор Perl (/usr/bin/perl)и файлы, обеспечивающие работу этого интерпретатора. Исполняемый файл Perl и дополнительные файлы надо скопировать в соответствующий каталог поддерева chroot. Кроме того, программы, необходимые для работы сервера, могут, в свою очередь, использовать файлы библиотек. В некоторых случаях объем данных, применяемых для поддержки языков сценариев, намного превышает объем Web-сервера.

- **Файлы устройств.** Ряд серверов непосредственно обращается к файлам устройств. Например, сервер резервного копирования взаимодействует с накопителем на магнитных лентах, а некоторым библиотекам и программам нужны специальные файлы устройств, такие как /dev/zero или /dev/null. Обычно файлы устройств располагаются в каталоге /dev. Копировать файлы из этого каталога бессмысленно, вместо этого надо повторно создать их в поддереве с помощью mknod. Соответствующая команда может выглядеть следующим образом: mknod /opt/chroot/dev/stO c 9 0. Файлы устройств предоставляют доступ к ресурсам компьютера, поэтому создавать их в поддереве chroot следует только в том случае, когда они действительно необходимы.

- **Специальные файловые системы.** Иногда серверы используют специальные файловые системы или инструменты, предназначенные для работы с такими файловыми системами. Например, некоторые серверы обращаются к /proc. Подобные каталоги нельзя непосредственно копировать. Вместо этого надо создать дополнительную запись в файле /etc/fstab и смонтировать файловую систему в пределах поддерева chroot. Исходную систему /proc нельзя удалять, ее надо дублировать. Необходимо помнить, что при наличии доступа сервера к /proc взломщику автоматически предоставляются дополнительные возможности для контроля над системой.

- **Базы данных с информацией о пользователях.** Во время работы ряд серверов обращается к /etc/passwd, /etc/group, /etc/shadow и другим файлам, содержащим информацию о пользователях. Серверам, которые используют Pluggable Authentication Module, необходима инфраструктура PAM, в частности, файл /etc/pam.conf, содержимое /etc/pam.d и /etc/security, а также библиотеки в файлах /lib и /lib/security (в именах файлов библиотек содержатся

символы рат). Для того чтобы выяснить, какие файлы необходимо скопировать в каталоги поддерева chroot, надо проанализировать содержимое пакета PAM.

- **Файлы протоколов.** Если сервер создает во время работы файлы протоколов, вам необходимо подготовить каталоги для их размещения. Некоторые серверы используют при создании файлов протоколов демон syslogd, в этом случае вам надо скопировать данную программу в поддерево chroot. Многие серверы можно сконфигурировать так, чтобы протоколирование выполнялось без использования syslogd.

Для серверов, самостоятельно вызывающих функцию chroot(), обычно приходится копировать в каталоги поддерева гораздо меньше файлов, чем для серверов, запускаемых с помощью программы chroot. Если сервер вызывает chroot(), то перед вызовом данной функции он обычно загружает библиотеки, системные файлы и остальные необходимы ему данные.

Чтобы не создавать угрозу безопасности системы, помещайте в каталоги поддерева chroot лишь минимальный набор файлов. Копировать файл следует только в том случае, если вы твердо знаете, что этот файл используется в работе сервера. Что же касается остальных файлов, выяснить, необходимы ли они, можно, запустив сервер на выполнение *(если в* сервере предусмотрен режим отладки, желательно включить его). Если серверу недостает какого-либо файла, он сообщит об этом.

Настройка сервера для работы в рамках поддерева chroot

Создав поддерево chroot, можно приступать к его использованию. Для этого надо сконфигурировать сервер для работы в рамках поддерева, организовать запуск сервера и обеспечить контроль доступа к поддереву chroot извне. Решение этих задач рассматривается в данном разделе. Здесь же будет приведен пример запуска сервера имен в пределах поддерева chroot.

Запуск сервера в рамках поддерева chroot

Если сервер осуществляет вызов функции chroot(), вероятнее всего, что в его конфигурационном файле содержится одна или несколько опций, предназначенных для управления выполнением в рамках поддерева chroot. Например, для ProFTPd предусмотрена директива <Anonymous>, которая задает имя каталога, используемого в качестве корневого каталога поддерева chroot. Чтобы обеспечить выполнение сервера с использованием поддерева chroot, необходимо выяснить, какие опции управляют данным режимом работы, и правильно установить их значения.

Если сервер не вызывает chroot(), необходимо в первую очередь убедиться, что он способен работать в обычном окружении Linux, для чего следует запустить сервер за пределами поддерева chroot. Затем надо скопировать исполняемые файлы сервера и конфигурационные файлы в каталоги поддерева и удостовериться, что это не повлияло на работу сервера. После настройки среды поддерева вы можете запускать сервер с помощью утилиты chroot. Соответствующая команда имеет следующий вид:

```
chroot новый_корневой_каталог имя_сервера [опции_сервера]
```

Здесь под новым корневым каталогом подразумевается каталог, который выполняет роль корня поддерева chroot. Кроме того, при вызове команды задаются имя сервера, предназначенного для запуска, и его опции; путь к серверу определяется относительно корневого каталога поддерева. Например, если исполняемый файл сервера имеет имя /opt/chroot/bin/server, где /opt/chroot — корневой каталог поддерева, то вызов chroot будет выглядеть следующим образом:

```
# chroot /opt/chroot /bin/server
```

Если в обычных условиях сервер запускается с помощью сценария SysV или локального сценария запуска, вы должны модифицировать сценарий, включив в него команду chroot. Вы также можете запретить выполнение сценария и организовать запуск сервера другим способом. Если в системе предусмотрен запуск сервера посредством суперсервера, необходимо разместить в поддереве chroot не только сервер, предназначенный для запуска, но и супервсервер. Кроме того, надо изменить команду запуска суперсервера, реализовав его запуск посредством chroot. Если такое решение вас не устраивает, измените способ запуска сервера, например, запустите его с помощью сценария SysV или локального сценария.

Управление доступом к каталогам поддерева chroot

Поддерево chroot реализует одностороннюю защиту — программы, выполняющиеся в рамках поддерева, не имеют доступа к ресурсам за его пределами. Поэтому вы можете ограничить доступ и в другом направлении. Для этого надо указать в качестве владельца каталогов поддерева chroot пользователя root и установить соответствующие права доступа к этим подкаталогам, например задать значение 0640 (rw-r-----). Запускать сервер следует от имени пользователя, который принадлежит группе, специально созданной для этой цели. В результате сервер будет иметь право читать файлы, находящиеся в каталогах поддерева chroot, а из-за пределов поддерева к данным сможет обращаться только пользователь root. Если же при работе сервера возникает необходимость в записи файлов, следует предусмотреть это при установке прав доступа.

Запуск сервера BIND в рамках поддерева chroot

Ранее описывался процесс подготовки сервера к запуску в рамках поддерева chroot. Чтобы лучше понять изложенный выше материал, желательно рассмотреть запуск конкретного сервера в подобном режиме. В качестве примера выберем сервер имен BIND, работа которого обсуждалась в главе 18. При подготовке сервера к работе в пределах поддерева chroot будет в основном использоваться конфигурация, устанавливаемая по умолчанию. Процедура инсталляции данного сервера в различных версиях Linux имеет свои характерные особенности; для данного примера выберем версию Debian 2.2.

В данном разделе рассматривается запуск сервера BIND с использованием программы chroot. В качестве примера сервера, вызывающего функцию chroot() самостоятельно, можно привести сервер FTP.

Прежде всего вам необходимо инсталлировать стандартный пакет BIND. Поскольку сервер инсталлируется в системе Debian, для его установки можно использовать программу apt-get.

```
# apt-get install bind
```

В процессе выполнения сценарий инсталляции спрашивает, следует ли добавить адрес локального сервера имен в файл **/etc/resolv.conf**. На этот вопрос я даю положительный ответ, но для демонстрации работы сервера в рамках поддерева **enroot** это не имеет значения. По окончании установки система **Debian** запускает сервер имен. Проверить, работает ли сервер, вы можете с помощью следующих двух команд:

```
f ps aux I grep named
root      7656  0.0  1.5  2184 1492 ?      S    13:29   0:00 \
/usr/sbin/named
# host awl.com localhost
awl.com                    A          165.193.123.224
```

Вторая команда позволяет убедиться в том, что сервер BIND установлен и работает: она выводит IP-адрес узла **awl.com**, причем для преобразования имени используется сервер на компьютере **localhost**. Имя **awl.com** вы можете заменить любым другим именем узла, расположенного в Internet, а вместо **localhost** можно указать IP-адрес или имя вашего компьютера. Если система сообщит о том, что команда не найдена (command not found), вам надо установить пакет **dnsutils**, содержащий программу **host**. (В других версиях Linux пакет подобного назначения может называться иначе, например **bind-utils**.)

Убедившись, что сервер работает, завершите его выполнение с помощью команды

```
f /etc/init.d/bind stop
```

Затем вам надо создать поддерево **chroot** и скопировать в него файлы BIND.

```
# mkdir -p /opt/chroot/usr/sbin /opt/chroot/var/cache/bind
# mkdir /opt/chroot/lib /opt/chroot/etc
# cp /usr/sbin/named /opt/chroot/usr/sbin
# cp -rp /etc/bind/ /opt/chroot/etc
```

Данная процедура подготавливает BIND для выполнения с помощью команды **chroot**. Такой подход используется лишь для демонстрации действия данной команды. В случае необходимости сервер BIND может самостоятельно вызывать функцию **chroot()**, поэтому выполнение сервера имен в рамках поддерева **chroot** можно организовать несколько проще. Однако при этом все равно придется создать поддерево и поместить в него конфигурационные файлы. Отпадает необходимость лишь в копировании исполняемых файлов сервера.

В результате выполнения приведенных выше команд создается основа поддерева и в соответствующий каталог помещаются исполняемые коды сервера имен и конфигурационные файлы. При подготовке сервера к выполнению важно получить информацию о библиотеках, необходимых для его выполнения. Получив сведения о библиотеках с помощью команды **ldd**, следует скопировать соответствующие файлы в каталог поддерева **chroot**.

```
f ldd /usr/sbin/named
        libc.so.6 => /lib/libc.so.6 (0x40017000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
# cp /lib/libc.so.6 /lib/ld-linux.so.2 /opt/chroot/lib
```

На этом этапе можно снова проверить функционирование сервера.

```
# chroot /opt/chroot /usr/sbin/named
# host awl.com localhost
awl.com                    A        165.193.123.224
```

Если сервер не работает, убедитесь в том, что в системе выполняется только один экземпляр named, и проверьте, все ли файлы вы скопировали в каталог поддерева chroot. Обеспечив нормальную работу сервера, измените сценарий запуска BIND (в системе Debian это /etc/init.d/bind) так, чтобы сервер запускался посредством команды chroot. Безусловно, вы можете запретить выполнение сценария SysV и запустить сервер имен другим способом. Многие сценарии SysV используют вспомогательные программы (в Debian это start-stop-daemon и ndc). Данные программы могут создавать файлы в каталоге /var/run, поэтому вам надо создать в поддереве chroot нужные каталоги и скопировать файлы программ.

```
# mkdir -p /opt/chroot/sbin /opt/chroot/var/run
# cp /usr/sbin/ndc /opt/chroot/usr/sbin
# cp /sbin/start-stop-daemon /opt/chroot/sbin
```

При редактировании сценария запуска SysV перед каждым вхождением start-stop-daemon и ndc надо добавить последовательность символов chroot /opt/chroot. Однако на этом работа не заканчивается, поскольку start-stop-daemon обращается к файловой системе /proc, которая не доступна из поддерева chroot. Чтобы обеспечить доступ к ней, необходимо внести изменения в файл /etc/fstab — скопировать строку, содержащую /proc, и изменить ее на /opt/chroot/proc. Затем вы должны вызвать команду mount -a, чтобы смонтировать /proc в поддереве chroot.

ВНИМАНИЕ Поскольку файловая система /proc предоставляет контроль над компьютером, дублировать ее нежелательно. Лучше отредактировать сценарий запуска SysV так, чтобы он не использовал start-stop-daemon, либо отказаться от сценария SysV и организовать запуск сервера другим способом.

Выполнив все описанные выше действия, вы можете запустить сервер с помощью сценария SysV и проверить его работу.

```
# /etc/init.d/bind start
# host awl.com localhost
awl.com                    A        165.193.123.224
```

Если вы хотите удостовериться в том, что сервер выполняется в среде поддерева chroot, вам надо удалить исполняемый файл сервера из каталога /usr/sbin и конфигурационные файлы из каталога /etc/bind, а потом перезапустить сервер. Если сервер работает, то выполняться он может только в рамках поддерева chroot.

Вместо того чтобы запускать сервер BIND посредством утилиты chroot, вы можете использовать опцию -t программы named, которая разрешает вызов функции chroot() сервером имен. Соответствующая команда имеет следующий вид:

```
# /usr/sbin/named -t /opt/chroot
```

Данный подход намного проще описанного выше, так как при этом вам придется копировать в каталоги поддерева chroot гораздо меньше файлов, в частности, вы можете оставить программу named и библиотеки в тех каталогах, в которых они были записаны при инсталляции. Скопировать конфигурационные файлы необходимо, поскольку сервер

имен читает их уже после вызова chroot(). При использовании опции -t упрощается подготовка сервера для запуска посредством сценария SysV, так как при этом нет необходимости дублировать файловую систему /proc.

Детали подготовки сервера к выполнению в рамках поддерева chroot зависят от типа сервера и версии Linux, однако общий ход процедуры остается прежним. Возможно, вам придется немного модифицировать среду поддерева chroot, например, изменить права доступа к каталогам или настроить сервер для запуска от имени пользователя, отличного от root.

Поддержка среды chroot

Поддерево chroot представляет собой чрезвычайно полезный инструмент, однако требует выполнения определенных действий по поддержке. Ниже перечислены вопросы, которым администратор должен уделять внимание при поддержке поддерева chroot.

- **Ротация.** Во всех версиях Linux реализован механизм ротации файлов протоколов. Если сервер записывает файлы протоколов в каталог поддерева chroot, необходимо настроить средства ротации для работы с файлами, находящимися в этом каталоге. В качестве альтернативного решения при вызове mount можно указать опцию --bind, при этом файл, предназначенный для хранения файлов протоколов, станет доступным из поддерева chroot. Однако такой подход можно применять только в тех системах в которых используется версия ядра не ниже 2.4.x. Если вы не уделите внимания файлам протоколов, они будут неограниченно расти и в конце концов займут все доступное дисковое пространство.

- **Обновление программ.** При обновлении программного обеспечения дополнительные файлы приходится копировать в каталоги поддерева chroot. Если вы не сделаете этого, на компьютере будет выполняться старый вариант сервера, т. е. в результате установки дополнений ошибки в программе не будут устранены. Необходимо также следить за изменениями в сценариях запуска, иначе может произойти так, что сервер будет выполняться за пределами поддерева chroot.

- **Обеспечение доступа к файлам.** Если ваш сервер использует файлы с данными, вы должны разместить их в поддереве chroot. Обычно при копировании таких файлов не возникает проблем. Вам необходимо лишь следить за тем, чтобы права доступа к файлам были установлены в соответствии с используемой вами схемой защиты.

- **Использование новых программ поддержки.** В некоторых случаях может возникнуть необходимость разместить в каталогах поддерева chroot новые программы поддержки. Так, например, если Web-сервер обеспечивает работу сценариев CGI, ему может потребоваться интерпретатор нового языка. Наряду с размещением новых программ поддержки в поддереве chroot необходимо удалять файлы, не используемые сервером. Этим вы устраните неоправданный риск при работе сервера.

Для решения описанных выше вопросов не требуется много времени. Необходимые действия в основном сводятся к однократной установке требуемых значений опций, в остальных случаях приходится обновлять сервер или изменять его конфигурацию.

Резюме

Ограничение сферы действия сервера поддеревом chroot позволяет уменьшить риск, связанный с работой этого сервера. Такой подход приемлем в основном для серверов, обращающихся в процессе выполнения к ограниченному набору файлов. Для того чтобы обеспечить функционирование сервера в рамках поддерева, надо продублировать в поддереве chroot некоторые каталоги и файлы системы Linux. В ряде случаев приходится копировать в каталог поддерева и исполняемый файл сервера. Некоторые серверы самостоятельно вызывают функцию chroot(), а для запуска других приходится применять программу chroot. Независимо от способа запуска, сервер не может обращаться за пределы поддерева. Корневой каталог поддерева chroot он воспринимает как корневой каталог всей файловой системы. При подготовке сервера к выполнению в рамках поддерева chroot следует выяснить, какие файлы нужны для его выполнения и с какими вспомогательными программами он должен взаимодействовать в работе. Соответствующие файлы надо скопировать в каталоги поддерева chroot. Чтобы сервер мог выполняться в поддереве chroot, необходимо также модифицировать стандартную процедуру его запуска. Среда chroot требует поддержки, но для выполнения соответствующих действий администратору приходится затрачивать не слишком много времени.

Глава 24

Расширенные средства маршрутизации

Несмотря на то что Linux считается операционной системой общего назначения, количество специальных применений Linux постоянно увеличивается. Известны даже случаи использования данной системы в устройствах PDA и видеомагнитофонах. Одним из специальных (хотя и не столь экзотических) применений Linux является *маршрутизация*. Маршрутизаторы не относятся к числу общеизвестных устройств, многие пользователи даже не знают об их существовании, однако они жизненно необходимы для нормальной работы Internet. К маршрутизаторам относятся как простые и недорогие устройства, предназначенные для подключения небольшой сети к Internet, так и высокопроизводительные средства, обеспечивающие передачу пакетов, поступающих по высокоскоростным магистралям. Linux можно без труда настроить для работы в качестве низкоуровневого маршрутизатора. Если же в сети насчитывается несколько десятков компьютеров, необходимо применять расширенные средства маршрутизации. Эти средства позволяют использовать при доставке пакетов систему приоритетов и взаимодействовать с другими компьютерами.

Если вы собираетесь создавать NAT-маршрутизатор на базе Linux, вам следует ознакомиться с материалом, изложенным в следующей главе.

, Базовая конфигурация маршрутизатора на базе Linux устанавливается достаточно просто, но при использовании расширенных средств маршрутизации могут возникнуть проблемы. В данной главе приведены лишь общие сведения о способах маршрутизации и об инструментальных средствах, используемых для этого. Дополнительную информацию вы найдете в документации на конкретные инструменты. Кроме того, вопросы, рассматриваемые в данной главе, подробно изложены в книге Лебланка (LeBlanc) и др. *Linux Routing* (New Riders, 2002).

Использование расширенных средств маршрутизации

Приступая к чтению данной главы, следует иметь в виду, что в ней рассматриваются расширенные средства маршрутизации. Если трафик, связанный с обменом внутренней сети с Internet, невелик и если от маршрутизатора требуется поддержка лишь простых статических маршрутов, вам нет никакой необходимости использовать средства, описанные в данной главе. На компьютере, содержащем две сетевые карты, вы можете реализовать перенаправление пакетов с помощью следующей команды:

```
# echo "1" > /proc/sys/net/ipv4/ip_forward
```

При наличии нескольких сетей для организации доставки пакетов достаточно простой таблицы маршрутизации.

Для того чтобы обеспечить маршрутизацию на локальном компьютере с несколькими интерфейсами, достаточно лишь правильно заполнить таблицу маршрутизации, а компьютеры, к которым он непосредственно подключен по сети, должны лишь знать о том, что этот компьютер выполняет функции маршрутизатора. В качестве примера рассмотрим компьютер под управлением Linux, который используется для подключения небольшой сети к Internet посредством линии SDSL. Если компьютер не поддерживает NAT, маршрутизатор провайдера, к которому рассматриваемый компьютер подключен посредством сетевого интерфейса, должен знать, что этот компьютер является маршрутизатором для локальной сети. Если маршрутизатор провайдера не имеет таких сведений, он будет передавать по назначению пакеты, отправленные из локальной сети, но не сможет доставить пакеты, переданные в ответ. В большинстве случаев необходимо, чтобы администратор сети, к которой подключена ваша сеть, настроил свой маршрутизатор для взаимодействия с вашим. Кроме того, вам надо сконфигурировать узлы вашей сети так, чтобы они использовали компьютер под управлением Linux в качестве маршрутизатора.

В данной главе рассматриваются вопросы принятия решений о маршрутах пакетов на основании исходного адреса, адреса назначения и типа протокола. От того, насколько правильно приняты такие решения, зависит эффективность работы Internet. Так, например, целесообразно присвоить высокий приоритет пакетам, которые соответствуют интерактивным протоколам, за счет задержки менее важных данных. Конфигурация, позволяющая выполнить подобные действия, обычно устанавливается на выделенных маршрутизаторах, предназначенных для обработки больших объемов информации.

В этой главе также рассматриваются протоколы маршрутизации, которые позволяют организовать взаимодействие с другими маршрутизаторами. Маршрутизаторы, поддерживающие эти протоколы, позволяют динамически заполнять таблицы маршрутизации, обеспечивая наиболее быструю доставку пакетов. Благодаря использованию этих протоколов повышается производительность сети, однако они в основном применяются тогда, когда маршрутизатор подключен к Internet через несколько сетевых интерфейсов. Если же для соединения с Internet используется лишь один интерфейс, применять протоколы маршрутизации бессмысленно, так как они не будут оказывать влияния на содержимое таблицы маршрутизации.

Расширенные опции ядра

В ядре 2.4.x предусмотрены расширенные опции маршрутизации. Они располагаются в меню Networking Options. Многие из них являются **подопциями** опции IP: Advanced Router; чтобы активизировать подопции, надо активизировать саму опцию IP: Advanced Router. Расширенные опции маршрутизации позволяют задавать особенности маршрутизации пакетов: способы назначения приоритетов, обработку приоритетов, указанных в принятых пакетах, поддержку типов пакетов и т. д. Чтобы активизировать опцию, надо установить переключатель в положение Y (или M, если вы хотите, чтобы средства поддержки этой опции были реализованы в виде модуля). Многие опции требуют настройки посредством специальных утилит. Некоторые из них достаточно сложны, поэтому в данном разделе приведены лишь общие сведения о соответствующих инструментах.

В различных версиях ядра наборы опций могут различаться: некоторые из них разделяются на отдельные опции, другие объединяются в одну. В данном разделе описаны опции ядра 2.4.17. В других версиях ядра опции могут отличаться от описанных здесь.

Политика маршрутизации

Одна из опций, определяющих использование расширенных средств маршрутизации Linux, называется IP: Policy Routing. Она поддерживает следующие способы маршрутизации.

- Фильтрация на основе маркеров. Пакеты, передаваемые по сети, могут содержать специальные данные — *маркеры*. В случае необходимости можно организовать передачу пакетов, помеченных такими маркерами, по специальным маршрутам. Фильтрацией на базе маркеров управляет опция IP: Use Netfilter MARK Value as Routing Key. Если вы собираетесь активизировать эту опцию, надо установить также опцию Packet Filtering, находящуюся в том же меню.

- Быстрое **NAT-преобразование**. Средства NAT позволяют "спрятать" компьютеры сети так, чтобы они были невидимы для остальных узлов Internet. При этом всю сеть представляет один компьютер, а единственный IP-адрес, выделенный для этого компьютера, используется для организации работы всей сети. Если вы хотите, чтобы ваша система функционировала как **NAT-маршрутизатор**, вы можете установить опцию IP: Fast NAT, однако это не обязательное условие NAT-маршрутизации. (Подробно средства NAT рассматриваются в главе 25.)

Компоненты, включаемые посредством описанных выше опций, используются при работе пакета `iproute2`, который взаимодействует с ядром и поддерживает расширенные средства маршрутизации. Этот пакет будет рассматриваться далее в настоящей главе.

Тип сервиса

В IP-пакетах предусмотрено специальное поле под названием TOS (Type-of-Service — тип сервиса). Это поле позволяет компонентам сети определять, какие из пакетов требуют специальной обработки. В результате подобной обработки для некоторых клиентов и серверов реализуются более быстрые и надежные соединения по сравнению с другими.

Для того чтобы разрешить обработку этого поля, надо активизировать опцию ядра IP: Use TOS Value as Routing Key.

Данная опция также используется при работе пакета `iproute2`. В поле TOS содержится числовое значение. Большинство маршрутизаторов игнорирует поле TOS, поэтому чаще всего его содержимое не влияет на качество соединения.

Передача пакетов по различным маршрутам

Большинство маршрутизаторов проверяет адрес назначения приходящего пакета на соответствие правилам, содержащимся в таблице маршрутизации. Например, в таблице может быть указано, что пакеты, направленные в сеть 10.201.0.0/16, должны передаваться через интерфейс `eth1`. Не исключено, что адрес пакета будет соответствовать двум правилам; это не приводит к возникновению конфликта. Предположим, что, помимо приведенного выше правила, в таблице указано, что пакеты, адресованные в сеть 10.201.34.0/24, должны передаваться через интерфейс pppO. Если некоторый пакет отвечает обоим условиям, к нему будет применено второе правило, как более конкретное. Если же в таблицу будет включено еще одно правило для сети 10.201.0.0/16, то пакет будет обработан посредством того правила, которое первым встретится маршрутизатору.

При активизации опции ядра IP: Equal Cost Multipath система будет вести себя следующим образом. Если пакет соответствует нескольким правилам маршрутизации, то правило, применяемое для обработки пакета, будет выбрано случайным образом. Такой алгоритм можно рассматривать как примитивный способ распределения нагрузки между различными соединениями. Более конкретному правилу отдается предпочтение перед более общим.

Протоколирование работы маршрутизатора

Опция IP: Verbose Route Monitoring управляет выводом сведений о маршрутизации в файл протокола. В обычных условиях ядро не протоколирует ход маршрутизации пакетов. Если данная опция установлена, регистрируются сведения о пакетах, корректность которых вызывает сомнения.

Протоколирование работы маршрутизатора предоставляет администратору информацию, которую он может использовать для того, чтобы принять меры по защите системы. Однако процедура регистрации требует дополнительных ресурсов, и если нагрузка на маршрутизатор велика, то его производительность может уменьшиться. Кроме того, при выполнении протоколирования маршрутизатор становится более уязвимым для атак с целью вывода серверов из строя. (При организации такой атаки злоумышленник передает в сеть большое количество пакетов, намереваясь создать такую нагрузку на сервер или маршрутизатор, с которой тот не сможет справиться, либо вызвать переполнение диска.)

Использование больших таблиц маршрутизации

Обычно ядро Linux настраивается для работы с таблицами маршрутизации, содержащими не больше 64 записей. Если этих записей не хватает для того, чтобы задать требуемую конфигурацию маршрутизатора, вам надо активизировать опцию IP: Large Routing Tables. Эта опция позволяет использовать таблицы маршрутизации большего размера.

Поддержка группового вещания

Как правило, в Internet-взаимодействии участвуют два компьютера, например, клиент обращается к Web-серверу, а Web-сервер возвращает клиенту ответ на его запрос. При этом пакет, передаваемый по сети, предназначен только одному получателю. Существует также широковещательная передача данных, когда передаваемый пакет адресован всем узлам локальной сети. Для широковещательной передачи в пределах локальной сети используется адрес получателя 255.255.255.255, который определяет все компьютеры, подключенные к этой сети. Если широковещательный пакет направляется в другую сеть, адрес получателя формируется путем замены части адреса, соответствующего узлу сети, на число, состоящее из единиц. Например, широковещательный адрес для сети 192.168.34.0/24 будет иметь вид 192.168.34.255. Широковещательный адрес используют клиенты DHCP для обращения к серверу DHCP. Кроме того, широковещательная передача применяется при работе некоторых других протоколов.

При групповом вещании пакеты адресуются одновременно нескольким получателям (но не всем узлам сети). Такой способ передачи информации применяется для трансляции аудио- и видеоданных. Для организации группового вещания предназначена система Multicast Backbone (MBONE; `http://www.cs.columbia.edu/~hgs/internet/mbone-faq.html`). Данная система реализует групповое вещание в пределах Internet. Существуют также средства группового вещания с ограниченными возможностями; такой тип группового вещания называется *локальными связями* (link-local). Эти средства не нашли широкого распространения, но иногда используются при взаимодействии маршрутизаторов.

Если вы хотите, чтобы ваш маршрутизатор поддерживал групповое вещание, установите опцию IP: Multicast Routing. Кроме того, для управления групповым вещанием также предусмотрены две подопции данной опции: IP: PIM-SM Version 1 Support и IP: PIM-SM Version 2 Support. Они управляют передачей сообщений группового вещания по сетям с ограниченной пропускной способностью.

Помимо опций ядра, для поддержки группового вещания должно использоваться специализированное программное обеспечение, например `mrouted`. Этот инструмент предназначен для настройки базовых средств группового вещания Linux. Если данная программа не входит в состав дистрибутивного пакета, вы можете получить ее, обратившись по адресу `ftp://ftp.rge.com/pub/communications/ipmulti/beta-test/`; подробная информация о ней приведена в документе `http://jukie.net/~bart/multicast/Linux-Mrouted-MiniHOWTO.html`. Если у вас установлена опция IP: PIM-SM Version 2 Support, вам понадобится программа pimd (`http://netweb.usc.edu/pim/pimd/`).

Качество сервиса

В большинстве случаев средства маршрутизации Linux обрабатывают пакеты по принципу "первый пришел/первым обработан" (first-come/first-served). Такая процедура дает хорошие результаты в тех случаях, когда пропускная способность линий достаточна для передачи всех пакетов с минимальной задержкой и когда нет необходимости предоставлять некоторым пакетам более высокий приоритет по сравнению с остальными. Если нагрузка на маршрутизатор велика, приходится использовать другие способы обработки пакетов, которые позволяют уменьшить поток данных на некоторые узлы или гарантировать своевременную доставку информации определенным пользователям или приложе-

ниям. Для установки подобных режимов маршрутизации в системе Linux предусмотрен ряд опций. Они объединены в подменю QoS and/or Fair Queueing, которое вызывается с помощью одноименного пункта меню Networking Options.

Активизация опций меню QoS and/or Fair Queueing сама по себе не приведет к изменению работы системы. Подобно другим опциям, которые управляют средствами расширенной маршрутизации, эти опции предполагают использование пакета **iproute2**. Если вы не уверены в том, что данные средства понадобятся вам, скомпилируйте компоненты ядра, предназначенные для их поддержки, в виде модулей. В результате соответствующие компоненты не будут занимать ресурсы компьютера, но вы всегда сможете воспользоваться ими.

ВНИМАНИЕ! Перед компиляцией ядра внимательно ознакомьтесь с документацией. В ядре 2.4.17 опция, реализующая алгоритм CSZ, работает некорректно. Активизация этой опции может привести к ненадежной работе маршрутизатора.

Использование iproute2

Пакет **iproute2** входит в состав многих дистрибутивных пакетов Linux. Часто он поставляется под именем **iproute**. Вы можете скопировать данный пакет с FTP-узла, предназначенного для его поддержки, обратившись по адресу **ftp://ftp.inr.ac.ru/ip-routing/**. Пакет **iproute2** содержит несколько программ, две из которых (ip и tc) рассматриваются в данной главе.

Использование ip

Программа ip предназначена для управления таблицами маршрутизации, в частности, правилами, определенными в них. Выполнение данной программы зависит от значений некоторых подопций опции IP: Advanced Router. Программа ip вызывается следующим образом:

ip *команда* [list | add I del] *селектор действие*

В утилите ip предусмотрено несколько команд. Наиболее важная из них — команда **rule**. Она позволяет добавлять (add), удалять (del) правила маршрутизации или отображать информацию о существующих правилах (list). Правила определяются с помощью селектора, который имеет структуру

[from *адрес]* [to адрес] [tos *тип_сервиса*]
[dev *имя_устройства*] [pref *число*]

Элементы **from** и to определяют IP-адреса, а элемент tos задает тип сервиса (тип сервиса представляет собой число, например 4). Элемент dev определяет имя сетевого устройства (например, ethO), а **pref** задает номер предпочтения. Набор элементов сообщает Linux о том, как идентифицируются пакеты, к которым должно быть применено данное правило. Действие, указываемое в команде, задается в следующем формате:

[table *идентификатор_таблицы*] [nat адрес]
[prohibit I reject I unreachable]

Идентификатор таблицы — это число, которое идентифицирует таблицу маршрутизации, элемент nat позволяет задать для пакета новый адрес источника, а **prohibit**,

reject и unreachable — это коды, определяющие различные варианты отказа от пакета.

Пример реальной команды ip приведен ниже.

```
# ip rule add from 172.20.24.128 dev ethO table 2
```

Правило, определяемое с помощью данной команды, указывает системе на то, что для обработки трафика с адреса 172.20.24.128 через ethO должна использоваться таблица маршрутизации 2. У вас, вероятно, возникнет вопрос, что значит таблица маршрутизации 2? В системе Linux для заполнения таблицы маршрутизации используется команда route. Расширенные средства маршрутизации позволяют работать с несколькими таблицами, создаваемыми посредством команды ip route. При обработке различных типов трафика можно быстро переключаться между разными таблицами. Приведенная выше команда сложнее обычной команды route, но она предоставляет возможности, которые не может обеспечить route. Вы можете использовать ip route так же, как и route, единственное отличие состоит в том, что вам необходимо задать номер таблицы. Например, для добавления маршрута в таблицу 2 можно использовать следующую команду:

```
ip route add 10.201.0.0/16 dev eth1 table 2
```

Если не принимать во внимание имя программы ip и элемент table 2, то данная команда эквивалентна команде route. Она сообщает системе о том, что все данные, адресованные в сеть 10.201.0.0/16, должны передаваться через интерфейс eth1.

Использование tc

Утилита tc использует средства ядра, которые активизируются посредством опций меню QoS and/or Fair Queueing. Данная программа управляет исходящим трафиком, в частности, не позволяет одному типу трафика монополизировать пропускную способность линии связи. В качестве примера рассмотрим следующую ситуацию. Предположим, что в организации имеются две подсети; каждая из них обслуживает офис, в котором работают около десяти пользователей. Если пользователи одной подсети запускают программы, генерирующие большой объем исходящих данных, производительность сетевых средств пользователей, работающих в другом офисе, может оказаться недопустимо низкой. Программа tc позволяет распределить ресурсы таким образом, что каждой из подсетей будет гарантированно выделяться часть пропускной способности линии.

Программа tc вызывается следующим образом:

```
tc [опции] объект команда
```

Ниже описаны элементы, передаваемые программе tc.

- *опции*. При вызове tc могут быть заданы опции -statistics (или -s), -details (или -d) и -raw (или -r).
- *объект*. Данный параметр может принимать значение qdisc, class или filter. Значение qdisc определяет *порядок обслуживания,* или *дисциплину очереди,* class задает набор пакетов, принадлежащих той или иной категории либо классу (в данном примере признаком принадлежности к классу является принадлежность к одной из подсетей), а filter генерирует правило фильтрации.
- *команда*. Команда — это набор параметров, которые определяют, какие действия программа tc должна выполнить с объектом. Состав команды зависит от объекта.

С помощью tc вы можете сгенерировать набор правил, описывающих сети, подключенные к компьютеру, и определяющих принцип выделения пропускной способности линии для каждой из этих сетей. Предположим, что вы хотите распределить пропускную способность линии, равную 100 Мбод, поровну между двумя подсетями, обслуживающими офисы. Предположим также, что ваш компьютер подключен к Internet посредством устройства ethO, а данные в обе подсети передаются через устройство eth1; одна из подсетей имеет IP-адрес 192.168.1.0/24, а другая — 192.168.2.0/24. Процесс настройки следует начать с определения порядка обслуживания очереди для ethl.

```
t tc qdisc add dev ethl root handle 10: cbq bandwidth 100Mbit \
avpkt 1000
```

Данную команду можно условно разделить на несколько частей.

- add dev ethl. Данный компонент команды сообщает системе о том, что вы добавляете дисциплину очереди для eth1.
- root. В некоторых случаях дисциплины могут составлять деревья. Этот параметр указывает на создание корневого узла нового дерева.
- handle 10. Этот компонент команды определяет метку (handle) для дисциплины.
- cbq. Вам необходимо сообщить системе, какой метод организации очереди должен быть использован. Метод CBQ (Class-Based-Queueing — очередь на базе классов) применяется чаще всего. Данный параметр отражается в имени одной из опций ядра в меню QoS and/or Fair Queueing.
- bandwidth 100Mbit. С помощью данного компонента вы сообщаете системе о пропускной способности линии. Если различные интерфейсы маршрутизатора подключены через линии с разной пропускной способностью, задается наименьшее значение.
- avpkt 1000. По сети могут передаваться пакеты различного размера, но, для того, чтобы планировать использование линии, система должна иметь хотя бы приблизительное представление о том, какими могут быть размеры пакетов. Конкретное значение данного параметра может отличаться от указанного здесь.

Теперь надо определить классы для сети и для каждой из подсетей. Для этого используются команды наподобие следующей:

```
# tc class add dev ethl parent 10:0 classid 10:1 cbq \
bandwidth lOOMbit rate lOOMbit allot 1514 weight 10Mbit \
prio 8 maxburst 20 avpkt 1000
```

Эта команда похожа на предыдущую, но она создает класс, определяющий одну из двух сетей. Обратите внимание на то, что данный класс задается для использования всей доступной пропускной способности. Впоследствии этот ресурс будет разделен между подсетями. В отличие от предыдущей команды, некоторые параметры изменены, кроме того, в состав этой команды входят дополнительные компоненты.

- class. Если в предыдущей команде был задан объект qdisc, то здесь присутствует объект class, определяющий класс.

- `parent 10:0`. Этот компонент команды задает корень дерева. К метке, определенной в предыдущей команде, добавляется значение 0.
- `classid 10:1`. Данный компонент задает идентификатор конкретного класса.
- `allot 1514`. С помощью этого параметра указывается значение MTU для сети (оно на несколько байт превышает реальное значение).
- `weight 1Mbit`. Данный параметр используется для настройки. Возможно, для конкретной сети необходимо специально подобрать его значение.
- `prio 8`. Этот компонент команды задает приоритет правила. Чем больше значение, тем выше приоритет.

Правила для подсетей задаются с помощью команд, подобных рассмотренной выше.

```
# tc class add dev eth1 parent 10:1 classid 10:100 cbq \
bandwidth 100Mbit rate 50Mbit allot 1514 weight 5Mbit \
prio 5 maxburst 20 avpkt 1000 bounded
# tc class add dev ethl parent 10:1 classid 10:200 cbq \
bandwidth lOOMbit rate 50Mbit allot 1514 weight 5Mbit \
prio 5 maxburst 20 avpkt 1000 bounded
```

Эти команды различаются только значением `classid`. Обе ссылаются на корневой класс, и каждая выделяет соответствующей подсети 50 Мбод пропускной способности линии. (Вы можете задать разные значения для каждой подсети, например 60 Мбод и 40 Мбод.) Параметр bounded указывает на то, что система не должна выделять классу часть пропускной способности линии, превышающую указанное значение. Часто такое ограничение бывает нежелательным, поскольку если из одной сети данные не передаются, другая сеть не сможет использовать остальную часть пропускной способности линии. Отказавшись от параметра bounded, вы обеспечите большую гибкость при работе через линии связи, в частности, предоставите офисам возможность "занимать" друг у друга пропускную способность линии. Если же обоим офисам потребуется передавать данные, этот ресурс будет распределен поровну.

Теперь необходимо связать дисциплину очереди с каждым из двух классов.

```
# tc qdisc add dev ethl parent 10:100 sfq quantum 1514b \
perturb 15
# tc qdisc add dev ethl parent 10:200 sfq quantum 1514b \
perturb 15
```

Данные команды аналогичны рассмотренной ранее команде, определяющей порядок обслуживания очереди. Они сообщают Linux о том, что для планирования трафика внутри подсети каждого офиса должна использоваться дисциплина SFQ (Stochastic Fairness Queueing — стохастическая организация очереди, обеспечивающая равный доступ). Эта дисциплина популярна, так как для ее реализации не требуется много ресурсов процессора. Если понадобится, можете задать другую дисциплину.

Команды, которые мы уже рассмотрели, не предоставляли ядру информацию, позволяющую разделить трафик, соответствующий различным подсетям (192.168.1.0/24 и 192.168.2.0/24). Поэтому необходимо выполнить следующие команды:

```
# tc filter add dev ethl parent 10:0 protocol ip prio 100 u32 \
match ip dst 192.168.1.0/24 flowid 10:100
# tc filter add dev ethl parent 10:0 protocol ip prio 100 u32 \
match ip dst 192.168.2.0/24 flowid 10:200
```

В отличие от предыдущих, *в* этих командах указан объект `filter`. Данные команды задают правила, которые связывают трафик подсети с соответствующим классом. Обоим правилам назначены одинаковые приоритеты и задан алгоритм u32, работающий с блоками IP-адресов.

Созданные правила управляют потоком данных из Internet в локальные сети. При желании вы можете создать аналогичный набор правил, действующих в противоположном направлении. Эти правила почти совпадают с предыдущими, но вместо внутреннего интерфейса `ethl` они должны ссылаться на внешний интерфейс ethO, и в двух последних командах `filter` вместо параметра dst должен быть указан параметр `src`.

Использование протоколов маршрутизации

Главная задача маршрутизатора — определить способ передачи пакетов. Предположим, например, что от маршрутизатора к целевому узлу ведут два маршрута. С помощью специальных инструментов, например программы ip, входящей в состав пакета `iproute2`, вы можете сообщить маршрутизатору, какой из путей следует выбрать. Маршрутизатор, настроенный подобным образом, будет применять для маршрутизации пакетов одни и те же правила до тех пор, пока вы не измените их с помощью ip. Подобное поведение маршрутизатора допустимо лишь в простых статических средах. В реальных условиях сетевое окружение постоянно изменяется: вводятся в строй новые линии связи, а существующие линии внезапно прекращают работу. Подобные изменения могут происходить далеко от вашей сети, и сведения о них не всегда будут поступать к вам. Иногда может возникнуть необходимость сообщить другим маршрутизаторам об изменениях в вашей сети, например, о появлении новой подсети с определенным IP-адресом. Для решения подобных задач предназначены протоколы маршрутизации, рассмотрению которых посвящен данный раздел.

Принцип действия протоколов маршрутизации

Ранее в этой главе был описан процесс настройки маршрутизатора, реализованного в системе Linux, для обработки пакетов в зависимости от адреса назначения, содержимого и других характеристик пакета. Протоколы маршрутизации предоставляют возможность учитывать состояние сетевой среды. Они позволяют получить информацию о том, достижима ли требуемая сеть и какова стоимость передачи пакета в эту сеть. Здесь понятие стоимости не связано с деньгами. Под стоимостью обычно понимают число узлов, которые должен посетить пакет, прежде чем он попадет на целевой узел. В роли стоимости также могут выступать другие характеристики сети, определяющие ее производительность. Стоимость передачи пакета по сети называется *метрикой*. Если маршрутизатор соединен с другими сетями посредством двух сетевых соединений, стоимость доставки пакета на целевой узел, или метрика пути к целевому узлу, зависит от того, посредством какого соединения будет передан этот пакет. Рассмотрим в качестве примера сетевую среду, условно показанную на рис. 24.1. На этом рисунке изображено пять сетей, принад-

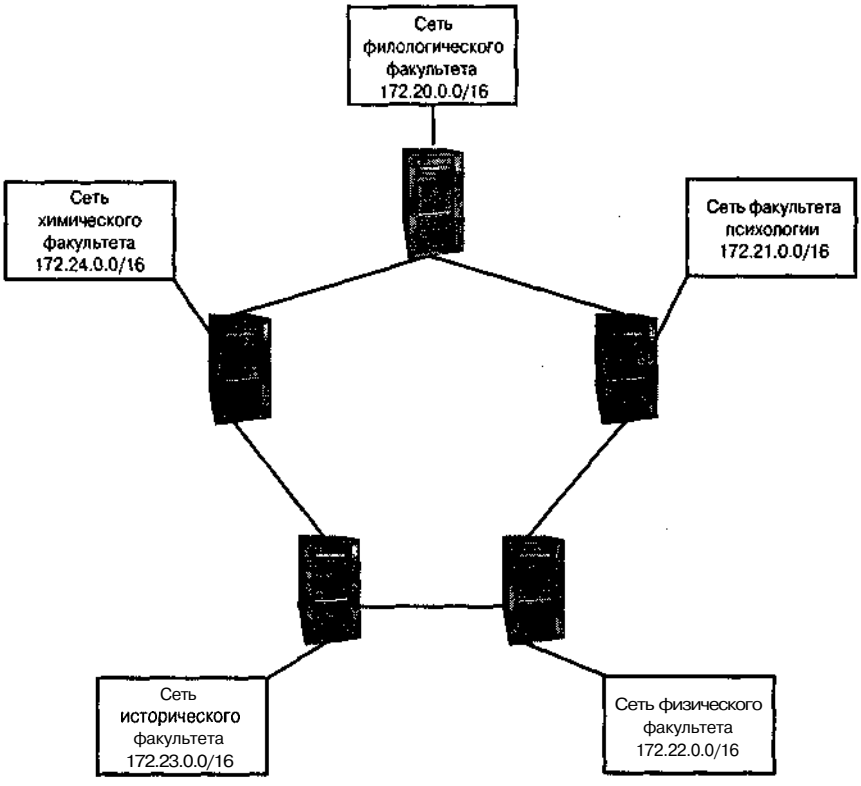


Рис. 24.1. Протоколы маршрутизации позволяют маршрутизаторам обмениваться информацией и определять наиболее короткий маршрут для передачи пакета

лежащих пяти факультетам университета. В каждой сети работает свой маршрутизатор, который соединен с двумя другими маршрутизаторами.

В данном примере описывается несколько сетей, соединенных между собой; такая структура называется internet (со строчной буквы). Аналогичные принципы используются и при выполнении маршрутизации во всей Internet (с прописной буквы).

При неоправданном посещении пакетом очередного узла увеличивается время его доставки в целевую сеть. Например, при передаче из сети филологического факультета в сеть физического факультета пакет посетит как минимум три маршрутизатора. Заметьте, что из сети филологического факультета в сеть физического факультета ведут два маршрута. Один из них проходит через маршрутизатор факультета психологии, а другой — через маршрутизаторы химического и исторического факультетов. Путь в направлении против часовой стрелки оказывается более длинным. Хорошо, если бы маршрутизатор филологического факультета имел информацию об обоих маршрутах. Если маршрутизатор факультета психологии выйдет из строя, пакеты можно будет передавать по альтернативному маршруту.

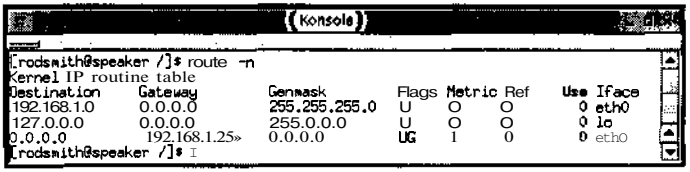


Рис. 24.2. Значение в поле Metric определяет стоимость передачи пакета по данному маршруту

Для решения данной задачи используется понятие метрики. Значение метрики, определяющее стоимость передачи пакета по сети, содержится в стандартной таблице маршрутизации Linux, которая заполняется с помощью команды route. На рис. 24.2 показано содержимое простой таблицы маршрутизации (на реальных маршрутизаторах размер таблицы гораздо больше, но ее формат остается неизменным). Обратите внимание на столбец под названием Metric. В нем указано, что число маршрутизаторов, которые пакет посетит по пути следования в сети 127.0.0.0/8 (localhost) и 192.168.1.0/24 (локальная сеть), равно нулю, а оставшийся маршрут проходит через один маршрутизатор. Данный пример чрезвычайно прост. Для маршрутизаторов, показанных на рис. 24.1, в таблице маршрутизации содержалась бы информация о разных маршрутах, отличающихся значениями в поле Metric. К сожалению, стандартные средства маршрутизации системы Linux игнорируют значение метрики; для того, чтобы метрика учитывалась, необходимо использовать один из протоколов маршрутизации, рассматриваемых в данной главе.

Протоколы маршрутизации позволяют маршрутизаторам обмениваться информацией о маршрутах, в том числе передавать значения метрики. Если все маршрутизаторы, показанные на рис. 24.1, будут поддерживать протокол маршрутизации, они сообщат друг другу сведения об имеющихся маршрутах и каждый из них построит таблицу маршрутизации, в которой будут указаны реальные значения метрики. Если один из маршрутизаторов выйдет из строя, остальные маршрутизаторы получат информацию об этом и перенаправят трафик в обход неисправного участка сети.

Протоколы маршрутизации используют следующие алгоритмы.

- **Дистанционно-векторный алгоритм.** Этот алгоритм отслеживает число маршрутизаторов, находящихся между текущим маршрутизатором и узлом назначения. При передаче пакета в некоторую сеть выбирается такой путь, на котором число маршрутизаторов будет минимальным. Данный алгоритм используется при работе протокола RIP (Routing Information Protocol — протокол маршрутной информации).

- **Алгоритм маршрутизации по состоянию канала.** Данный алгоритм связывает информацию о стоимости передачи пакета с каждым соединением. Маршрутизатор, использующий такой алгоритм, выбирает путь к целевой сети, для которого значение стоимости минимально. Стоимость не обязательно должна быть равна числу маршрутизаторов, при ее определении могут также учитываться различия в быстродействии сетевых соединений. Данный алгоритм используется при работе протокола OSPF (Open Shortest Path First — первоочередное открытие кратчайших маршрутов).

Использование routed

В системе UNIX традиционно используется протокол RIP. В Linux он реализуется демоном routed, входящим в состав одноименного пакета. Маршрутизаторы, поддерживающие RIP, обмениваются адресами сетей (например, 172.22.0.0) и связанными с ними метриками (в качестве метрики принимается число маршрутизаторов между маршрутизатором, который должен отправить пакет, и целевой сетью). Значения метрики могут лежать в пределах от 0 до 15. Если на пути к целевой сети лежит больше 15 маршрутизаторов, длина маршрута считается бесконечной и информация об этом маршруте удаляется из таблицы. При работе протокола RIP используется дистанционно-векторный алгоритм, а значение метрики оценивается очень грубо. Протокол RIP в основном применяется в небольших и средних сетях; для управления передачей данных по магистралям Internet он не используется.

Когда маршрутизатор получает информацию от другого маршрутизатора, он либо добавляет запись о маршруте в таблицу, либо заменяет существующую запись с более высоким значением метрики, либо удаляет маршрут, если полученное значение метрики, увеличенное на единицу, превышает 15.

При использовании программы routed в системе Linux обычно не возникает проблем. Для ее запуска применяются средства, рассмотренные в главе 4. Работой сервера управляет конфигурационный файл /etc/gateways, в котором содержится список начальных маршрутов. Пример записи в файле /etc/gateways приведен ниже.

```
net 0.0.0.0 gateway 172.22.7.1 metric 1 active
```

В данном примере определяется маршрут по умолчанию (net 0.0.0.0), для которого задан шлюз 172.22.7.1. Метрика маршрута равна 1. Ключевое слово active указывает на то, что этот маршрут может быть обновлен. Если вы хотите, чтобы маршрут сохранялся в таблице в неизменном виде, надо заменить ключевое слово active на passive. Для работы routed можно использовать файл /etc/gateways, поставляемый в составе пакета. В процессе работы демон routed может, передавая широковещательные запросы, находить другие маршрутизаторы, использующие протокол RIP. После обнаружения маршрутизатора с ним начинается обмен информацией о маршрутах.

Использование GateD

Несмотря на то что протокол RIP традиционно используется в системе UNIX, область его применения ограничена. Одно из ограничений связано с тем, что пакет не может быть передан по маршруту, насчитывающему больше 15 маршрутизаторов; это не позволяет использовать данный протокол в больших сетях. Еще одна проблема связана с медленной сходимостью алгоритма. При изменении структуры сети для достижения стабильного состояния таблицы маршрутизации может потребоваться несколько минут. И наконец, RIP не поддерживает маски подсетей, поэтому он может использоваться только для сетей, соответствующих классам A, B и C. Если, например, сеть класса C разбита на несколько подсетей, маршрутизатор, поддерживающий RIP, передает адрес подсети как адрес всей сети класса C. В результате возникают проблемы при обмене данными между различными подсетями.

В версии 2 протокола RIP (RIPv2) была добавлена поддержка маски подсети. Для хранения данных о маске использовалось поле, которое в исходном варианте RIP было зарезервировано. Протокол RIPv2 реализован в программе GateD (http://www.gated.

net). Работой GateD управляет конфигурационный файл /etc/gated.conf. Для изменения конфигурации GateD используется утилита gdc, которая поставляется в составе того же пакета. Настройка GateD не требует много усилий. В процессе выполнения программа взаимодействует с другими маршрутизаторами, поддерживающими протокол RIP или RIPv2, и модифицирует содержимое таблицы маршрутизации. Подобно другим демонам, GateD запускается с помощью сценария SysV либо локального сценария запуска.

Помимо RIP и RIPv2, GateD также поддерживает протокол маршрутизации OSPF. Другие средства маршрутизации, например Zebra, также поддерживают несколько протоколов.

Использование Zebra

Наряду с рассмотренными выше программами маршрутизации в Linux применяется инструмент Zebra, который представляет собой пакет, состоящий из нескольких доменов и поддерживающий следующие протоколы.

- RIP. Zebra поддерживает протоколы RIP и RIPv2, а также версию RIP для IPv6, которая называется RIPng. Для взаимодействия по протоколам RIP и RIPv2 используется сервер ripd, а поддержка RIPng реализована в программе ripngd.
- **OSPF.** Для работы по протоколу OSPF используется программа ospfd, а вариант OSPF для IPv6 реализован в программе ospf6d. Подобно RIP, OSPF применяется для маршрутизации пакетов в сетевых структурах, насчитывающих несколько локальных сетей.
- BGP (Border Gateway Protocol — пограничный шлюзовый протокол) широко используется в Internet. Для поддержки данного протокола предназначен сервер bgpd.

Общее управление работой пакета осуществляет программа zebra. Серверы, входящие в состав пакета, используют ее для обновления таблицы маршрутизации. Программа zebra выполняется как сервер; обратиться к ней можно с помощью клиентской программы telnet.

Каждый из демонов маршрутизации выполняется независимо от других. Например, если вам нужно обеспечить поддержку RIP или RIPv2, вы можете запустить только программы zebra и ripd. Работой каждого сервера управляет отдельный конфигурационный файл, расположенный в каталоге /etc или /etc/zebra. Имя файла совпадает с именем соответствующего демона. Например, содержимое файла /etc/zebra/ospfd.conf определяет конфигурацию сервера ospfd. Все конфигурационные файлы строятся по единому принципу. Символы ! и # являются признаками комментариев. Опции, используемые для определения конфигурации, перечислены ниже.

- hostname. В качестве значения данной опции задается имя узла, выполняющего функции маршрутизатора.
- password. Программа zebra использует пароль для управления доступом других систем и серверов. Пароль необходимо задать в каждом конфигурационном файле. Этот пароль предоставляет ограниченный доступ к серверу.
- enable password. Данная опция позволяет задать специальный административный пароль, используемый программой zebra. Этот пароль надо задать в, том случае, если вам необходимо изменить конфигурацию сервера.

- router *протокол*. Конфигурационные файлы серверов требуют указания протокола. Так, в файле `ripd.conf` указывается router rip, в файле `ospfd.conf` — router ospf, а в файле `bgpd.conf` — router bgp *номер_автономной_системы*. (Номера автономных систем назначаются подобно IP-адресам. Если вы хотите применить BGP только в своей локальной сети, вам надо использовать номер автономной системы в диапазоне 64512-65535.)

В процессе работы программы zebra вы можете изменить ее конфигурацию, обратившись к ней с помощью клиентской программы telnet. При обращении указывается порт 2601. Пример вызова telnet приведен ниже.

```
$ telnet localhost 2601
```

После ввода пароля надо задать одну из следующих команд: enable (получение доступа к командам настройки), configure (изменение конфигурации) или show (отображение сведений о текущей конфигурации). На каждом этапе работы вы можете получить информацию о доступных командах и опциях; для этого надо ввести символ ?. Если вы работали с маршрутизаторами Cisco, то команды Zebra знакомы вам.

Резюме

На каждом компьютере, подключенном к сети, в том числе и на рабочей станции, должна содержаться таблица маршрутизации, которая дает возможность направлять сетевой трафик по требуемому маршруту. Стандартные сетевые утилиты Linux, например ifconfig и route, подходят для заполнения таблицы маршрутизации на рабочей станции, сервере и даже на низкоуровневом маршрутизаторе. Если же маршрутизатор выполняет сложные действия по управлению пакетами, на компьютере должны присутствовать расширенные средства маршрутизации. Linux поддерживает подобные средства на нескольких уровнях. Поскольку за маршрутизацию отвечает ядро Linux, ряд инструментов, предназначенных для настройки маршрутизатора, требуют, чтобы некоторые опции ядра, ответственные за маршрутизацию, были активны. Популярный пакет iproute2 предоставляет инструменты маршрутизации, в частности, позволяет работать с несколькими таблицами маршрутизации или сформировать схему QoS с тем, чтобы обеспечить некоторому пользователю или сети возможность использовать определенную часть пропускной способности линии. Протоколы маршрутизации обеспечивают взаимодействие маршрутизаторов, позволяют обмениваться информацией о маршрутах и определять на основании полученных данных оптимальный путь к целевой сети.

Глава 25

Настройка средств обработки пакетов с помощью iptables

Средства ядра Linux, реализующие стек протоколов TCP/IP, получают данные от приложения, оформляют их в виде информационных пакетов и передают по сети. Из принимаемых пакетов извлекается содержащаяся в них информация и передается приложению. Считается, что ядро не должно изменять данные, за исключением тех преобразований, которые предусмотрены протоколами TCP/IP, однако это правило не всегда выполняется. Утилита `iptables` позволяет сконфигурировать ядро Linux так, что оно будет фильтровать и даже преобразовывать пакеты данных на основании различных критериев. Такими критериями могут быть адрес источника и адрес назначения, указанные в пакете. Способность организовать процесс фильтрации пакетов позволяет использовать `iptables` для реализации брандмауэров и преобразователей NAT (Network Address Translation — преобразование сетевых адресов). В данной главе рассматриваются создание брандмауэров и NAT-преобразователей, а также вопросы перенаправления портов и протоколирования хода обработки пакетов. Все решения, о которых идет речь в этой главе, в основном направлены на обеспечение безопасности локальных сетей или отдельных компьютеров.

Средства `iptables` позволяют реализовать простые брандмауэры и другие инструменты фильтрации пакетов. Если же брандмауэр, предназначенный для защиты вашей сети, должен выполнять сложные функции, то для его создания вам понадобится дополнительная информация. Необходимые сведения вы можете получить в книге Зиглера (Ziegler) *Linux Firewalls, 2nd Edition* (New Riders, 2001), а также в книге Сонненрейча (Sonnenreich) и Йетса (Yates) *Building Linux and OpenBSD Firewalls* (Wiley, 2000). Вторая из рекомендованных книг в основном ориентирована на применение инструмента `ipchains`, который использовался для создания брандмауэров до появления `iptables`.

Что такое iptables

Для обработки сетевых пакетов ядро 2.4.x использует процедуру, подобную той, которая условно изображена на рис. 25.1. В начале обработки ядро выясняет, предназначен ли пакет для локального компьютера или должен быть перенаправлен на другой узел сети. В зависимости от ответа на этот вопрос, пакет передается одной из двух *цепочек:* INPUT

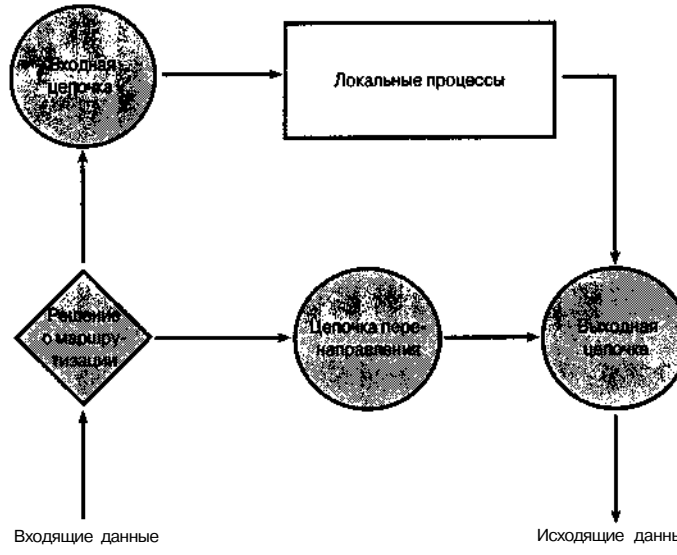


Рис. 25.1. Для обработки информационных пакетов сетевое ядро Linux использует несколько цепочек

или FORWARD. Эти цепочки могут обрабатывать информацию различными способами, но по умолчанию они не изменяют данные. Цепочка INPUT передает информацию локальным процессам. В роли *локальных процессов* могут выступать клиентские программы (например, Netscape, `telnet` и др.) или серверы (Apache, `telnetd` и др.). В большинстве случаев эти программы выполняются как пользовательские процессы, но они могут быть и процессами ядра. Примерами приложений, которые выполняются как процессы ядра, являются средства поддержки NFS, реализованные в ядре, и Web-сервер kHTTPd. Как информация, генерируемая локальными процессами, так и выходные данные цепочки FORWARD предаются для обработки с помощью цепочки OUTPUT.

Информационный пакет не обязательно должен проходить весь цикл обработки, показанный на рис. 25.1. Некоторые пакеты могут быть блокированы одной из цепочек; не исключено также, что локальный процесс, получив пакет, не станет отвечать на него. В некоторых случаях транзакция инициируется локальным процессом. Ответ на запрос, сгенерированный локальным процессом, будет получен на входе системы.

Каждая из цепочек, показанных на рис. 25.1, предоставляет возможность обрабатывать пакеты. Фильтрация пакетов осуществляется на основании анализа таких данных, как IP-адрес источника и назначения, порт источника и назначения, а также интерфейс, через который передаются пакеты. Каждая из цепочек представляет собой набор правил, на соответствие которым проверяются пакеты. Если пакет соответствует условию правила, над ним выполняются *действия,* предусмотренные в правилах. При создании брандмауэров используются идентификаторы, определяющие действия. К ним относятся ACCEPT (принять пакет для обработки), DROP (игнорировать пакет), QUEUE (передать пакет пользовательскому процессу) и RETURN (прекратить обработку и вернуться к вызы-

вающей цепочке). Некоторые действия требуют активизации опций ядра. К ним относятся REJECT (отвергнуть пакет, сообщив об этом отправителю), MASQUERADE (используется при организации NAT-преобразования) и LOG (применяется для протоколирования хода фильтрации).

Цепочки объединяются в *таблицы*. Цепочки, показанные на рис. 25.1, составляют таблицу filter, которая используется для обработки стандартных типов трафика. Стандартными таблицами также являются nat (она используется при построении NAT-преобразователей) и mangle (с ее помощью осуществляются некоторые типы преобразования пакетов). Вы можете поместить в таблицу новые цепочки и вызвать их из существующих цепочек. Это позволяет реализовать сложные процедуры фильтрации.

Таблицы и цепочки являются средствами ядра Linux, а iptables — это программа, которая выполняется как пользовательский процесс и предоставляет возможность управлять таблицами и цепочками. Программу iptables можно использовать для добавления правил к любой из цепочек, показанных на рис. 25.1, а также к другим цепочкам. Например, вы можете включить в цепочку INPUT правила, блокирующие все пакеты, в заголовке которых указан определенный порт назначения, или добавить в цепочку OUTPUT правила, запрещающие передавать пакеты системе, взаимодействие с которой по каким-либо причинам запрещено. С помощью этих и других цепочек вы можете реализовать брандмауэр, NAT-преобразователь или другое средство защиты системы.

Изменения, вносимые утилитой iptables, носят временный характер; информация о них удаляется после перезагрузки компьютера. По этой причине для работы с iptables следует создавать сценарии. В состав некоторых дистрибутивных пакетов, например Red Hat и Mandrake, включаются инструментальные средства, упрощающие создание брандмауэров и NAT-преобразователей. Сценарий, предназначенный для создания правил посредством утилиты iptables, обычно запускается как сценарий SysV или локальный сценарий запуска.

Альтернативные средства фильтрации

Программа iptables была создана для работы с ядром 2.4.x. С ранними версиями ядра использовались другие инструменты. Например, для взаимодействия с соответствующими средствами ядра 2.2.x применялась программа ipchains, а для работы с ядром 2.0.x — программа ipfwadm. Смена инструментов отражает изменения в структуре ядра. Программа iptables дает возможность работать с такими средствами ядра 2.4.x, которые отсутствовали в ядре 2.2.x. Например, она позволяет выполнять *проверку пакетов с учетом состояния* (stateful packet inspection), при которой учитываются характеристики соединения. Проверка пакетов с учетом состояния предоставляет дополнительные возможности по организации защиты компьютеров.

При работе с версиями ядра, предшествующими версии 2.4.x, вам придется использовать ipchains или ipfwadm. В данной главе не уделяется внимание работе с этими программами, поэтому всю необходимую информацию вам придется искать в документации на соответствующий инструмент. Для работы со средствами фильтрации пакетов, которые будут реализованы в последующих версиях ядра, наверное, будут разработаны новые инструментальные средства. Вероятнее всего, что общие принципы их работы будут такими же, какие используются в iptables, поэтому знание этой программы пригодится при работе с версиями ядра, которые придут на смену версии 2.4.x.

Если вы хотите продолжать работу с инструментами ipfwadm и ipchains, вы можете использовать их и для взаимодействия с ядром 2,4,x, но для этого надо настроить соответствующим образом ядро системы. Программы ipfwadm и ipchains позволяют решать те же задачи, которые решаются при работе с версиями 2.0.x и 2.2.x, но вы не сможете воспользоваться новыми возможностями, предоставляемыми ядром 2.4.x. Некоторые из правил фильтрации пакетов, реализуемые посредством iptables, дублируют соответствующие возможности TCP Wrappers, xinetd и средств контроля доступа к отдельным серверам. Все эти инструменты позволяют ограничить возможность взаимодействия с серверами на основе анализа IP-адресов. Если одно и то же ограничение может быть реализовано несколькими инструментами, я рекомендую не ограничиваться использованием одного из них. При одновременном применении нескольких средств последствия ошибки в конфигурации или в коде одной из программ будут устранены другими программами. По сравнению с прочими инструментами подобного назначения iptables реализует средства более низкого уровня, поэтому ограничения, накладываемые с помощью этой программы, охватывают большее число протоколов и серверов. Например, если xinetd защищает только серверы, запускаемые с его помощью, то iptables позволяет ограничить доступ ко всем серверам.

Конфигурация ядра для работы с iptables

Для того чтобы использовать iptables, необходимо активизировать соответствующие средства ядра. В версии ядра 2.4 все необходимые для этого опции сосредоточены в меню Networking Options и некоторых его подменю. Опции, которые необходимо активизировать, перечислены ниже.

- Network Packet Filtering. Данная опция расположена в меню Networking Options.
- Connection Tracking. Эта опция находится в подменю Netfilter Configuration меню Networking Options. Данная опция используется при создании NAT-преобразователей. (Все последующие опции также расположены в подменю Netfilter Configuration.)
- FTP Protocol Support. При работе NAT-преобразователя особые трудности связаны с поддержкой протокола FTP. В системе Linux для этой цели создан специальный модуль ядра.
- IP Tables Support. Данная опция также необходима для работы NAT-преобразователя. При выборе этой опции становится доступным большое число подопций, соответствующих различным типам проверки. Чтобы обеспечить наибольшую гибкость, желательно выбрать все подопций. Особенно важна подопция Connection State Match Support, поскольку она используется для проверки пакетов с учетом состояния.
- Packet Filtering. Несмотря на то что эта опция не является абсолютно необходимой для создания брандмауэров и NAT-преобразователей, она расширяет набор доступных возможностей. Поэтому я рекомендую вам активизировать ее.

- REJECT Target Support. Данная подопция опции Packet Filtering добавляет правило, которое может быть использовано при создании брандмауэров. Поэтому имеет смысл активизировать эту опцию.
- Full NAT. Средства, включаемые с помощью данной опции, требуются для реализации многих возможностей NAT, включая те, которые описаны в данной главе.
- MASQUERADE Target Support. Данная подопция опции Full NAT необходима для реализации IP-маскировки — разновидности NAT-преобразования, которая будет описана ниже. В справочной информации, вызываемой после щелчка на кнопке Help, сказано, что опция MASQUERADE Target Support нужна только при использовании динамических внешних IP-адресов, однако это не так. Данная опция требуется при выполнении IP-маскировки, независимо от того, являются ли внешние IP-адреса динамическими.
- Packet Mangling. Средства ядра, включаемые с помощью данной опции, нужны, если вы собираетесь использовать таблицу mangle. *Я* рекомендую вам активизировать опцию Packet Mangling.
- LOG Target Support. Если вы хотите протоколировать работу брандмауэра или маршрутизатора, данная опция позволит вам сделать это.
- ipchains (2.2-style) Support. Если вы хотите использовать сценарии брандмауэра, ориентированные на работу с ipchains, вам необходимо активизировать данную опцию. Для выполнения этих сценариев вам также потребуется программа ipchains.
- ipfwadm (2.0-style) Support. Если вы хотите использовать сценарии брандмауэра, предназначенные для работы с ipfwadm, вам необходимо активизировать данную опцию. Для выполнения этих сценариев вам также потребуется программа ipfwadm.

СОВЕТ ✔ Опции, включающие поддержку ipchains и ipfwadm, являются взаимоисключающими и не совместимы с опциями IP Tables Support и Connection Tracking. Поэтому нельзя одновременно включать опции, предназначенные для работы с iptables и более старыми инструментами подобного назначения. Однако вы можете скомпилировать все необходимые средства как модули и загружать тот или иной модуль по мере необходимости. Такая конфигурация оправдана в том случае, если вы применяете один из старых инструментов, но планируете переходить на использование iptables. Во многих дистрибутивных пакетах ядро скомпилировано подобным образом по умолчанию.

Если вы скомпилировали некоторые средства ядра в виде модулей, вам необходимо организовать загрузку этих модулей. Обычно загрузку модулей предусматривают в сценарии запуска брандмауэра. Например, если средства поддержки работы iptables находятся в модуле ip_tables, в сценарии запуска должна присутствовать команда insmod ip_tables. Чтобы найти другие модули, предназначенные для загрузки, надо просмотреть каталог /lib/modules/версия/net/ipv4/netfilter. Включив требуемые средства в состав ядра, вы избавитесь от необходимости загружать модули, но при этом увеличатся размеры ядра.

Проверка текущей конфигурации iptables

Перед тем как приступать к решению каких-либо задач, предполагающих использование iptables, необходимо проверить текущую конфигурацию. В составе некоторых дистрибутивных пакетов поставляются инструменты для создания брандмауэров, и не исключено, что к данному моменту они уже были запущены. Чтобы проверить конфигурацию системы, надо указать при вызове iptables опцию -L. Добавив опцию -t имя-таблицы, вы сможете получить информацию о состоянии конкретной таблицы. (Чаще всего проверяется состояние таблицы filter, но вы можете также указать при вызове iptables таблицу nat или mangle.) При запуске с использованием опции -L программа iptables выведет данные, подобные приведенным ниже.

```
# iptables -L -t filter
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Данные, отображаемые при вызове этой программы, указывают на то, что в стандартной таблице filter правила отсутствуют. Если правила, определяющие действия брандмауэра, уже заданы, вам необходимо выяснить, какой сценарий создает их, и запретить его выполнение. (Часто для установки правил используется сценарий SysV, содержащийся в файле firewall, или сценарий с другим подобным именем.) Для того чтобы исключить правила из цепочки, надо использовать опцию -F.

```
# iptables -F INPUT -t filter
```

Подобные команды для цепочек, содержащихся в таблице filter и в других таблицах, часто включают в начало сценария брандмауэра. Это гарантирует, что вновь определяемые правила не будут конфликтовать с правилами, созданными ранее.

Создание брандмауэра средствами iptables

Утилита iptables может решать различные задачи. Одной из таких задач является создание брандмауэров. Брандмауэр можно реализовать как на компьютере, выполняющем функции маршрутизатора, так и на рабочих станциях или серверах. При настройке брандмауэра, осуществляющего фильтрацию пакетов, сначала задается политика по умолчанию, а затем определяются исключения. В процессе работы брандмауэр анализирует IP-адреса, номера портов и другие характеристики пакетов.

Что такое брандмауэр

Обычно, когда говорят о брандмауэре, имеют в виду компьютер, расположенный между двумя сетями и управляющий доступом из одной сети в другую. Несмотря на то что маршрутизатор также управляет обменом пакетами между различными сетями, эти инструменты существенно отличаются друг от друга. Брандмауэр может блокировать до-

ступ компьютеров одной сети к некоторым службам другой сети. Например, брандмауэр может запретить обращения по протоколу Telnet из Internet к компьютерам локальной сети. Маршрутизатор не выполняет подобных действий. Брандмауэр, в свою очередь, также осуществляет не все операции, выполняемые маршрутизатором. Так, например, брандмауэры, выступающие в роли proxy-серверов, частично обрабатывают запросы, направленные другим системам, преобразуют их так, что они выглядят как сформированные самим брандмауэром, и перенаправляют ответы системам, от которых были получены запросы. Брандмауэры, выполняющие функции proxy-серверов, представляют собой мощные средства защиты. Они позволяют даже защитить компьютеры от вирусов, встроенных в программы Java и JavaScript.

В данном разделе рассматриваются брандмауэры, выполняющие фильтрацию пакетов. Они действуют на нижнем уровне стека протоколов TCP/IP, контролируют данные, содержащиеся в заголовках отдельных пакетов, и даже проверяют, корректно ли осуществляются транзакции. Часто брандмауэры реализуются на компьютерах, выполняющих роль маршрутизаторов, но они также могут быть установлены на рабочих станциях и серверах. Если брандмауэр расположен на отдельном компьютере, он защищает лишь ресурсы этой машины и не оказывает влияния на работу других узлов сети.

Многие рассматривают брандмауэры как инструменты, предназначенные для защиты локальных сетей от нежелательного воздействия из Internet. Действительно, брандмауэры очень часто используются в подобных целях. (Пример такого брандмауэра показан на рис. 25.2.) Однако брандмауэры часто выполняют и другие функции. Например, вы можете создать брандмауэр, который будет защищать узлы Internet от атаки, предпринимаемой с узлов локальной сети. Брандмауэр может блокировать все протоколы, за исключением некоторых, необходимых вам, и даже запретить обмен с определенными компьютерами посредством ряда протоколов. Например, вы имеете возможность разрешить обращение к порту 25 удаленных компьютеров только почтовому серверу. (Подобную конфигурацию брандмауэра используют некоторые провайдеры для борьбы со спамом.) Контроль обращений к внешним узлам не позволит недобросовестным пользователям локальной сети нанести вред удаленному компьютеру, а также даст возможность выявить вирусы и программы типа "троянский конь", которые тем или иным способом попали на компьютеры локальной сети. Несмотря на то что подобные меры в основном направлены на защиту внешних узлов, они могут оказаться полезными и для вас, так как предотвратят конфликты с администраторами внешних сетей.

В некоторых случаях правила брандмауэра можно использовать для перенаправления обращений. При этом пакет, адресованный одной системе, передается другой системе. Правила перенаправления в сочетании со средствами NAT могут применяться для защиты серверов, работающих в локальной сети. Осуществляя перенаправление пакетов, можно добиться того, что запрос будет обработан неожиданным для клиента способом. Например, вместо того, чтобы блокировать исходящие SMTP-соединения, вы можете перенаправить их на локальный почтовый сервер. Если брандмауэр настроен так, что запросы на установление SMTP-соединений, сгенерированные сервером SMTP, пропускаются беспрепятственно, перенаправление SMTP-запросов от клиентов приведут к тому, что почта будет доставляться адресатам. (Чтобы это произошло, надо также настроить локальный сервер SMTP в качестве ретранслятора для локальных компьютеров.) Следует заметить, что подобный подход применим лишь для отдельных типов серверов.

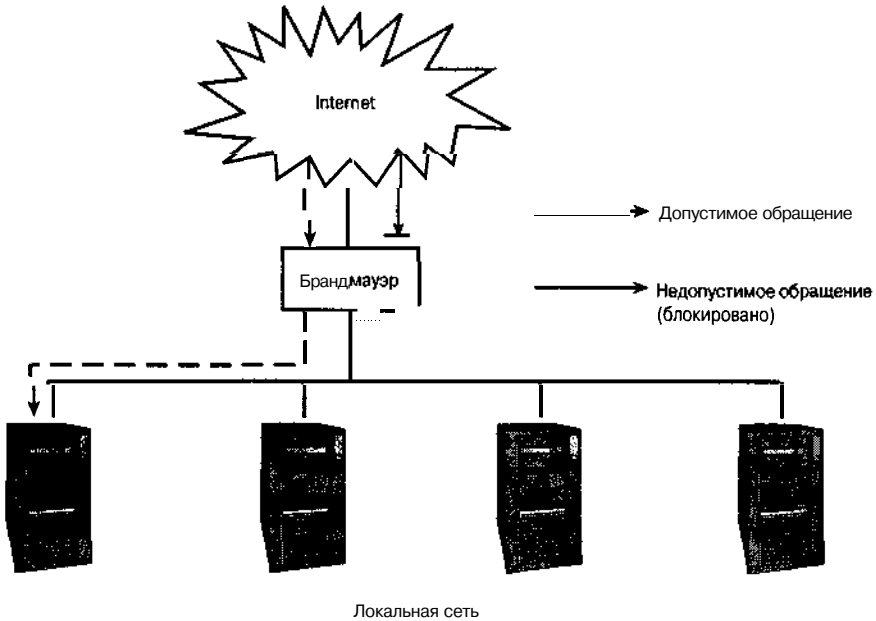


Рис. 25.2. Брандмауэры, выполняющие фильтрацию пакетов, позволяют блокировать некоторые типы обращений к локальной сети

Как видно на рис. 25.1, для того, чтобы обеспечить фильтрацию пакетов в системе Linux, надо настроить цепочки INPUT, FORWARD и OUTPUT. Назначение каждой из этих цепочек кратко описано ниже.

- Цепочка INPUT защищает локальные процессы. Эту цепочку используют как брандмауэры, совмещенные с маршрутизаторами, так и брандмауэры, установленные на рабочих станциях и серверах.

- Цепочка FORWARD принимает непосредственное участие в маршрутизации пакетов. Если вы хотите превратить маршрутизатор в брандмауэр, осуществляющий фильтрацию пакетов, вам надо сконфигурировать эту цепочку.

- Цепочка OUTPUT блокирует передачу нежелательных выходных данных. Эту цепочку используют как брандмауэры, расположенные на отдельных компьютерах, так и брандмауэры, совмещенные с маршрутизаторами. С ее помощью можно ограничить возможности локальных клиентов по использованию протоколов или запретить им взаимодействие с некоторыми узлами.

Брандмауэры, совмещенные с маршрутизаторами, чаще всего применяют правила, содержащиеся в цепочках INPUT и FORWARD, а брандмауэры на рабочих станциях и серверах в основном работают с правилами в цепочках INPUT и OUTPUT. В некоторых случаях результаты использования правил в различных цепочках совпадают, в особенности это справедливо для цепочек FORWARD и OUTPUT, Различие лишь в том, что цепочка OUTPUT воздействует как на перенаправляемый трафик, так и на трафик, сгенерирован-

ный локальным компьютером, в то время как цепочка FORWARD контролирует только перенаправляемый трафик.

Формирование политики по умолчанию

Первым шагом, предпринимаемым при настройке брандмауэра, является формирование *политики по умолчанию*. Политика по умолчанию — это выражение, определяющее, что должен делать брандмауэр, если пакет не удовлетворяет ни одному из правил. Для создания политики по умолчанию используется опция -P утилиты iptables.

```
# iptables -P INPUT DROP
# iptables -P OUTPUT DROP
# iptables -P FORWARD DROP
```

В данном примере задается политика по умолчанию для трех стандартных цепочек, содержащихся в таблице filter. В качестве политики по умолчанию может быть указано любое из описанных ранее действий (ACCEPT, DROP, QUEUE, RETURN и т. д.). Наиболее часто используются действия ACCEPT, DROP и REJECT. ACCEPT указывает Linux на то, что все пакеты должны передаваться, а DROP заставляет систему игнорировать все пакеты. REJECT, подобно DROP, также указывает на то, что пакеты должны отвергаться, но при этом Linux оповещает источник о том, что пакет не принят (подобное сообщение источник получает и в том случае, если в системе нет ни одного сервера, ожидающего обращения через порт, указанный в заголовке пакета). Если брандмауэр должен обеспечивать высокую степень защиты, в качестве политики по умолчанию указывается DROP или REJECT, однако при этом все пакеты, передача которых не разрешена явным образом, будут отвергнуты. Если задана политика по умолчанию ACCEPT, то необходимо явно запретить все типы пакетов, которые не должны быть пропущены через брандмауэр. Составление правил, блокирующих все недопустимые типы пакетов, часто представляет собой достаточно сложную задачу, причем всегда остается опасность, что какое-либо условие останется не учтенным. С другой стороны, задавая политику по умолчанию DROP или REJECT, надо лишь разрешить прохождение некоторых типов пакетов через брандмауэр. Обычно при работе системы число типов пакетов ограничено, поэтому данному подходу следуют большинство системных администраторов.

Определение правил

Для создания правил используется опция --append (или -A) программы iptables. После этой опции задается один или несколько критериев, затем указывается опция --jump (или -j), за которой следует действие ACCEPT, DROP или REJECT. Вызов iptables, предназначенный для создания действия, выглядит следующим образом:

```
# iptables --append CHAIN критерий_выбора --jump действие
```

Сокращенно та же команда может быть записана так:

```
# iptables -A CHAIN критерий_выбора -j действие
```

Вместо --append при вызове iptables могут быть указаны следующие опции.

- --delete, или -D. Эта опция удаляет правило из существующей цепочки.
- --insert, или -I. С помощью данной опции вы можете включить правило в середину цепочки. При этом необходимо задать номер правила. Если номер не ука-

зан, iptables включит правило в начало цепочки (при использовании опции --append правило помещается в конец цепочки).

- --replace, или -R. Эта опция дает возможность заменить правило. Задавая данную опцию, следует указать номер заменяемого правила.
- --list, или -L. Данная опция отображает все правила в цепочке.

Для утилиты iptables предусмотрены также другие опции. Информацию о них вы можете получить на страницах справочной системы, посвященных iptables. В следующем разделе рассматривается формирование критериев выбора, передаваемых iptables. В одной команде можно задать несколько критериев, например, вы можете ограничить доступ по номеру порта и по IP-адресу.

Ядро системы читает правила в цепочке по порядку и применяет первое из них, которому соответствует пакет. Если вы хотите задать исключение из какого-либо правила (например, запретить доступ к порту Telnet для всех узлов, кроме машин, принадлежащих локальной сети), вы должны поместить исключение перед основным правилом. Политика по умолчанию по сути представляет собой правило, находящееся в самом конце цепочки. Ему удовлетворяют все пакеты, которые не соответствуют ни одному другому правилу в цепочке.

Использование номеров портов при выполнении фильтрации

При выполнении фильтрации пакетов могут анализироваться порты источника и назначения. Например, брандмауэр, находящийся на компьютере, на котором выполняется почтовый сервер, можно настроить для передачи пакетов, в которых указан порт назначения 25. Для этого используется опция --destination-port(--dport). Аналогичных результатов можно добиться, используя опцию --protocol (-p), в качестве значения которой указывается тип протокола (tcp, udp, icmp или all). Опция --source-port (--sport) выполняет подобные действия, но задает порт источника. Команды, определяющие правила фильтрации на основе номеров портов, выглядят следующим образом:

```
# iptables -A INPUT -p tcp --dport 25 -j ACCEPT
# iptables -A OUTPUT -p tcp --sport 25 -j ACCEPT
```

Эти команды обеспечивают прием пакетов, направленных серверу, который ожидает поступление запросов через порт 25, и передачу пакетов, возвращаемых сервером в ответ на запрос (в них указан порт источника 25). В результате, даже если политика по умолчанию отвергает пакеты, сервер сможет получить почту от внешних серверов. Заметьте, если в качестве политики по умолчанию указано действие DROP или REJECT, вы должны включить в цепочку INPUT правило, разрешающее принимать пакеты, направленные серверу, а в цепочку OUTPUT — правило, разрешающее передавать пакеты, сгенерированные данным сервером. Для этого при определении правила для цепочки INPUT задается опция --destination-port, а при определении правила для цепочки OUTPUT — опция --source-port. Если вы забудете создать одно из правил, то сервер сможет получать запросы, но не сможет генерировать ответы на них, либо, наоборот, через брандмауэр будут пропускаться только данные, сгенерированные сервером, а информация, направленная серверу, будет отвергаться. Для брандмауэра, совмещенного с маршрутизатором, надо также включить в цепочку FORWARD правила, созданные с использованием опций

--destination-port и --source-port, в противном случае данные через брандмауэр передаваться не будут. Вы можете использовать в качестве условий номера портов в сочетании с IP-адресами. Это позволит не только ограничить обмен определенным типом протокола, но и разрешить его лишь для отдельных компьютеров. Так, например, вы сможете создать правила, согласно которым взаимодействовать с внешними узлами по протоколу SMTP будет иметь право только почтовый сервер.

Если вы используете политику по умолчанию DROP или REJECT, вам необходимо разрешить клиентским программам взаимодействовать с внешними серверами. Для этого выполните следующие действия.

- Разрешите доступ к серверным портам внешних компьютеров. Соответствующее правило, включаемое в цепочку INPUT, должно создаваться с указанием опции --source-port, а при создании правила, помещаемого в цепочку OUTPUT, должна использоваться опция --destination-port. Для брандмауэра, совмещенного с маршрутизатором, необходимо также включить в цепочку FORWARD правила, созданные с помощью опций --source-port и --destination-port. Возможно, что наряду с номерами портов вам потребуется задать IP-адреса ваших компьютеров. Таким способом вам следует разрешить обращение вовне по каждому из протоколов, которые используются клиентами, выполняемыми в вашей сети.

- Разрешите доступ к непривилегированным портам компьютеров вашей сети. Номера непривилегированных портов лежат в диапазоне 1024–65535. В опциях --source-port и --destination-port указываются границы диапазона, разделенные двоеточием, например --source-port 1024:65535. Для принимаемых пакетов вы можете указать также опцию ! syn. Правилам, в которых указана опция --syn, соответствуют только пакеты, содержащие запросы на установление соединений, а символ ! означает отрицание, т. е. заданному правилу будут удовлетворять только пакеты, которые были переданы серверами в ответ на запросы клиентов.

Использование IP-адресов при выполнении фильтрации

При создании правил могут указываться IP-адреса или блоки IP-адресов. IP-адрес источника задается с помощью опции --source (-s), а IP-адрес назначения — посредством опции --destination(-d). Например, если вы хотите запретить взаимодействие с компьютерами сети 172.24.0.0/16, вам надо создать правила, которые отвергали бы пакеты, переданные из указанной сети, а также пакеты, адресованные компьютерам этой сети. Соответствующие команды имеют следующий вид:

```
# iptables -A INPUT -s 172.24.0.0/16 -j DROP
# iptables -A OUTPUT -d 172.24.0.0/16 -j DROP
```

Опции -s и -d часто используются вместе с опциями, определяющими номера портов. Таким образом, вы можете сформировать правила, согласно которым взаимодействовать по сети будут иметь право только определенные компьютеры, обращающиеся по определенным портам. Предположим, например, что вы создаете брандмауэр для защиты локальной сети, но хотите при этом разрешить удаленным пользователям, работающим в сети 10.34.176.0/24, обращаться к серверам SSH локальной сети (серверы SSH ожидают обращения через порт 22). Для этого надо определить следующие команды:

```
# iptables -A FORWARD -s 10.34.176.0/24 -p tcp \
  --destination-port 22 -j ALLOW
t iptables -A FORWARD -d 10.34.176.0/24 -p tcp \
  --source-port 22 -j ALLOW
```

Поскольку в данном примере модифицируется только цепочка FORWARD, пользователям не предоставляется доступ к серверу SSH компьютера, на котором выполняется брандмауэр (если такой сервер имеется на этой машине). Возможно, вы захотите создать правила, которые разрешали бы обращаться к этому серверу с компьютеров локальной сети. Если адрес вашей локальной сети 192.168.9.0/24, то соответствующие команды будут выглядеть так:

```
# iptables -A INPUT -s 192.168.9.0/24 -p tcp \
  --destination-port 22 -j ALLOW
# iptables -A OUTPUT -d 192.168.9.0/24 -p tcp \
  --source-port 22 -j ALLOW
```

Использование информации об интерфейсах при выполнении фильтрации

При создании правил фильтрации можно указывать сетевой интерфейс, например ppp0 или eth1. Данный подход в основном используется на компьютерах с несколькими интерфейсами, выполняющих функции маршрутизаторов. Применение в составе правила сведений об интерфейсе позволяет противодействовать фальсификации адресов, в частности, включению в заголовки пакетов, приходящих извне, адресов компьютеров локальной сети. Правила, в которых интерфейс задается с помощью опции --in-interface (-i), как правило, помещаются в цепочки INPUT и FORWARD, а правила, создаваемые с использованием опции --out-interface (-o), обычно предназначены для включения в цепочки FORWARD и OUTPUT. Предположим, что адрес вашей локальной сети 192.168.9.0/24, маршрутизатор, совмещенный с брандмауэром, подключен к ней с помощью интерфейса eth1, а соединение маршрутизатора с Internet осуществляется посредством интерфейса eth0. Правила, препятствующие фальсификации адресов, имеют следующий вид:

```
# iptables -A INPUT -s 192.168.9.0/24 -i eth0 -j DROP
# iptables -A FORWARD -s 192.168.9.0/24 -i eth0 -j DROP
# iptables -A FORWARD -s !192.168.9.0/24 -i eth1 -j DROP
# iptables -A OUTPUT -s !192.168.9.0/24 -i eth1 -j DROP
```

Первые две команды отвергают поступающие извне (через интерфейс eth0) пакеты, адресованные маршрутизатору или компьютерам локальной сети, в которых указано, что они отправлены из локальной сети. Последние две команды блокируют пакеты, направленные в Internet (поступающие через интерфейс eth1), в которых указан IP-адрес источника, не совпадающий с адресами компьютеров локальной сети.

Проверка пакетов с учетом состояния

Одна из самых новых возможностей фильтрации пакетов, реализованных в системе Linux, позволяет учитывать при проверке пакетов состояние соединения. Средства, рассмотренные ранее в этой главе, позволяли обрабатывать отдельные пакеты, независимо от того, являлись ли они частью соединения или были специально сгенерированы для организации атаки. (Ранее уже встречалась опция --syn, позволяющая определить

пакет, содержащий запрос на установление соединения. Существуют средства, которые предоставляют возможность включить свои пакеты в набор пакетов, передаваемых в рамках действующего соединения. Такое включение пакетов называется перехватом TCP-соединения.) Средства проверки пакетов с учетом состояния определяют принадлежность пакетов к текущему соединению, анализируя последовательные номера, IP-адреса, указанные в заголовках, и другие характеристики пакетов. Правила, реализующие такую проверку, позволяют отвергать посторонние пакеты, включенные в состав данных, которые передаются в рамках существующего соединения.

Для включения средств проверки пакетов с учетом состояния используется опция `--state`, предваряемая опцией -т *состояние*. Для опции `--state` можно задать одно или несколько значений. Если вы указываете несколько значений, они должны разделяться запятыми. Символ ! перед опцией `--state` изменяет ее действие на обратное. Ниже перечислены допустимые параметры опции `--state`.

- INVALID. Проверка показала, что пакет не принадлежит известному соединению и может оказаться фальсифицированным.
- NEW. Пакет пытается установить новое соединение.
- ESTABLISHED. Пакет соответствует существующему соединению.
- RELATED. Пакет не является частью существующего соединения, но его присутствие допустимо (например, это может быть ICMP-пакет, сообщающий об ошибке).

Опция ! `--state` INVALID эквивалентна опции `--state` NEW, ESTABLISHED,RELATED.

Рассмотрим пример проверки с учетом состояния. Предположим, что, настраивая брандмауэр на отдельном компьютере, вы задали политику по умолчанию DROP или REJECT, но хотите разрешить взаимодействие с сервером HTTP через порт 80. Вы можете задать поверку с учетом состояния, в ходе которой будут отвергаться пакеты, не предназначенные для установления соединений, не принадлежащие к существующим соединениям и не относящиеся к пакетам, присутствие которых допустимо. Команды, предназначенные для создания правил, имеют следующий вид:

```
# iptables -A INPUT -m state -p tcp --dport 80 \
  --state NEW,ESTABLISHED,RELATED -j ACCEPT
# iptables -A OUTPUT -m state -p tcp --sport 80 \
  --state ESTABLISHED,RELATED -j ACCEPT
```

Эти правила включают проверку входящих и исходящих пакетов. Для проверки пакетов, передаваемых с компьютера, значение NEW опции `--state` не задано, так как новое соединение может устанавливаться только по инициативе клиента. Эти правила препятствуют перехвату существующих соединений с Web-сервером.

Проверка пакетов с учетом состояния может осуществляться только в тех системах, в которых используется версия ядра 2.4.x. Предыдущими версиями ядра такая возможность не поддерживается. Это может стать одним из стимулов перехода к использованию iptables.

Использование дополнительных опций

Программа iptables поддерживает большое количество опций, которые могут быть использованы для создания брандмауэров. Например, посредством опции --new-chain (-N) можно создать новую цепочку, указав опцию --fragment (-f), можно создать правило, которое будет применяться ко второму и к последующим фрагментам фрагментированного пакета, а опция --tcp-flags дает возможность организовать проверку на присутствие флагов в составе TCP-пакета. Дополнительную информацию об этих и о других опциях вы можете получить на страницах справочной системы Linux, посвященных iptables.

Сценарий для создания брандмауэра

В завершение разговора о средствах фильтрации пакетов рассмотрим сценарий, который создает брандмауэр. Код сценария представлен в листинге 25.1. Данный сценарий предназначен для компьютера, на котором выполняется Web-сервер и который поддерживает SSH-соединения с компьютерами, подключенными к локальной сети.

ВНИМАНИЕ

Сценарии брандмауэров, предназначенные для практического применения, содержат гораздо больший объем кода по сравнению представленным в листинге 25.1. Для удобства чтения в данном листинге при вызове iptables не указывается путь к файлу. В реальных сценариях это недопустимо. Не указывая путь к файлу, вы создаете угрозу безопасности системы. Код, приведенный в листинге 25.1, может послужить основой для создания более сложных сценариев.

Листинг 25.1. Простой сценарий, использующий iptables для создания брандмауэра

```
#!/bin/sh

iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD

iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Разрешить NDS-трафик
iptables -A INPUT -p udp --sport 53 -j ACCEPT
iptables -A OUTPUT -p udp --dport 53 -j ACCEPT

# Разрешить обмен клиентов с локальной сетью
iptables -A INPUT -m state -p tcp --dport 1024:65535 \
  --state ESTABLISHED,RELATED -s 192.168.9.0/24 -j ACCEPT
iptables -A OUTPUT -m state -p tcp --sport 1024:65535 \
  ! --state INVALID -d 192.168.9.0/24 -j ACCEPT

# Разрешить все HTTP-соединения
iptables -A INPUT -m state -p tcp --dport 80 \
```

```
  ! --state INVALID -j ACCEPT
iptables -A OUTPUT -m state -p tcp --sport 80 \
  --state ESTABLISHED, RELATED -j ACCEPT

# Разрешить обращения к SSH-серверу
I из локальной сети (192.168.9.0/24)
iptables -A INPUT -m state -p tcp --dport 22 \
  ! --state INVALID -s 192.168.9.0/24 -j ACCEPT          \
iptables -A OUTPUT -m state -p tcp --sport 22 \
  --state ESTABLISHED,RELATED -d 192.168.9.0/24 -j ACCEPT

# Разрешить прохождение локального трафика через интерфейс lo
iptables -A INPUT -s 127.0.0.1 -i lo -j ACCEPT
iptables -A OUTPUT -d 127.0.0.1 -o lo -j ACCEPT
```

Ниже описаны некоторые особенности кода, приведенного в листинге 25.1.

- **Удаление существующих правил и установка политики по умолчанию.** В первых шести строках программа iptables вызывается для удаления правил, присутствующих в цепочках, и установки политики по умолчанию. В качестве политики по умолчанию задается действие DROP. Несмотря на то что компьютер не выполняет маршрутизацию пакетов, политика по умолчанию задается также и для цепочки FORWARD. Это делается на случай, если на компьютере будет установлен еще один сетевой интерфейс.

- **Взаимодействие с сервером DNS.** Для того чтобы компьютер мог взаимодействовать с сервером DNS, две строки, следующие за комментариями "Разрешить NDS-трафик", предоставляют компьютеру возможность обращаться к удаленным серверам DNS (UDP-порт 53). Возможности соединения не ограничиваются одним адресом; компьютеру разрешено взаимодействовать с любым сервером имен. Если понадобится, вы можете наложить более жесткие ограничения.

- **Обмен с клиентами локальной сети.** Строки, следующие за комментариями "Разрешить обмен клиентов с локальной сетью", открывают путь трафику, связанному с непривилегированными портами (1024-65535). В цепочки INPUT и OUTPUT включены правила проверки пакетов с учетом состояния. Заметьте, что правило в цепочке INPUT запрещает установление новых соединений, поэтому, даже если на компьютере будет находиться сервер, принимающий обращения через непривилегированные порты, другие компьютеры не смогут обратиться к нему. Цепочки INPUT и OUTPUT ограничивают взаимодействие компьютерами локальной сети. При создании реального брандмауэра следует рассмотреть возможность замены этих правил более конкретными, которые разрешали бы прохождение данных между непривилегированными локальными портами и портами, используемыми для поддержки отдельных протоколов.

- **Трафик, связанный с Web-сервером.** Web-сервер, выполняющийся на компьютере, должен принимать обращения от любого узла сети, поэтому в правилах, регламентирующих обмен с Web-сервером, не указывается IP-адрес. Для того чтобы

противодействовать перехвату соединения, в этих правилах задана проверка пакетов с учетом состояния.

- **Трафик, связанный с сервером SSH.** Правила, определяющие взаимодействие с сервером SSH, во многом напоминают правила для Web-сервера, но в них указаны IP-адреса. В результате эти правила разрешают обращение к SSH-серверу только с компьютеров локальной сети.

- **Трафик обратной петли.** Для решения ряда системных задач, связанных с организацией работы системы, Linux использует интерфейс обратной петли (`lo`). Правила брандмауэра разрешают передачу данных через этот интерфейс. Проверка пакетов с учетом состояния не выполняется, так как трафик, направленный через интерфейс lo, поступает только по адресу 127.0.0.1.

Создание NAT-преобразователя с помощью iptables

Брандмауэры являются чрезвычайно полезными инструментами, но возможности `iptables` не ограничиваются созданием брандмауэров. В некоторых ситуациях большую помощь могут оказать NAT-преобразователи, которые также создаются посредством `iptables`. Протокол NAT позволяет модифицировать некоторые элементы TCP- и IP-пакетов, расширяя тем самым возможности адресации. Средства NAT настраиваются достаточно просто, но прежде чем приступить к настройке, следует выяснить, что такое NAT и какие возможности предоставляет этот инструмент.

Что такое NAT

Средства NAT позволяют изменять в процессе маршрутизации содержимое TCP- и IP-пакетов. В частности, при NAT-преобразовании изменяется IP-адрес источника и назначения в составе пакета. Ниже описаны ситуации, в которых оправданы подобные изменения адреса.

- **Установление соответствия между внешними и внутренними адресами.** Возможно, что, получив в свое распоряжение блок IP-адресов, вы не захотите перенастраивать сеть и будете использовать адреса, предназначенные для внутреннего пользования. Используя средства NAT, вы можете установить взаимно-однозначное соответствие между обычными Internet-адресами, выделенными для вашей сети, и адресами, которые реально присвоены вашим компьютерам.

- **Временное изменение адресов.** Средства NAT можно использовать для перенаправления запросов, адресованных некоторой системе, на другой компьютер. Предположим, например, что ваш Web-сервер вышел из строя и вы временно разместили его на другом компьютере. Проблему перенаправления запросов можно решить, изменяя конфигурацию сервера DNS, но средства NAT позволяют сделать это гораздо быстрее.

- **Распределение нагрузки.** С помощью NAT можно поставить в соответствие одному IP-адресу два компьютера внутренней сети и переключаться между ними при

передаче запросов. Такая форма распределения нагрузки считается очень грубой, но если один сервер не справляется со своей задачей, можно использовать подобное решение. Следует, однако помнить, что существуют более совершенные способы распределения нагрузки, не связанные с использованием NAT.

- Расширение адресного пространства. Если в вашем распоряжении имеется лишь ограниченное число IP-адресов, вы можете "спрятать" несколько компьютеров за одним IP-адресом. Такая возможность обычно используется в небольших сетях, подключенных к Internet по коммутируемой линии либо через соединение с широкой полосой пропускания. Если провайдер выделил для сети лишь один адрес, с помощью **NAT-преобразования** можно обеспечить работу всех компьютеров сети.

Расширение адресного пространства является наиболее частым применением NAT. Данная разновидность NAT-преобразования называется *IP-маскировкой*. В этом разделе будет рассматриваться именно этот способ использования NAT.

Средства NAT применяются совместно со средствами маршрутизации. В роли маршрутизатора, поддерживающего NAT, может выступать компьютер под управлением Linux. Настройка ядра системы производится с помощью программы **iptables**. Обычно компьютер, предназначенный для выполнения NAT-преобразования, содержит два сетевых интерфейса: посредством одного из них компьютер подключается к Internet, а с помощью другого соединяется с внутренней **сетью**.

Внешние узлы не должны идентифицировать компьютер, выполняющий NAT-преобразование, как маршрутизатор. В этом состоит одно из отличий NAT-маршрутизатора от обычного маршрутизатора.

Для того чтобы лучше понять работу NAT, рассмотрим процесс преобразования адреса с помощью NAT-маршрутизатора. Взаимодействие с сервером, расположенным в Internet, начинается по инициативе клиента (например, Web-броузера), который находится в сети, защищенной с помощью NAT-маршрутизатора. Предположим, что этот клиент пытается обратиться к Web-броузеру по адресу 172.18.127.45. Он генерирует HTTP-запрос; в пакетах, содержащих этот запрос, указывается локальный IP-адрес клиента (предположим, 192.168.9.32). Клиент передает запрос компьютеру, выполняющему роль локального шлюза; этот компьютер осуществляет NAT-преобразование. Получив пакет с запросом к Web-серверу, NAT-маршрутизатор анализирует его содержимое, заменяет IP-адрес источника на свой IP-адрес (допустим, что его адрес 10.34.176.7) и передает пакет по назначению. Web-сервер считает, что пакет поступил с компьютера, выполняющего функции NAT-маршрутизатора, поэтому направляет ему ответ. Получив ответ сервера, NAT-маршрутизатор распознает его как ответ на запрос, переданный с компьютера 192.168.9.32, выполняет обратное преобразование, заменяя адрес назначения в пакете, а затем передает пакет, содержащий ответ, клиенту. Рис. 25.3 иллюстрирует этот процесс. При этом ни клиент, ни сервер не знают о том, что адрес был преобразован средствами NAT, поэтому при использовании NAT-маршрутизатора не требуется изменять конфигурацию компьютеров в сети.

NAT-преобразование, а в особенности IP-маскировка, автоматически обеспечивает защиту компьютеров в локальной сети. Поскольку внешним компьютерам доступен только один IP-адрес, они не могут установить непосредственное соединение с внутренним компьютером. В локальную сеть извне передаются только ответы на запросы, отправленные

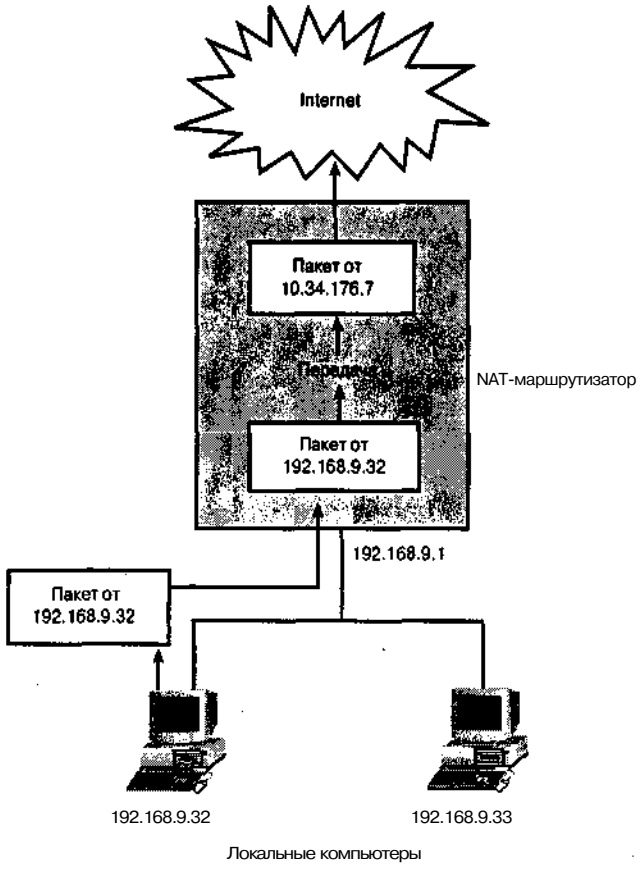


Рис. 25.3. NAT-маршрутизатор изменяет IP-адреса в пакетах

ими. По этой причине продукты NAT часто называют брандмауэрами, хотя между этими инструментами имеются существенные различия.

Помимо преимуществ, NAT имеет существенные недостатки.

- Автоматически создаваемая защита затрудняет размещение сервера во внутренней сети, расположенной за NAT-маршрутизатором, и обеспечение внешнего доступа к этому серверу. Чтобы доступ к серверу извне стал возможен, надо использовать перенаправление портов.

- Не все протоколы нормально взаимодействуют с NAT. Иногда IP-адреса используются для обработки содержимого пакетов, в других случаях на обоих концах соединения могут работать серверы. Средства, реализующие NAT в системе Linux, обеспечивают поддержку некоторых протоколов, но если вы используете видеоконференции или средства шифрования, то при обмене с Internet через NAT-маршрутизатор могут возникнуть проблемы.

- Несмотря на то что NAT защищает компьютеры локальной сети, не следует думать, что их безопасность гарантирована. Угрозу для ваших компьютеров могут представ-

лять также вирусы и программы типа "троянский конь", попадающие в систему по другим каналам.

Опции iptables для осуществления NAT-преобразования

Средства поддержки NAT в системе Linux содержатся в таблице **nat**, которая уже упоминалась выше. Подобно таблице **filter**, nat содержит три цепочки: PREROUTING, **POSTROUTING** и OUTPUT. Несмотря на совпадение имен, цепочка OUTPUT в таблице nat отличается от одноименной цепочки в таблице **filter**. Для активизации средств NAT надо вызвать две следующие команды:

```
# iptables -t nat -A POSTROUTING -o внешний_интерфейс -j \
  MASQUERADE
I echo ''1'' > /proc/sys/net/ipv4/ip_forward
```

Для загрузки NAT-модуля ядра перед вызовом iptables может потребоваться выполнение команды **modprobe** iptable_nat.

В качестве внешнего интерфейса в первой из двух приведенных команд указывается интерфейс, посредством которого осуществляется соединение с Internet, например **ppp0** или **eth1**. Эта команда указывает Linux на то, что для всего сетевого трафика, проходящего через маршрутизатор, надо выполнить IP-маскировку. Вторая команда разрешает ядру Linux осуществить маршрутизацию (эта команда используется также в маршрутизаторах, не поддерживающих NAT).

При настройке NAT-маршрутизатора обычно включают средства фильтрации пакетов. Несмотря на то что NAT-маршрутизатор надежно защищает компьютеры локальной сети от атаки извне, вам надо защитить сам маршрутизатор, а также ограничить возможности компьютеров локальной сети по установлению соединений с Internet. Даже если компьютер, защищенный NAT-маршрутизатором, используете только вы, не исключено появление на нем вирусов и программ типа "троянский конь", которые могут инициировать нежелательные обращения к внешним узлам. Возможно, вы захотите включить проверку пакетов с учетом состояния, чтобы пресечь попытки перехвата соединений, предпринимаемые из вашей локальной сети. NAT-команды задаются посредством того же сценария, который используется для установки правил брандмауэра.

По возможности не следует запускать на компьютере, выполняющем функции NAT-маршрутизатора, никакие серверы. Если злоумышленник получит контроль над этим сервером, он сможет проникнуть в вашу сеть. Если у вас не хватает средств на приобретение отдельного компьютера, вы можете установить NAT-маршрутизатор на машине устаревшей модели. Для этой цели подойдет даже компьютер 80486.

Перенаправление портов

Бывают ситуации, при которых обращение к одному узлу должно быть перенаправлено на другой узел либо на другой порт того же компьютера. Эта задача решается с помощью перенаправления портов, организуемого с помощью программы iptables.

Задачи, решаемые с помощью перенаправления портов

Перенаправление портов может потребоваться в следующих ситуациях.

- Если вы перемещаете сервер с одного компьютера на другой, но по каким-либо причинам не можете изменить конфигурацию сервера DNS. Перенаправление портов позволяет разрешить эту проблему.
- Если вы хотите, чтобы один и тот же сервер отвечал на обращения по разным портам. Вы можете настроить систему для перенаправления запроса с одного порта на другой. Некоторые серверы могут принимать обращение через несколько портов, в этом случае перенаправление портов не требуется.
- Если вы хотите установить в сети, защищенной NAT-маршрутизатором, сервер и обеспечить доступ к нему извне. В этом случае вы можете настроить NAT-маршрутизатор для перенаправления трафика, связанного с одним из внешних портов, внутренней системе.

Необходимость в пренаправлении запросов компьютерам локальной сети часто возникает при использовании NAT-маршрутизатора. При этом уровень защиты несколько снижается, так как вы предоставляете внешним узлам доступ к одному из компьютеров (точнее, к одному из портов да нем). Подобное решение допускает выполнение лишь одного внутреннего сервера, ожидающего обращения через определенный порт. Если вы хотите запустить во внутренней сети два сервера одного типа, доступных извне (например, два Web-сервера), то один из них должен использовать нестандартный порт либо эти серверы должны быть представлены на внешних компьютерах с разными IP-адресами.

Опции iptables для перенаправления портов

Обеспечить перенаправление портов на компьютере под управлением Linux, поддерживающем NAT, можно различными способами. Один из них состоит в использовании iptables. Соответствующая команда имеет следующий вид:

```
# iptables -t nat -A PREROUTING -p tcp -i external-interface \
  --destination-port port-num -j DNAT --to dest-addr:port-num
```

Ниже описаны компоненты данной команды.

- Опция, определяющая таблицу NAT (`-t nat`).
- Опция -A PREROUTING, указывающая на то, что изменения должны вноситься в состав пакета перед выполнением маршрутизации. Базовые средства NAT применяются после маршрутизации, но перенаправление портов предшествует маршрутизации.
- Опция, которая задает перенаправление TCP-портов (`-p tcp`).
- Правило, применяемое к пакетам, направленным через внешний интерфейс (`-i` *внешний_интерфейс*) по конкретному порту (`-destination-port` *номер_порта*).
- Опция -j DNAT, указывающая на то, что вместо NAT источника (SNAT) выполняется NAT назначения (DNAT).

- Опция `--to адрес_назначения:номер_порта`, сообщающая, что пакет должен быть направлен на указанный адрес с использованием указанного номера порта. В качестве адреса назначения можно указать, например, адрес 192.168.9.33, а номер порта может быть равен 80. Заметьте, что номер порта, задаваемый посредством опции `--to`, не обязательно должен совпадать с номером порта, представляющим значение опции `--destination-port`.

В случае необходимости вы можете задать команды перенаправления портов для каждого из серверов, выполняющихся во внутренней сети. Как и при определении базовой конфигурации NAT и правил брандмауэра, соответствующие команды желательно включить в состав сценария, вызываемого при загрузке системы.

Существуют и другие инструменты, предназначенные для перенаправления портов. Такую возможность предоставляет, например, суперсервер `xinetd`. Поскольку `xinetd` выполняется как пользовательский процесс, он не позволяет добиться такой эффективности, как средства перенаправления, реализованные в составе ядра.

Протоколирование хода обработки пакетов

Команды и опции `iptables`, рассмотренные в данной главе, не предполагали протоколирование действий по преобразованию пакетов. Однако в некоторых случаях необходимо иметь информацию о блокированных попытках доступа к важным портам или о пакетах, отвергнутых в результате проверки с учетом состояния. Такая информация поможет выявить попытки взлома системы, предпринимаемые извне.

ВНИМАНИЕ Несмотря на то что файлы протоколов представляют собой важный источник информации, протоколирование существенно снижает производительность маршрутизатора и делает систему более уязвимой для атак, предпринимаемых с целью вывода служб из строя. Если злоумышленник знает, что результаты выполнения некоторых операций записываются в файл протокола, он будет передавать большое количество пакетов, получая которые система станет интенсивно выполнять протоколирование своей работы. В результате размеры файлов протоколов будут неограниченно расти, что может привести к переполнению диска. Поэтому при протоколировании надо соблюдать осторожность. Желательно разместить файлы протоколов в отдельном разделе, чтобы остальные программы не пострадали, если процесс протоколирования выйдет из под контроля.

В программе `iptables` предусмотрено специальное действие LOG, управляющее протоколированием. В отличие от других действий, действие LOG не приводит к прекращению дальнейшей проверки; если пакет соответствует правилу, в котором указано данное действие, ядро продолжает поверку, используя остальные правила текущей цепочки. Действие LOG позволяет решить следующие задачи.

- С помощью действия LOG можно протоколировать события, состоящие в появлении пакетов, которые не удовлетворяют другим правилам. Например, вы можете включить правило для записи информации о тех пакетах, которые не прошли проверку с учетом состояния.

СОВЕТ ✔ Протоколирование фактов появления пакетов, которые вы не собираетесь отвергать, может быть удобным средством отладки, так как файл протокола позволяет убедиться, что эти пакеты поступают на ваш компьютер. Тот же результат можно получить с помощью других инструментов, например, программы сбора пакетов, но в некоторых случаях файлы протоколов удобнее использовать.

- Если вы установили политику по умолчанию DENY или REJECT, вы можете включить правило протоколирования в конце цепочки и получать таким образом информацию о пакетах, по умолчанию отвергаемых системой.
- Если в вашей системе установлена политика по умолчанию ACCEPT, вы можете получать информацию об отвергнутых пакетах, продублировав каждое запрещающее правило аналогичным правилом, в котором действие DENY или REJECT заменено на LOG.

В качестве примера рассмотрим следующие правила брандмауэра, для которого установлена политика по умолчанию ACCEPT. Эти правила предназначены для блокирования попыток обмена с сетью 172.24.0.0/16; информация об отвергнутых пакетах записывается в файл протокола.

```
# iptables -A INPUT -s 172.24.0.0/16 -j LOG
# iptables -A OUTPUT -d 172.24.0.0/16 -j LOG
# iptables -A INPUT -s 172.24.0.0/16 -j DROP
# iptables -A OUTPUT -d 172.24.0.0/16 -j DROP
```

Первые две команды совпадают с двумя последними, за исключением того, что вместо действия DROP в них указано действие LOG. Второе и третье правила можно поменять местами, при этом результаты не изменятся. Между правилами, предусматривающими действия LOG и DROP, можно включить дополнительные правила, но при этом становится менее очевидно, что данные правила связаны между собой.

В результате протоколирования в файл /var/log/messages записываются сведения, подобные приведенным ниже.

```
Nov 18 22:13:21 teela kernel: IN=ethO OUT=
MAC=00:05:02:a7:76:da:00:50:bf:19:7e:99:08:00 SRC=192.168.1.3
DST=192.168.1.2 LEN=40 TOS=OxlO PREC=0xOO TTL=64 ID=16023 DF
PROTO=TCP SPT=4780 DPT=22 WINDOW=32120 RES=OxOO ACK URGP=0
```

В состав записи входят следующие данные.

- Дата и время. Первый компонент записи сообщает время получения пакета.
- Имя системы. В данном примере компьютер имеет имя teela.
- Входной интерфейс. Поле IN=ethO указывает на то, что пакет был получен через интерфейс eth0.
- Выходной интерфейс. Данный пакет является входным, поэтому поле OUT= отсутствует в составе записи.
- MAC-адрес. В поле MAC= указываются два MAC-адреса: локальной и удаленной систем.

- **IP-адреса источника и назначения.** Поля SRC= и DST= содержат соответственно IP-адреса источника и назначения.
- **Порты источника и назначения.** Поля SPT= и DPT= содержат соответственно порты источника и назначения.
- **Информация о пакете.** Остальные поля предоставляют дополнительные сведения о пакете, в частности, его длину (LEN=), время жизни (TTL=) и другие данные.

При определении правила LOG могут быть заданы дополнительные опции, которые позволяют указать, какие сведения должны быть записаны в файл протокола. Наиболее часто применяется опция `--log-prefix` *префикс*. Она позволяет задать строку длиной до 29 символов, которая дает возможность идентифицировать правило, вызвавшее появление этой записи.

Резюме

Программа `iptables` часто применяется для создания брандмауэров, настройки средств NAT, организации перенаправления портов и протоколирования хода обработки пакетов. Часто различные способы обработки пакетов используются совместно. Так, например, на одном компьютере могут быть реализованы брандмауэр, NAT-маршрутизатор и протоколирование хода обработки. Каждый вызов `iptables` задает отдельное правило, но для решения большинства задач необходимо определить несколько правил. Поэтому последовательность вызовов `iptables` организуется в виде сценария.

Глава 26

Организация виртуальной частной сети

Одна из проблем передачи данных в Internet связана с шифрованием информации. Во многих часто применяющихся протоколах, например Telnet и FTP, не предусмотрено кодирование информации. Данные, в том числе пользовательское имя и пароль, передаются в незашифрованном виде. Такая ситуация может считаться приемлемой в локальной сети, где администратор имеет возможность контролировать действия пользователей, но в Internet, где между передающим и принимающим узлами находится несколько маршрутизаторов, передавать важную информацию с помощью подобных протоколов недопустимо.

ВНИМАНИЕ Не следует считать, что в локальной сети информация полностью защищена. Не исключено, что взломщик получит контроль над компьютером сети и использует его для дальнейшего сбора информации. Применение протоколов, предусматривающих кодирование данных, позволяет исправить ситуацию. Для повышения степени защиты локальной сети можно использовать систему Kerberos, описанную в главе 6.

Иногда у пользователей возникает необходимость обратиться к ресурсам локальной сети с удаленных компьютеров. Некоторые из них работают дома или в дороге на портативных компьютерах. Один из способов, позволяющих обеспечить работу удаленных пользователей, не подвергая данные существенному риску, состоит в организации *виртуальной частной сети* (VPN — Virtual Private Network). Такая сеть предоставляет удаленному пользователю доступ к ресурсам так, как будто он работает в пределах локальной сети. Клиент и сервер VPN создают виртуальные сетевые интерфейсы и связывают их через Internet, причем данные передаются в закодированном виде. Таким образом, VPN позволяет связать удаленные компьютеры или удаленные сети с локальной сетью. В данной главе приводятся основные сведения, касающиеся конфигурации средств VPN, а также рассматриваются протоколы VPN: PPTP и FreeS/WAN, обеспечивающие работу.

Использование VPN

VPN позволяет расширить локальную сеть за счет взаимодействия с внешними компьютерами. Очевидно, что если локальная сеть подключена к Internet, внешние пользователи могут обращаться к ней без VPN. Однако VPN имеет ряд преимуществ перед обычными типами сетевого обмена.

- **VPN создает иллюзию локального доступа.** Во многих локальных сетях используются средства защиты против нежелательных обращений извне. Так, например, доступ к сетям и отдельным компьютерам ограничивается с помощью брандмауэров, кроме того, для контроля взаимодействия применяются также TCP Wrappers и средства, предоставляемые суперсервером и различными серверными программами. VPN позволяет обращаться к компьютеру так, как будто взаимодействие происходит в пределах локальной сети, что упрощает настройку многих серверов.

- **VPN обеспечивает шифрование данных, передаваемых посредством протоколов, не предусматривающих кодирование.** Если бы средства VPN не обеспечивали шифрование данных, то виртуальную сеть, созданную с их помощью, нельзя было бы назвать частной. Кодируя информацию, передаваемую посредством таких протоколов, как NFS и Telnet, VPN существенно упрощает обмен важными данными по Internet. (Следует заметить, что такой обмен организуется чрезвычайно сложными средствами, простым он кажется лишь с точки зрения клиента или сервера.) Клиенты и серверы, взаимодействующие в рамках VPN, не надо настраивать специальным образом; всю работу по обеспечению защиты выполняют средства VPN. Очевидно, что кодирование передаваемых данных можно обеспечить за счет применения специальных протоколов. Если вам приходится решать ограниченное число задач, для решения которых необходимо передавать кодированные данные, имеет смысл использовать другие протоколы, обеспечивающие секретность информации. Настроить их проще, чем VPN.

VPN чаще всего применяется для организации взаимодействия между несколькими удаленными офисами. Предположим, что офисы вашей компании расположены в Бостоне и Сан-Франциско. Объединить их сети можно посредством VPN, при этом сотрудники смогут обращаться к серверам в разных сетях, не подвергая опасности передаваемые данные. Принцип организации VPN условно показан на рис. 26.1. В роли VPN-маршрутизатоов, показанных на данном рисунке, могут выступать обычные маршрутизаторы, NAT-преобразователи или компьютеры, на которых реализованы брандмауэры. В дополнение к обычным функциям маршрутизации VPN-маршрутизаторы также создают защищенные соединения, по которым передаются данные.

Несмотря на то что на рис. 26.1 показано соединение лишь между двумя сетями, технология VPN позволяет объединить в виртуальную сеть любое количество локальных сетей.

Еще одно назначение VPN состоит в предоставлении отдельным пользователям доступа к сетям. Пользователь может подключить свой домашний или портативный компьютер к сети посредством линии с широкой полосой пропускания или с помощью коммутируемого соединения. В этом случае VPN-маршрутизатор работает непосредственно с удаленным компьютером пользователя. Удаленный компьютер также является VPN-маршрутизатором, но он обрабатывает только собственный трафик. Подобная конфигурация показана на рис. 26.2.

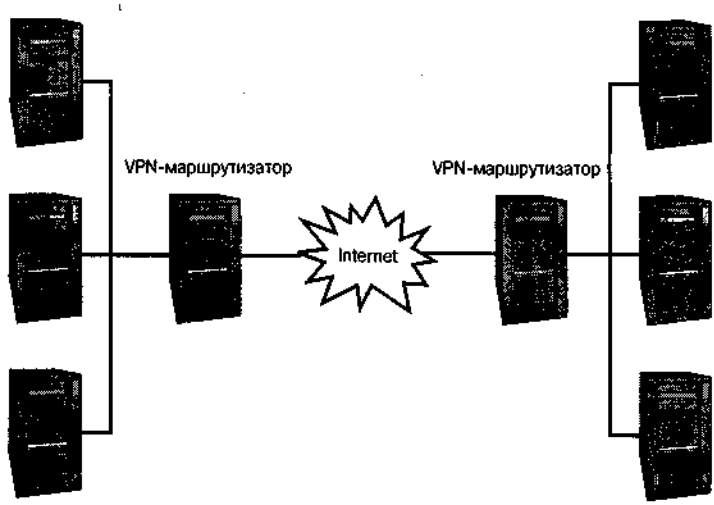


Рис. 26.1. VPN реализуются с помощью маршрутизаторов, которые имеют возможность кодировать данные, направленные на некоторые компьютеры или в некоторые сети

Принимая решение о создании VPN, необходимо учитывать пропускную способность линий. Для подключения нескольких удаленных сетей к центральной сети необходимо располагать линиями, обеспечивающими большую скорость передачи, способными удовлетворить запросы пользователей. Многие протоколы, которые разрабатывались для применения в локальных сетях, генерируют большие объемы данных. Они нормально работают в среде Ethernet, обеспечивающей скорость обмена 100 Мбод, но если инфор-

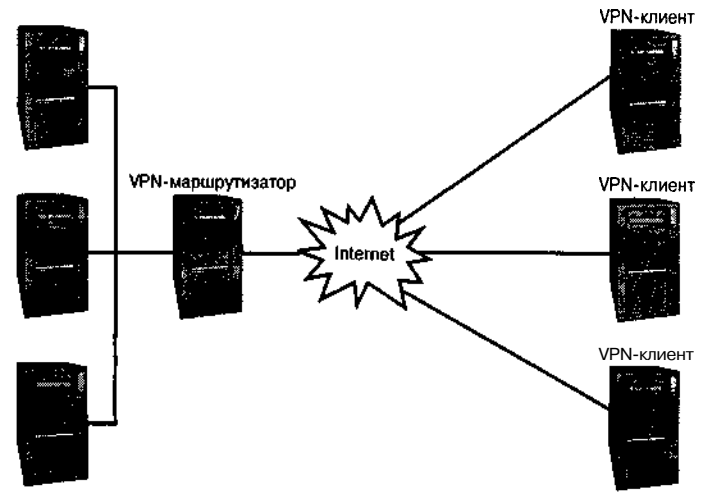


Рис. 26.2. VPN позволяет подключить к сети отдельные удаленные компьютеры

мацию придется передавать по линии T1 с пропускной способностью 1,5 Мбод, производительность станет недопустимо низкой. Если же сети удаленных офисов подключены через ADSL-соединения, вам необходимо учитывать, что такие соединения асимметричны по своей природе. Если пропускная способность в одном направлении может достигать 600-1500 Кбод, то в другом направлении скорость передачи данных не будет превышать 100–300 Кбод. Еще хуже обстоит дело с индивидуальными пользователями, компьютеры которых часто подключаются к Internet по коммутируемой линии, пропускная способность которой не превышает 56 Кбод.

Даже при использовании линий с достаточной пропускной способностью технология VPN имеет свои недостатки. Если VPN реализована некорректно, она может создавать реальную угрозу безопасности виртуальной сети. Предположим, что удаленный пользователь взаимодействует с локальной сетью компании посредством VPN и локальная сеть надежно защищена брандмауэром. Если взломщик сумеет получить контроль над компьютером пользователя, он сможет воспользоваться им для дальнейшего проникновения в сеть компании.

Еще одна проблема, возникающая при использовании VPN, состоит в том, что программные средства очень сложно настроить. Поэтому, если ваши потребности в передаче зашифрованной информации ограничены, вам, возможно, имеет смысл использовать вместо VPN какой-нибудь из протоколов, обеспечивающих кодирование данных, например SSH.

Инструменты, предназначенные для организации VPN

В настоящее время отсутствуют стандартные инструментальные средства, позволяющие создать VPN. Стандарты, регламентирующие работу VPN, находятся в стадии разработки. Ниже приведены три наиболее часто употребляемых инструмента, предназначенные для создания VPN.

- **PPTP** (Point-to-Point Tunneling Protocol — протокол межузлового туннелирования) был создан консорциумом PPTP Forum, в который входят несколько компаний, занимающихся разработкой сетевых средств. Протокол PPTP часто используется для организации взаимодействия сотрудников, работающих дома, с сетями предприятий. Средства поддержки PPTP входят в состав последних версий Windows. Существует также PPTP-сервер для Linux; он называется PoPToP (`http://poptop.lineo.com`).
- **FreeS/WAN**. Проект FreeS/WAN (`http://www.freeswan.org`) посвящен созданию VPN-инструмента для Linux. Этот инструмент распространяется в исходных кодах. Он очень популярен для организации VPN, в которые входят компьютеры под управлением Linux.
- SSH. Возможность протокола SSH поддерживать туннелирование соединений посредством других протоколов также может использоваться для создания VPN.

В данной главе будут рассматриваться первые два из описанных выше подходов к созданию VPN. PPTP — очень популярный инструмент. Его очень удобно использовать в тех

случаях, когда Windows-клиент должен непосредственно подключаться к VPN-маршрутизатору. Данное средство также реализовано для других операционных систем; существуют даже специальные устройства, называемые *коммутаторами удаленного доступа* (remote access switch). Инструмент FreeS/WAN не пользуется большой популярностью в операционных средах, отличных от Linux. Однако он часто применяется для организации VPN в тех случаях, когда роль VPN-маршрутизаторов выполняют компьютеры под управлением Linux.

Настройка PPTP в системе Linux

Поскольку PPTP не разрабатывался специально для Linux, чтобы инсталлировать соответствующие средства на компьютере под управлением Linux, необходимо приложить определенные усилия. Сервер PoPToP взаимодействует с PPP-демоном pppd. Для обеспечения безопасности система должна уметь шифровать данные, но соответствующие средства в программе pppd отсутствуют. Поэтому демон pppd необходимо заменить его расширенной версией. **PPTP-клиенты** созданы как для Linux, так и для Windows; очевидно, что они настраиваются по-разному.

Инсталляция PoPToP

Инструмент PoPToP поставляется в составе некоторых версий Linux, например Debian и Mandrake. Соответствующий пакет чаще всего имеет имя `pptpd` или `pptpd-server`. Пакет, поставляемый с системой Linux, обычно проще настраивать, чем универсальный пакет, распространяемый по Internet. Если в вашем дистрибутивном пакете Linux нет инструмента PoPToP, вы можете скопировать его с Web-узла PoPToP, расположенного по адресу `http://poptop.lineo.com`.

По умолчанию средства PoPToP в системе Linux не обеспечивают должного уровня защиты при организации VPN. Причина в том, что PPTP применяет специальные средства кодирования PPP, которые не поддерживаются стандартной программой pppd. В частности, работа PPTP базируется на использовании протокола **MPPE** (Microsoft Point-to-Point Encryption — межузловое кодирование Microsoft). Для поддержки кодирования вам надо инсталлировать **MPPE-дополнения** для стандартной программы pppd и для ядра Linux. Этот процесс будет описан далее в данной главе.

Установка конфигурации сервера PoPToP

После инсталляции пакета PoPToP вам надо активизировать его. Для этого выполните следующие действия.

1. Отредактируйте файл `/etc/ppp/options`. Этот файл управляет работой программы pppd, которая поддерживает соединение между VPN-маршрутизатором и удаленной системой PPTP. Файл `/etc/ppp/options` должен содержать записи наподобие приведенных ниже.

```
debug
    name имя_сервера
    auth
    require-chap
```

```
proxyarp
192.168.1.1:192.168.1.100
```

Большинство из этих записей необходимо для работы PPTP. Последняя строка может отсутствовать; она задает адрес, используемый VPN-маршрутизатором в локальной сети (192.168.1.1), и адрес, присваиваемый VPN-клиенту (192.168.1.100). Если вы не зададите эту строку, будет использоваться IP-адрес, указанный в конфигурации VPN-клиента. В данном случае имя сервера — это доменное имя VPN-сервера.

2. Укажите в файле /etc/ppp/chap-secrets имя пользователя и пароль, которые вы хотите использовать для регистрации. В приведенном ниже примере задано имя пользователя **vpn1** и пароль **vpnpass**.

```
vpn1 * vpnpass *
```

ВНИМАНИЕ | Пароль хранится в файле /etc/ppp/chap-secrets в незакодированном виде, поэтому вам необходимо принять меры для защиты этого файла. Владельцем его должен быть пользователь root и право чтения файла должен иметь только его владелец. Если злоумышленник получит контроль над сервером PoPToP, он сможет прочитать этот файл. По этой причине на компьютере, выполняющем функции VPN-маршрутизатора, должно присутствовать как можно меньше серверов.

3. Найдите в файле /etc/inittab ссылку на pptpd и закомментируйте соответствующую запись, включив в начало строки символ #. Затем введите команду telinit Q, чтобы внесенные изменения были учтены. В результате вы получите возможность вручную запустить pptpd и протестировать конфигурацию данной программы. После создания конфигурации, пригодной для работы, удалите символ комментариев из соответствующей строки файла /etc/inittab или запустите сервер другим способом.

4. От имени пользователя root введите команду pptpd, запустив тем самым сервер.

В результате выполненных действий сервер будет запущен и PPTP-клиент сможет устанавливать взаимодействие с системой. Поскольку средства шифрования не доступны, для установления соединения вам надо также отключить шифрование и на стороне клиента. В следующем разделе рассказывается о том, как разрешить кодирование данных для PoPToP.

ВНИМАНИЕ | Несмотря на то что соединение с PoPToP без поддержки кодирования позволяет проверить конфигурацию системы, это соединение нельзя использовать для реальной работы. Основная цель VPN состоит в том, чтобы обеспечить шифрование передаваемых данных, поэтому при отключении кодирования средства VPN будут бесполезны.

Работой PPTP управляют также опции, содержащиеся в файле который обычно располагается в каталоге /etc или /etc/ppp. /etc/ppp. Некоторые из этих опций описаны ниже.

- debug. Данная опция сообщает PoPToP о том, что в файл протокола должны быть записаны дополнительные данные. Они могут понадобиться в том случае, если при установлении соединения возникают проблемы.
- localip. Клиент PPTP использует два IP-адреса: один — для локальной сети, второй — для удаленного клиента. Локальные IP-адреса можно задать с помощью опции localip. В качестве значения опции задается список адресов, разделенных запятыми, или диапазон адресов. Например, опция `localip 192.168.9.7,192.168.9.100-150` задает адрес 192.168.9.7 и все адреса в диапазоне от 192.168.9.100 до 192.168.9.150. Убедитесь, что другие компьютеры в вашей локальной сети не используют эти адреса.
- remoteip. Данная опция задает IP-адреса, используемые удаленными клиентами. Эти адреса обычно принадлежат диапазону адресов, используемых во внутренних сетях. IP-адреса задаются в таком же формате, как и для опции localip.
- listen. Указав в качестве значения данной опции IP-адрес, связанный с одним интерфейсом, можно сообщить программе pptpd о том, что она должна принимать обращения только через этот интерфейс. По умолчанию PoPToP принимает обращения через все интерфейсы.

Обеспечение кодирования данных

PoPToP использует программу pppd, которая, в свою очередь, использует средства ядра. В частности, PoPToP требует, чтобы демон pppd поддерживал кодирование, а pppd требует, чтобы средства поддержки кодирования присутствовали в ядре Linux. Поэтому для шифрования данных при работе PoPToP необходимо дополнить как pppd, так и ядро системы.

Проще всего решить эту задачу, используя дополненные варианты pppd и ядра Linux. Соответствующие программы можно получить, обратившись по адресу `http://mirror.binarix.com/ppp-mppe/`. Вам необходимо скопировать следующие два файла.

- Ядро Linux. Дополненное ядро Linux содержится в файле, имя которого начинается с kernel, например `kernel-2.4.9-13mppe.i386.rpm`. Некоторые из этих пакетов содержат двоичный код ядра, скомпилированный для конкретного типа системы, а другие — исходный код ядра. Если вы скопируете исходный код, вам надо будет сконфигурировать и скомпилировать его для вашей системы.
- Пакет ppp. Измененный пакет pppd находится в файле `ppp-2.4.1-3mdk.i586.rpm` или в другом файле с подобным именем. Содержимое пакета надо установить вместо файла pppd.

На узле `http://mirror.binarix.com/ppp-mppe/` находятся двоичные программы, скомпилированные для Mandrake, поэтому если вы работаете с этой версией Linux, вы можете воспользоваться кодами, представленными на данном узле. Не исключено, что вы найдете программы, сконфигурированные для другой системы.

Если в вашей системе не используется RPM, вы можете преобразовать форматы пакетов с помощью утилиты alien. Эта программа входит в состав Debian и поддерживает RPM, пакеты Debian и tar-архивы.

Дополненные программы вы также можете получить, обратившись на узел `http://pptpclient.sourceforge.net`. Здесь можно найти клиентские программы, пакеты `ppp-mppe`, представляющие собой программы pppd с поддержкой MPPE, а также модули ядра с поддержкой MPPE.

Если вы по каким-либо причинам не можете или не хотите использовать готовые двоичные коды, вам следует самостоятельно внести изменения в состав демона PPP и ядра системы. Для этого надо скопировать пять файлов.

- Ядро Linux. Исходный код стандартного ядра Linux можно получить, обратившись на узел `http://www.kernel.org`. Использовать стандартное ядро Linux предпочтительнее, чем ядро из дистрибутивного пакета, так как последнее обычно бывает модифицировано, в результате чего внесение изменений может быть затруднено.

- Исходный код pppd. Исходный код демона PPP можно получить по адресу `ftp://cs.anu.edu.au/pub/software/ppp/`.

- OpenSSL. MPPE-дополнения требуют, чтобы в вашей системе были установлены OpenSSL и файлы заголовков OpenSSL. Требуемые данные можно скопировать с узла `http://www.openssl.org`.

- Дополнения к ядру Linux. На узле `http://mirror.binarix.com/ppp-mppe/` надо найти файлы, начинающиеся с `linux` и заканчивающиеся `patch.gz`, например `linux-2.4.16-openssl-0.9.6-bmppe.patch.gz`.

- Дополнения к pppd. Дополнения к программе pppd также доступны по адресу `http://mirror.binarix.com/ppp-mppe/`. Имя соответствующих файлов начинается с ppp и заканчивается `patch.gz`, например `ppp-2.4.1-openssl-0.9.6-mppepatch.gz`.

Для того чтобы исключить несоответствие версий, надо начать с файлов дополнений, а затем подобрать соответствующие версии ядра и пакета pppd. Чтобы дополнить и использовать полученные средства, надо распаковать архивы с исходными кодами ядра и pppd, распаковать файлы дополнений (выполнив команду `gunzip filename.patch.gz`), дополнить исходные коды (cd *каталог_с_исходными_кодами*; patch -p1 < patchfile.patch), сконфигурировать пакеты (make `menuconfig` или make `xconfig` — для ядра Linux и `./configure` — для pppd), скомпилировать пакеты (make `bzImage` и make modules — для ядра Linux и make — для pppd) и инсталлировать пакеты (выполнив команду make `modules\_install` и сконфигурировав LILO для Linux, а также выполнив команду make `install` для pppd).

Независимо от того, используете ли вы дополненные варианты программ или дополнили коды и скомпилировали программы самостоятельно, для того, чтобы обеспечить поддержку кодирования, надо перезагрузить компьютер с указанием нового ядра.

Настройка PPTP-клиента

Если PPTP-клиент предназначен для выполнения в системе Windows, настроить его для работы с PoPToP несложно, так как в системе Windows предусмотрена поддержка PPTP. Для обеспечения работы PPTP-клиента в системе Linux требуется дополнительное программное обеспечение. В любом случае после установления VPN-соединения VPN-

клиент работает так, как будто он является частью локальной сети. Отличие от реальной работы в локальной сети состоит в том, что скорость обмена оказывается гораздо меньшей.

Использование PPTP-клиента в системе Linux

PoPToP — это PPTP-сервер, выполняющийся в системе Linux. Для того чтобы обеспечить взаимодействие компьютера, работающего под управлением Linux, с PoPToP или другим PPTP-сервером, необходим дополнительный программный пакет PPTP-Linux. Его можно получить, обратившись по адресу `http://cag.lcs.mit.edu/~cananian/Projects/PPTP/` или `http://pptpclient.sourceforge.net`. На узле `http://pptpclient.sourceforge.net` содержатся исходные коды PPTP-Linux в виде TAR-архивов и в формате RPM, а также двоичный код для процессоров x86 и Alpha. Вам следует скопировать пакет PPTP-Linux и, если необходимо, скомпилировать его и установить на свой компьютер.

Подобно PoPToP, для поддержки кодирования PPTP-Linux использует программу pppd и средства ядра. Поэтому, чтобы могли установить соединения с шифрованием передаваемых данных, вам надо внести изменения в состав pppd и ядра. Способ внесения необходимых изменений рассматривался в предыдущем разделе. Требуемые для этого инструментальные средства содержатся на узле PPTP-Linux.

В состав пакета PPTP-Linux входит сценарий инсталляции `pptp-command`. Для установки PPTP-Linux необходимо выполнить следующие действия.

1. Запустите сценарий по команде pptp-command.
2. Сценарий выведет список из четырех опций: start, stop, setup и quit. Для активизации процедуры установки введите число 3.
3. Сценарий отобразит список из девяти пунктов, соответствующих вариантам настройки. Введите 2, чтобы выбрать пункт Add a New CHAP secret.
4. Система запросит локальное имя вашей системы. Это имя будет присвоено системе при работе в VPN. Если VPN-маршрутизатор выполняется в системе Windows, вам надо задать имя домена NetBIOS. Например, вы можете указать arbor\\maple, в результате чего ваша система получит имя maple в домене arbor.
5. Система запросит удаленное имя вашей системы. В большинстве случаев можно принять значение по умолчанию (пустую строку). Удаленное имя необходимо только тогда, когда в сети многократно используется одно локальное имя с различными паролями.
6. Система запросит пароль. Этот пароль должен совпадать с паролем, заданным при настройке PoPToP или другого VPN-сервера.
7. Сценарий снова отобразит список из девяти пунктов, позволяющий выбрать вариант настройки. На этот раз вам надо выбрать пункт 5 Add a NEW PPTP Tunnel.
8. Система отобразит список туннелей. Вероятнее всего, этот список будет пустым; в нем будет содержаться только пункт Other. Если вы увидите подходящий вам пункт, выберите его, но в большинстве случаев приходится выбирать Other.

9. Система запросит определение туннеля, в частности имя, IP-адрес VPN-сервера и атрибуты маршрутизации. Атрибуты маршрутизации совпадают с параметрами команды route. Например, выражение add -host 172.19.87.1 gw DEF_GW указывает на то, что система должна использовать адрес 172.19.87.1 в качестве шлюза по умолчанию.

10. Сценарий еще раз выведет список из девяти пунктов. Выберите пункт 7 Configure resolv.conf.

11. Выберите конфигурацию туннеля, созданную на шаге 9. Система запросит DNS-информацию, которая будет помещена в файл /etc/resolv.conf. Введите соответствующие данные.

12. Список из девяти пунктов появится на экране снова. Выберите пункт 8 Select a default tunnel.

13. Система запросит имя туннеля по умолчанию. Выберите туннель, созданный на шаге 9 (или любой другой).

14. При очередном появлении списка из девяти пунктов выберите пункт 9 Quit. Это приведет к завершению программы установки.

После выполнения описанных выше действий программа PPTP-Linux готова к взаимодействию с PPTP-сервером. Для подготовки PPTP VPN-соединения используется сценарий pptp-command. В списке, отображаемом на шаге 3, надо выбрать пункт 1 (start). Программа запросит номер туннеля. После указания этого номера подготовка PPTP VPN-соединения завершится.

Проверить настройку VPN-соединения можно в таблице маршрутизации либо при подготовке к взаимодействию с сервером в системе VPN. Если VPN-сервер не доступен, проверьте VPN-маршрутизатор с помощью утилиты ping. Вы также можете использовать программу traceroute, чтобы выяснить, проходят ли пакеты через VPN-соединение. Если по обычному Internet-соединению пакеты проходят, это означает, что таблица маршрутизации составлена некорректно. Если путь к VPN-системам через VPN PPP-соединение отсутствует, Linux попытайтесь направить пакеты в сеть через обычное соединение.

Использование PPTP-клиента в системе Windows

Очень часто PPTP-клиенты устанавливаются на компьютерах под управлением Windows, которые используют сотрудники, вынужденные часто работать вне офиса. PPTP-клиенты входят в состав систем Windows 9x/Me и Windows NT/2000/XP, но по умолчанию они не инсталлируются. Программное обеспечение PPTP будет работать только при наличии действующего Internet-соединения. Ниже описана процедура запуска PPTP-клиента в системе Windows Me.

1. Дважды щелкните на пиктограмме Add/Remove Programs в окне Control Panel. В результате на экране отобразится диалоговое окно Add/Remove Programs Properties.

2. В окне Add/Remove Programs Properties щелкните на вкладке Windows Setup.

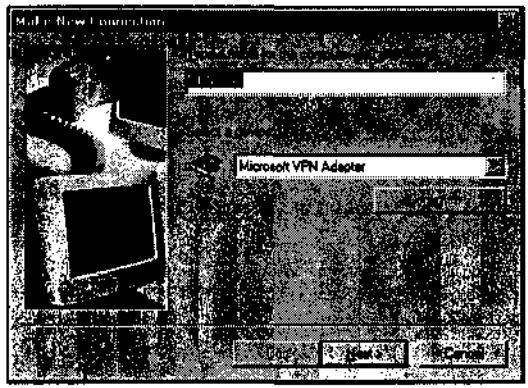


Рис. 26.3. При создании VPN-соединения выбирайте Microsoft VPN Adapter, а не модем, через который устанавливается соединение

3. Дважды щелкните на пункте Communications списка типов компонентов. На экране появится диалоговое окно Communications.

4. В окне Communications выберите пункт Virtual Private Networking.

5. Щелкните на кнопке OK сначала в окне Communications, а затем в окне Add/Remove Programs Properties. В результате в системе Windows будет установлено программное обеспечение PPTP. Если вам будет предложено перезагрузить компьютер, сделайте это.

6. После перезагрузки системы откройте папку Dial-Up Networking в Control Panel.

7. Дважды щелкните на пиктограмме Make New Connection. Вы увидите окно Make New Connection Wizard, показанное на рис. 26.3.

8. Введите имя, идентифицирующее соединение, и выберите устройство Microsoft VPN Adapter (см. рис. 26.3).

9. Щелкните на кнопке Next. В окне Make New Connection отобразится поле редактирования, в котором надо ввести имя или IP-адрес сервера VPN.

10. Щелкните на кнопке Next. Система оповестит вас о том, что новое устройство создано. Щелкните на кнопке Finish.

В результате выполненных действий в окне Dial-Up Networking появится новая пиктограмма. Если вы дважды щелкнете на ней, Windows отобразит диалоговое окно Connect To, показанное на рис. 26.4. Укажите в нем имя пользователя и пароль и, если необходимо, измените имя или IP-адрес сервера VPN. После щелчка на кнопке Connect система установит соединение (это может занять несколько секунд). После этого в вашей системе появится дополнительный IP-адрес, соответствующий сети сервера VPN. Вы можете обращаться к компьютерам этой сети как к узлам локальной сети, например, просматривать сетевое окружение, пользуясь средствами My Network Places (в ранних версиях Windows

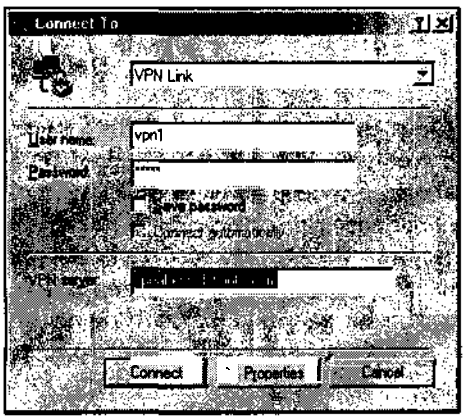


Рис. 26.4. Диалоговое окно Connect To предоставляет контроль над VPN-соединением

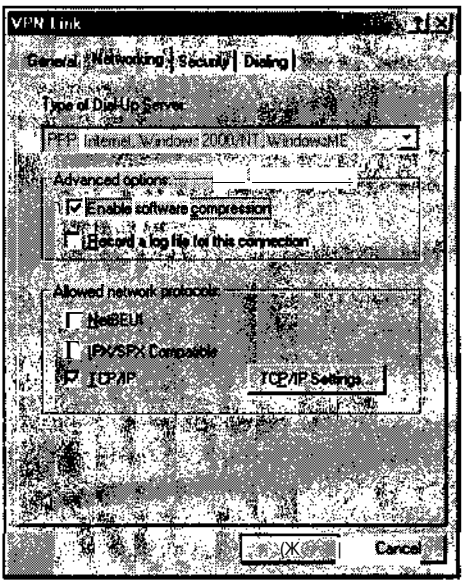


Рис. 26.5. С помощью средств настройки клиента задаются параметры VPN-взаимодействия, используемые по умолчанию

использовалось название Network Neighborhood). Ресурсы сети будут доступны вам как локальные ресурсы. Однако не забывайте, что физически сеть не является локальной, поэтому скорость обмена с узлами этой сети будет значительно меньше, чем у компьютеров, реально подключенных к ней.

Как показано на рис. 26.4, в диалоговом окне Connect To флажок опции сохранения пароля не установлен. Если вы установите его, то, установив флажок опции Connect Automatically, вы укажете Windows, что соединение надо инициировать при загрузке системы. После щелчка на кнопке Properties будут доступны дополнительные средства настройки. На экране отобразится диалоговое окно, показанное на рис. 26.5, имя которого совпадает с именем VPN-соединения. Чаще всего используются опции, представленные на вкладках Networking и Security. С помощью элементов, расположенных на вкладке Networking, вы можете выполнять сжатие данных и протоколировать ход сеанса, кроме того, можно указать, какие протоколы должны поддерживаться средствами VPN. Щелкнув на кнопке TCP/IP Settings, вы можете указать системе на необходимость получать IP-адрес и адреса серверов DNS у сервера PPTP. На вкладке Security указываются имя пользователя, пароль и имя домена NetBIOS. С помощью элементов, расположенных на ней, вы можете разрешить или запретить кодирование пользовательского имени и пароля.

Настройка сервера FreeS/WAN

Сервер FreeS/WAN выполняет те же функции, что и сервер PPTP, но он ориентирован на работу в системе Linux. Как и сервер PPTP, FreeS/WAN поддерживает защищенные соединения в незащищенном сетевом окружении, например в Internet. Первое, что надо

сделать для обеспечения работы FreeS/WAN, — это получить и инсталлировать программное обеспечение. Затем можно настроить систему и устанавливать соединения.

FreeS/WAN — очень сложный пакет, и в данной главе рассматриваются лишь некоторые его средства. Дополнительную информацию вы найдете в документации на FreeS/WAN, в частности в руководстве по настройке (`http://www.freeswan.org/freeswan_trees/freeswan-1.91/doc/config.html`).

Инсталляция FreeS/WAN

FreeS/WAN иногда поставляется с версиями Linux SuSE и Mandrake. Если в вашей системе нет пакета FreeS/WAN, скопируйте его с Web-узла FreeS/WAN, расположенного по адресу `http://www.freeswan.org`. На нем, а точнее на сервере FTP (`ftp://ftp.xs4all.nl/pub/crypto/freeswan/`), на который ссылаются Web-страницы, находятся исходные коды программ. Для поддержки FreeS/WAN требуются нестандартные средства ядра, поэтому при установке данного сервера вам надо перекомпилировать ядро. Если FreeS/WAN входит в состав дистрибутивного пакета Linux, это значит, что необходимые изменения ядра уже выполнены. При изложении материала данного раздела предполагается, что вы скопировали исходные коды FreeS/WAN с Web-сервера.

Для компиляции исходных кодов FreeS/WAN вам понадобятся следующие компоненты.

- Стандартные инструменты разработки. Для подготовки сервера FreeS/WAN нужны такие инструменты, как GCC, make, набор библиотек и файлов заголовков. Эти компоненты по умолчанию устанавливаются при инсталляции большинства версий Linux.
- Исходные коды ядра. При компиляции FreeS/WAN автоматически вносятся изменения в исходные коды Linux, расположенные в каталоге `/usr/src/linux`. Если вы собираетесь изменить конфигурацию сервера, сделайте это перед инсталляцией FreeS/WAN.
- Библиотека GMP. FreeS/WAN использует библиотеку GMP (`http://www.swox.com/gmp/`). Данная библиотека поставляется в составе многих дистрибутивных пакетов Linux. Если она отсутствует, вам следует установить ее.
- Библиотека ncurses. При настройке FreeS/WAN может потребоваться библиотека ncurses. Она не является необходимым компонентом, но наличие ее желательно. Эта библиотека часто используется, поэтому не исключено, что она установлена в вашей системе.

Для инсталляции FreeS/WAN выполните следующие действия.

1. Убедитесь, что на вашем компьютере установлены все описанные выше компоненты.
2. Распакуйте пакет FreeS/WAN, выбрав для размещения его содержимого произвольный каталог, например подкаталог каталога `usr/src`. После распаковки не копируйте и не перемещайте каталог `freeswan-`*`версия`*, так как это может повредить символьные ссылки.

3. Сделайте текущим каталог с исходными кодами FreeS/WAN и введите одну из команд, предназначенных для конфигурирования пакета, внесения изменений в ядро Linux и создания FreeS/WAN. Команда make oldgo ориентирована на использование существующей конфигурации ядра и установок FreeS/WAN, заданных по умолчанию, команда make ogo вызывает make config, команда make menugo использует для настройки ядра make menuconfig, а команда make xgo вызывает make xconfig. Использование последних трех команд позволяет изменить конфигурацию ядра.

4. Введите команду make kinstall для создания ядра. В результате будут созданы ядро и необходимые модули, а также вызвана команда make modules_install для инсталляции модулей.

5. Измените конфигурацию LILO, GRUB или другого инструмента, используемого для загрузки ядра Linux. Вам надо скопировать файл ядра из каталога /usr/src/linux/arch/architecture-code/boot, редактировать файл /etc/lilo.conf (или другой конфигурационный файл) и ввести команду lilo (или другую команду загрузки).

6. Перезагрузите компьютер. В процессе перезагрузки следите за тем, чтобы ядро системы было указано правильно.

С этого момента ядро вашей системы может поддерживать FreeS/WAN. В процессе установки должен быть создан файл /etc/ipsec.secrets, содержащий ключи кодирования. Этот файл будет использоваться в дальнейшем, сейчас же важно убедиться, что он есть в наличии и содержит некоторые ключи (ключи выглядят как наборы шестнадцатеричных цифр).

Для того чтобы можно было использовать средства FreeS/WAN, настройте как минимум два компьютера, принадлежащих различным сетям. В большинстве случаев обе системы выполняют роль маршрутизаторов, но, если понадобится, вы можете инсталлировать FreeS/WAN на отдельном компьютере, который должен взаимодействовать с удаленной сетью.

Редактирование конфигурационных файлов

FreeS/WAN использует два конфигурационных файла: /etc/ipsec.secrets и /etc/ipsec.conf. Эти файлы предназначены для различных целей. В файле /etc/ipsec.secrets содержатся ключи кодирования, а в файле /etc/ipsec.conf — опции общего назначения.

Создание ключей

Как было сказано ранее, при создании FreeS/WAN должен быть создан файл /etc/ipsec.secrets, содержащий ключи кодирования. Если этот файл не был создан или если ключи в нем отсутствуют, сгенерируйте ключи с помощью команды

```
# ipsec rsasigkey 128 > /root/rsa.key
```

Эта команда создает 128-битовый ключ и помещает его в файл /root/rsa.key. Указав значение параметра, отличающееся от приведенного в данном примере, вы можете сгенерировать ключ другой длины. Данные, полученные в результате выполнения этой

команды, нельзя непосредственно использовать. В начало файла надо включить следующую строку:

```
: RSA {
```

Перед и после RSA обязательно должны быть пробелы. Кроме того, в конец файла надо включить как минимум один пробел, указав за ним закрывающую фигурную скобку (}). Полученный результат надо скопировать в файл `/etc/ipsec.secrets`. Описанные выше действия надо выполнить на обоих VPN-маршрутизаторах, реализованных с помощью FreeS/WAN.

В составе данных, сгенерированных посредством `ipsec rsasigkey`, содержится закомментированная строка, начинающаяся с `#pubkey=`. В ней указан открытый, или общий, ключ. Этот ключ надо передать системе, с которой должна взаимодействовать данная система.

Установка опций в файле ipsec.conf

В большинстве случаев при инсталляции FreeS/WAN создается файл `/etc/ipsec.conf`. Установки по умолчанию, как правило, не обеспечивают выполнение сервером требуемых функций, но содержимое этого файла можно использовать как базу для дальнейшей настройки системы. В файле `/etc/ipsec.conf` содержатся три основных раздела: `config setup`, `conn %default` и `conn remotename`.

Установка локальных опций

В разделе `config setup` содержатся локальные опции. В файле `/etc/ipsec.conf`, создаваемом по умолчанию, этот раздел имеет следующий вид:

```
config setup
  # THIS SETTING MUST BE CORRECT or almost nothing will work;
  # %defaultroute is okay for most simple cases.
  interfaces=%defaultroute
  # Debug-logging controls: "none" for (almost) none, "all" \
for lots.
  klipsdebug=none
  plutodebug=none
  # Use auto= parameters in conn descriptions to control \
startup actions.
  plutoload=%search
  plutostart=%search
  # Close down old connection when new one using same ID shows \
up.
  uniqueids=yes
```

Наиболее важный компонент данного раздела — опция `interfaces`, которая сообщает FreeS/WAN о том, какие интерфейсы следует использовать для поддержки VPN-соединений. Значение по умолчанию `%defaultroute` указывает на то, что FreeS/WAN должен использовать маршрут по умолчанию. Однако вы можете указать конкретные интерфейсы. В следующем примере опция `interfaces` задает использование интерфейсов `eth0` и `ppp1`:

```
interfaces="ipsec0=eth0 ipsec1=ppp1"
```

Опции klipsdebug и plutodebug задают протоколирование функций KLIPS (Kernel IP Security — IP-защита ядра) и демона Pluto. Демон Pluto является частью пакета FreeS/WAN и поддерживает обмен ключами. Если в процессе работы возникают проблемы, вам надо задать для этих опций значение all.

Pluto может загружать соединения в память или автоматически запускать их при запуске FreeS/WAN. Опции plutoload и plutostart показывают, над какими соединениями надо выполнять соответствующие действия. В большинстве случаев можно принять значения данных опций по умолчанию, но, возможно, вы захотите указать лишь некоторые соединения; для этого надо задать их имена.

Установка опций, используемых по умолчанию для описания соединений

Отдельные соединения описываются в разделах, которые начинаются с ключевого слова conn. Наряду с реальными соединениями FreeS/WAN поддерживает соединение %default. В разделе, соответствующем этому соединению, обычно содержатся следующие опции.

- keyingtries. Если соединение установить не удалось, FreeS/WAN предпримет новую попытку. Значение 0 данной опции указывает на то, что попытки установить соединение должны продолжаться бесконечно. Если вы хотите ограничить число попыток, укажите в качестве значения опции keyingtries конкретное число.
- authby. По умолчанию применяется метод аутентификации, задаваемый значением authby=rsasig. Согласно этому методу, для аутентификации должны использоваться ключи RSA. Существует другой способ аутентификации, но в данной главе он рассматриваться не будет.

Помимо приведенных выше опций, в данный раздел можно включить опции, которые будут рассматриваться в следующем разделе. Если вы обнаружите, что одна и та же опция содержится в описании нескольких соединений, перенесите ее в раздел по умолчанию. При этом снижается вероятность появления ошибок, а размеры конфигурационного файла уменьшаются.

Установка удаленных опций, ориентированных на конкретные системы

Каждое соединение требует настройки, для выполнения которой надо изменить содержимое соответствующего раздела conn. За ключевым словом conn следует имя соединения, а затем — опции. В начале строки, содержащей опцию, должен стоять хотя бы один пробел. Многие из опций, включаемых в раздел conn, определяют сетевые интерфейсы. Рассмотрим рис. 26.6, на котором изображена типичная сеть, созданная с помощью FreeS/WAN. VPN-маршрутизатор, расположенный сверху, считается "левым". Вам необходимо указать FreeS/WAN IP-адреса, используемые в данной конфигурации. Для этого используются следующие опции.

- leftsubnet. Локальная подсеть, к которой подключен маршрутизатор FreeS/WAN. В примере, показанном на рис. 26.6, это сеть 172.16.0.0/16.
- left. Адрес, связанный с внешним интерфейсом сервера VPN. В большинстве случаев для этой опции задается значение %defaultroute, но вы можете указать конкретный IP-адрес. На рис. 26.6 это адрес 10.0.0.1.
- leftnexthop. IP-адрес обычного маршрутизатора, к которому подключена система VPN. В примере, показанном на рис. 26.6, используется адрес 10.0.0.10. Подобная

информация необходима, так как KLIPS не применяет обычные средства маршрутизации, реализованные в ядре, а передает данные непосредственно следующему маршрутизатору.

- **leftfirewall**. Если подсеть, обслуживаемая VPN-маршрутизатором, содержит IP-адреса, которые не маршрутизируются обычными средствами (например, если она использует средства NAT), либо если VPN-маршрутизатор выполняет также функции брандмауэра, необходимо задать **leftfirewall=yes**.
- rightnexthop. В качестве значения этой опции задается IP-адрес обычного маршрутизатора, который доставляет пакеты удаленной сети.
- right. Данная опция указывает на внешний сетевой интерфейс удаленного VPN-маршрутизатора. Подобно **left**, вы можете принять значение этой опции по умолчанию.
- **rightsubnet**. Блок IP-адресов удаленной подсети. В примере, показанном на рис. 26.6, это сеть 192.168.1.0/24.
- **left**id. Идентификатор "левой" системы. Это может быть IP-адрес, имя домена или имя узла, которому предшествует символ @ (например, **@vpn.threeroomco.**com). Имя узла, перед которым указан символ @, означает, что система не должна пытаться преобразовать имя в IP-адрес.
- **rightid**. Значение данной оппции идентифицирует "правую" часть VPN-соединения.
- **left**rsasigkey. Открытый **RSA-ключ** из файла **/etc/ipsec.** secrets на "левой" стороне VPN-соединения.
- rightrsasigkey. Открытый **RSA-ключ** из файла **/etc/ipsec.secrets** на "правой" стороне VPN-соединения.
- auto. Эта опция совместно с опциями plutoload и **plutostart** определяет, какие соединения должны загружаться и устанавливаться при запуске FreeS/WAN. Если установлены значения **plutoload=%search** и **auto=add**, соединения, соответствующие конфигурации, загружаются, а если заданы значения plutstart=%search и auto-start, соединения устанавливаются.

Не имеет значения, какую из систем вы назовете "левой", а какую "правой". Соединению следует присвоить имя, которое идентифицировало бы обе его стороны. Например, если одна из сетей расположена в Бостоне, а другая в Цинциннати, вы можете использовать имя boscinci, а затем называть систему, находящуюся в Бостоне, "левой", а систему, расположенную в Цинциннати, "правой". Одна и та же конфигурация используется на обеих сторонах соединения; FreeS/WAN выясняет, на какой стороне он находится, анализируя существующие сетевые соединения.

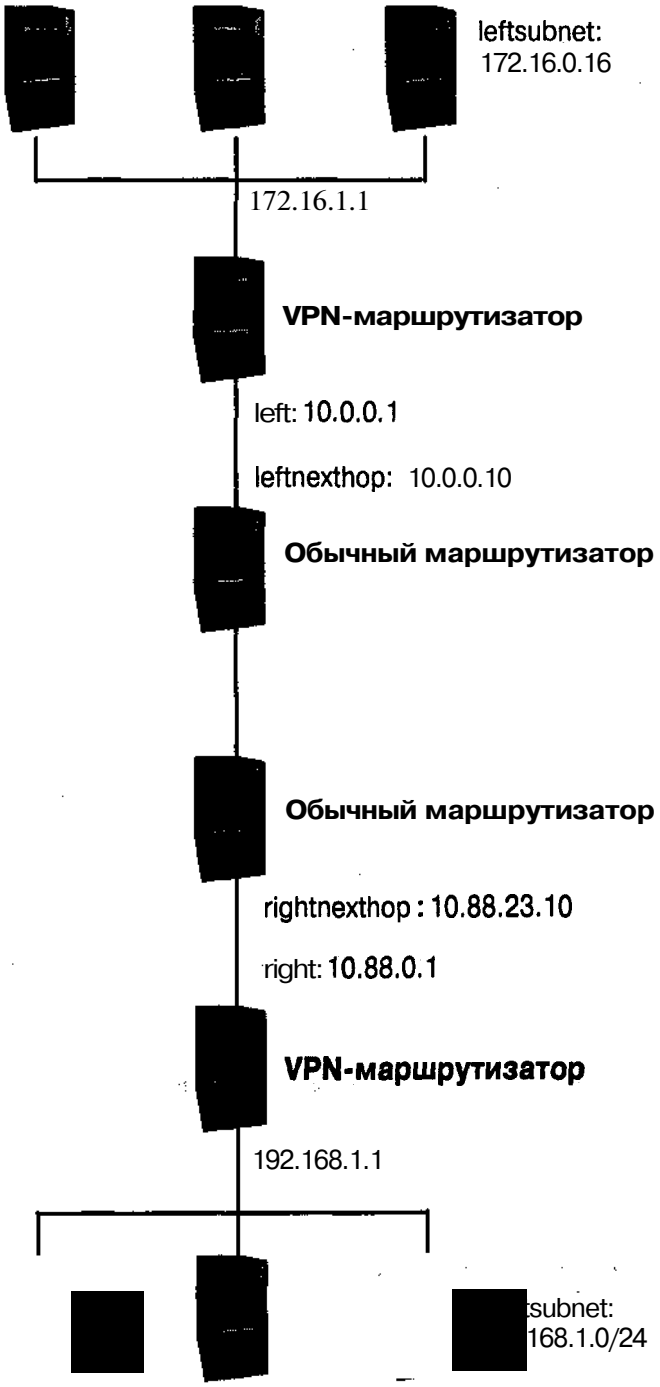


Рис. 26.6. Для определения конфигурации FreeS/WAN надо указать адреса, которые используются при формировании VPN

Установление соединения

Соединение между двумя маршрутизаторами FreeS/WAN устанавливается следующим образом: на одной стороне программа ipsec запускается в режиме демона, а на другой стороне та же программа используется для инициализации соединения. Если настройка выполнена корректно, то соединение должно устанавливаться автоматически, как только программа ipsec начнет выполняться на обеих сторонах. Не имеет значения, какая из систем использует программу в режиме демона; в отличие от PPTP, FreeS/WAN не различает клиентскую и серверную системы.

Для того чтобы запустить программу ipsec в режиме демона, надо выполнить команду

```
# ipsec setup start
```

После выполнения этой команды система, в зависимости от значений опций plutoload, plutostart и auto, загружает соединение, ожидает установления соединения или инициирует соединение. Если при настройке системы вы указали на обеих сторонах соединения опцию auto=add, можете запустить сервер на одной стороне и ожидать запроса на установление соединения, переданного другой системой. Чтобы запрос был передан, надо выполнить команду

```
# ipsec auto --up имя
```

При вызове программы ipsec задается имя соединения, например boscinci. В результате выполнения данной команды система предпримет попытку установить соединение. Для того чтобы проверить, успешной ли была эта попытка, надо использовать команду ipsec look. Если в составе ответа будет содержаться таблица маршрутизации VPN, это означает, что соединение было установлено корректно. Вы также можете использовать для проверки обычные программы сетевого обмена, например ping, traceroute или telnet, однако, если удаленная сеть доступна из Internet, эти инструменты не могут подтвердить тот факт, что связь осуществляется именно через VPN.

После окончания настройки сети можно изменить конфигурацию в файле /etc/ipsec.conf. Конфигурацию можно скорректировать таким образом, что соединение будет автоматически устанавливаться при выполнении команды ipsec setup start. Для запуска сервера FreeS/WAN используется сценарий SysV или локальный сценарий запуска.

Вопросы защиты при использовании VPN

Система VPN призвана повысить безопасность при обмене по сети. Однако она же может открыть злоумышленнику доступ к сетевым ресурсам. Взаимодействие компьютеров и сетей посредством VPN условно показано на рис. 26.1, 26.2 и 26.6. Однако из этих рисунков неочевиден тот факт, что многие из соединений по сути представляют собой два соединения. В качестве примера рассмотрим VPN на базе PPTP, в которой сеть использует VPN-маршрутизатор для взаимодействия с компьютерами под управлением Windows. Реально компьютер Windows имеет два интерфейса: один предназначен для обмена средствами VPN, другой представляет собой обычное Internet-соединение. Логическая структура такой системы представлена на рис. 26.7.

Обычно клиенты VPN считаются клиентами, пользующимися доверием, и при работе с ними не принимают меры предосторожности, как при обмене с обычными Internet-

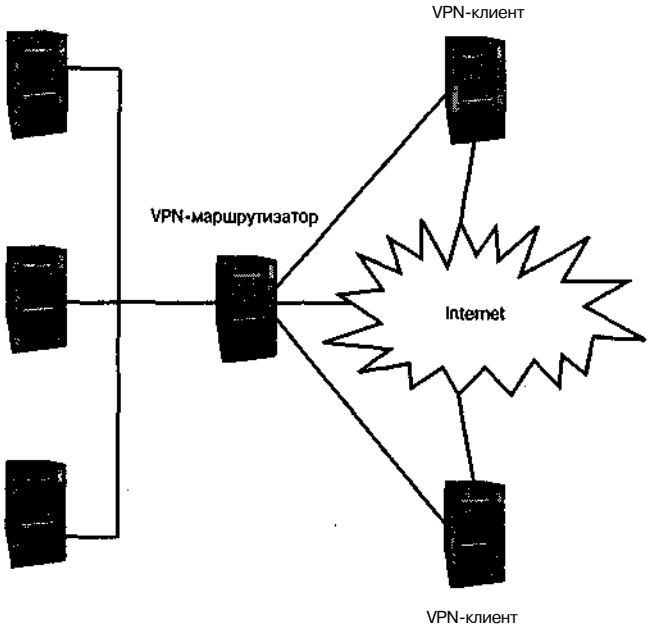


Рис. 26.7. Несмотря на то что VPN-соединение обеспечивает защиту передаваемых данных, взаимодействующие системы поддерживают обычные Internet-соединения, которые могут быть использованы для организации атаки

узлами (по сути, мерами предосторожности являются сами VPN-соединения). В результате, если в защите клиентов Windows имеются недостатки, они могут использоваться для организации атаки. Несмотря на то что сеть защищена брандмауэром, VPN-клиенты рассматриваются как локальные компьютеры, поэтому данные, передаваемые ими, не обрабатываются брандмауэром. Если на компьютере, выполняющем роль VPN-клиента, появился вирус, не исключено, что он сможет найти путь на узлы локальной сети, несмотря на наличие брандмауэра.

Повысить уровень безопасности при использовании VPN можно следующими способами.

- Обеспечить защиту на обеих сторонах VPN. Если обе стороны VPN-соединения одинаково хорошо защищены, то в безопасности будет и вся виртуальная сеть. Такой подход эффективен, когда VPN-соединение устанавливается между локальными сетями: VPN-маршрутизаторы и брандмауэры обеспечивают достаточный уровень защиты. Если же VPN-соединение устанавливается с отдельными компьютерами, реализовать этот способ сложнее, поскольку удаленных VPN-узлов может быть много, кроме того, системный администратор не имеет возможности контролировать их работу. Если пользователь решит установить на своем домашнем компьютере программу, неидеальную с точки зрения безопасности, помешать ему невозможно.

- Контроль обмена с VPN-клиентами. Определив правила брандмауэра, ограничивающие доступ к сетевым ресурсам для удаленных VPN-клиентов, вы тем самым

исключите их из числа узлов, пользующихся доверием. На первый взгляд кажется, что при таком подходе теряются все преимущества VPN, но вы можете задать для VPN-клиентов лишь часть тех ограничений, которые устанавливаются для обычных Internet-узлов. Если пользователь VPN не работает со средствами X Window, вы можете блокировать для его компьютера X-протоколы, в то время как при работе в локальной сети применение этих протоколов разрешено. Этим вы сократите возможности организации атаки с использованием VPN-узла.

Хорошие результаты дает сочетание этих подходов. Вы можете потребовать от сотрудников, использующих PPTP-клиенты, установить на своих компьютерах брандмауэры и одновременно ограничить их доступ, предоставив им возможность обращаться только к отдельным узлам сети и использовать лишь определенные протоколы. Для создания правил, ограничивающих доступ, можно использовать программу iptables, которая рассматривалась в главе 25. Если вы имеете возможность контролировать обе стороны VPN-соединения, вы сможете больше внимания уделить реализации первого подхода и принять одинаковые меры безопасности для обеих систем.

Резюме

Система VPN позволяет расширить локальную сеть за счет взаимодействия с внешними компьютерами. С ее помощью можно предоставить некоторым удаленным пользователям и даже целым сетям возможность обращаться к ресурсам локальной сети, недоступным для обычных Internet-узлов. Этот инструмент может оказаться очень полезным для организации обмена данными с пользователями, работающими с портативными компьютерами, и для объединения различных локальных сетей в одну виртуальную сеть. Функционирование VPN обеспечивается с помощью программных продуктов, часть которых ориентирована на работу в системе Linux. Наиболее часто для организации VPN используется протокол PPTP, реализованный в программах PoPToP и PPTP-Linux, а также протокол FreeS/WAN. PPTP в основном применяется для подключения к сети отдельных VPN-клиентов, а FreeS/WAN — для объединения нескольких сетей. Принимая решение об использовании VPN, необходимо рассмотреть вопросы защиты, в особенности учесть тот факт, что VPN-клиенты могут стать источниками опасности для ресурсов локальной сети.

Предметный указатель

Научно-популярное издание

Родерик В. Смит

Сетевые средства Linux

Литературный редактор *И. А. Попова*
Верстка *А. И. Полинчик*
Художественный редактор *С. А. Чернокозинский*
Корректоры *Л. А. Гордиенко,*
О. В. Мишутина,
Л. В. Чернокозинская

Издательский дом "Вильяме".
101509, Москва, ул. Лесная, д. 43, стр. 1.
Изд. лиц. ЛР № 090230 от 23.06.99
Госкомитета РФ по печати.

Подписано в печать 27.02.2003. Формат 70x100/16.
Гарнитура Times. Печать офсетная.
Усл. печ. л. 47,7. Уч.-изд. л. 44,6.
Тираж 3000 экз. Заказ № 2537.

Отпечатано с диапозитивов в ФГУП "Печатный двор"
Министерства РФ по делам печати,
телерадиовещания и средств массовых коммуникаций.
197110, Санкт-Петербург, Чкаловский пр., 15.

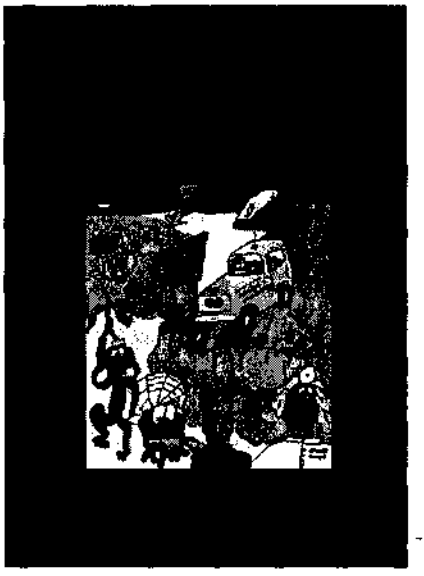
Эта книга задумывалась как учебник для вузов и как справочное руководство для специалистов, поэтому она написана на высоком профессиональном уровне. Специалисты могут почерпнуть в ней подробное описание технологии сетей TCP/IP и структуры Internet. Автор книги не ставил перед собой цель заменить описание существующих стандартов протоколов. Тем не менее книгу можно рассматривать как великолепную отправную точку в изучении технологии глобальных сетей, поскольку в ней изложены основы и сделан акцент на принципах их работы. Кроме того, книга дает читателю ориентиры для поиска дополнительной информации, которые было бы трудно получить на основе изучения отдельных стандартов протоколов.

Эта книга предназначена для программистов, стремящихся изучить тонкости создания сетевых приложений для Linux. В ней рассматриваются принципы взаимодействия типа клиент/сервер и приведены алгоритмы работы клиентских и серверных компонентов распределенных программ. Каждый проект проиллюстрирован практическим примером, и, наряду с этим, описаны необходимые методы организации сетевого взаимодействия, включая шлюзы уровня приложений и тунелирование. Кроме того, в книге рассматривается несколько стандартных прикладных протоколов, на примере которых описаны алгоритмы и методы реализации. В последних главах рассматриваются некоторые тонкости управления параллельной работой и дан обзор методов, позволяющих программисту оптимизировать производительность приложений. Поскольку книга в основном посвящена описанию способов использования, а не принципов работы объединенной сети, для ее изучения не требуется предварительная подготовка по сетям.

www.williamspublishing.com

РУКОВОДСТВО АДМИНИСТРАТОРА LINUX

Эви Немет,
Гарт Снайдер,
Трент Хейн



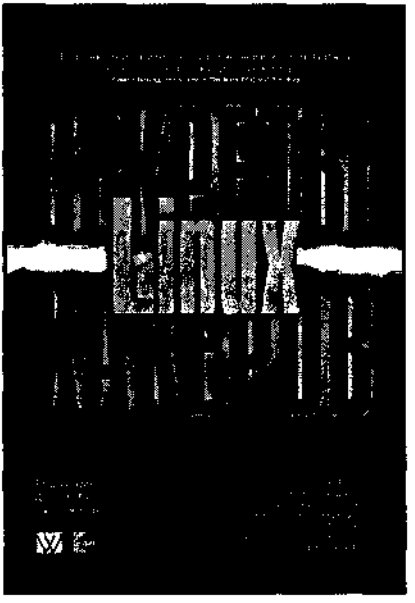
www.williamspublishing.com

в продаже

Эта книга является надежным помощником системного администратора Linux и служит источником практических советов и полезных сведений по теории системного администрирования. Книга в первую очередь является практическим руководством, цель которого — не пересказывать содержание документации, а поделиться с читателями коллективным опытом авторов в области системного администрирования. Примеры в большинстве случаев взяты из практики эксплуатации реальных систем со всеми их подводными камнями и нюансами. В книге рассмотрены три основных дистрибутива Linux: Red Hat 7.2, SuSE 7.3 и Debian 3.0. Эти дистрибутивы выбраны потому, что они наиболее популярны и позволяют продемонстрировать весь спектр подходов к вопросу администрирования Linux-систем. В то же время большая часть материала книги применима и к другим дистрибутивам общего назначения. Это одна из немногих книг, предназначенных не для широкого круга пользователей, а для системных администраторов, работающих в среде Linux. Изложенный материал будет полезен как профессионалам, так и новичкам, еще только постигающим тонкости этой увлекательной и трудной работы.

СЕКРЕТЫ ХАКЕРОВ.
БЕЗОПАСНОСТЬ LINUX - ГОТОВЫЕ РЕШЕНИЯ

Брайан Хетч,
Джеймс Ли,
Джордж Курц



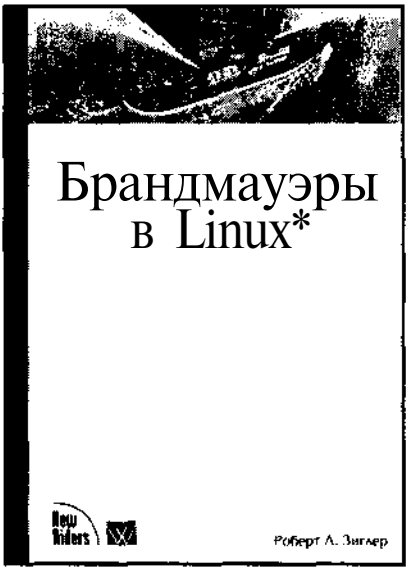
www.williamspublishing.com

в продаже

Данную книгу можно назвать продолжением всемирно известного бестселлера *Секреты хакеров: безопасность сетей — готовые решения, 2-е изд.,* в которой все внимание сосредоточено на безопасности при работе в ОС Linux. Ее авторы уже много лет являются ведущими и признанными специалистами в области защиты компьютерных систем. Это позволило им рассмотреть проблемы хакинга в Linux на новом, не имеющем аналогов уровне. Книга относится к тому редкому типу книг, которые наглядно объясняют, что именно происходит, когда злоумышленники атакуют системы Linux. Читателям продемонстрировано, чем Linux отличается от других Unix-подобных систем, раскрыты хакерские методы осуществления всех типов атак, которые используются для получения несанкционированного доступа к системам Linux, нарушения работы их служб и взлома компьютерных сетей. Детально изучены средства противодействия атакам хакеров и методы оперативного выявления вторжения. В этой книге нет пустых мест — после описания реальных листингов выполняемых атак предоставляются такие же реальные рецепты отражения каждой конкретной атаки. Так как материал книги изложен простым и доступным языком, с использованием наглядных примеров, то книга будет полезна самому широкому кругу читателей — на-чиная от обычных пользователей домашних компьютеров и заканчивая высококвалифицированными системными администраторами крупных компаний.

БРАНДМАУЭРЫ В LINUX

Робер т Л. Зиглер



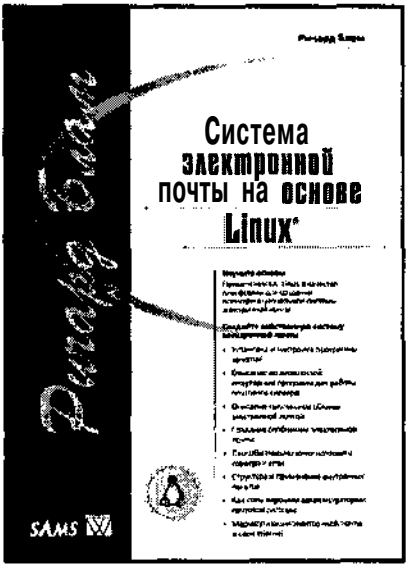
www.williamspublishing.com

в продаже

По мере развития Internet все более актуальным становится вопрос защиты от несанкционированного доступа. Одним из средств такой защиты является брандмауэр, отделяющий локальную сеть организации от Internet. Автор данной книги рассматривает вопросы установки и конфигурации брандмауэра в системе Linux, начиная с простого брандмауэра, предназначенного для защиты одной системы и заканчивая сложной архитектурой с двумя брандмауэрами и DMZ. Автор описывает принципы фильтрации пакетов и приводит примеры правил брандмауэра, выполняющих подобную фильтрацию. Рассмотрение правил брандмауэра производится на примере программы ipchains, поддерживаемой в системе Red Hat .6.0. В приложении приводится пример сценария брандмауэра. Кроме того, администраторам, которые по каким-либо причинам не могут использовать ipchains, предлагается пример брандмауэра на основе программы ipfwadm. Даже защитив локальную сеть брандмауэром, администратор не имеет права терять бдительности и считать, что его компьютерам ничего не грозит. Средства защиты реализованы практически в каждом сервере и эту возможность необходимо использовать. Автор подробно рассматривает особенности конфигурации различных серверов, позволяющих ограничить число узлов, с которых разрешен доступ, и исключить возможность утечки данных из локальной сети.

СИСТЕМА ЭЛЕКТРОННОЙ ПОЧТЫ НА ОСНОВЕ LINUX. РУКОВОДСТВО АДМИНИСТРАТОРА

Ричард Блам



www.williamspublishing.com

в продаже

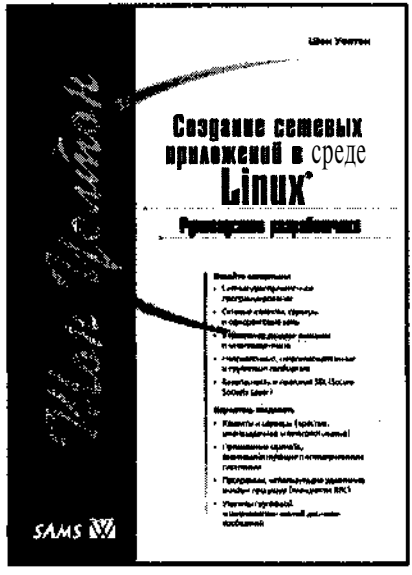
Основной целью создания этой книги является оказание помощи сетевым администраторам небольших учреждений в организации высокопроизводительной и качественной системы электронной почты, которая бы не уступала коммерческим аналогам. Кроме того, если предполагается подключение вашей системы электронной почты к сети Internet, то здесь вы найдете подробные материалы по этой теме. Вся информация, представленная в книге, может использоваться либо в учебных целях, либо В качестве справочника (или и в том, и другом качестве).

В книге приводятся примеры установки и настройки сервера электронной почты (почтового сервера) на базе ОС Linux для фиктивной организации. Автор старался сконцентрировать внимание на офисных компьютерных сетях.

И хотя многие концепции программы sendmail предполагают ее применение в крупных корпоративных сетях и даже провайдерами Internet, целью автора не было создание справочника для них (хотя эта книга имеет сходство со справочником). Все главы писались, исходя из проблем, с которыми сталкиваются администраторы систем электронной почты небольшого офиса.

СОЗДАНИЕ СЕТЕВЫХ ПРИЛОЖЕНИЙ В СРЕДЕ LINUX. РУКОВОДСТВО РАЗРАБОТЧИКА

Шон Уолтон



www.williamspublishing.com

в продаже

Данная книга в основном посвящена программированию сокетов на языке C в среде Linux. В ней шаг за шагом рассказывается о том, как писать профессиональные сетевые клиентские, серверные и одноранговые приложения. Рассматриваются вопросы теоретического и практического построения эффективных и гибких приложений, использующих давно существующие и новые сетевые технологии.

Книга разбита на пять частей.

В части *Создание сетевых клиентских приложений* представлены вводные сведения о сокетах, определены основные термины, описаны различные типы сокетов и принципы адресации, изложена базовая теория сетей.

В части *Создание серверных приложений* рассматриваются серверные технологии, алгоритмы многозадачности, механизмы ввода-вывода и параметры сокетов.

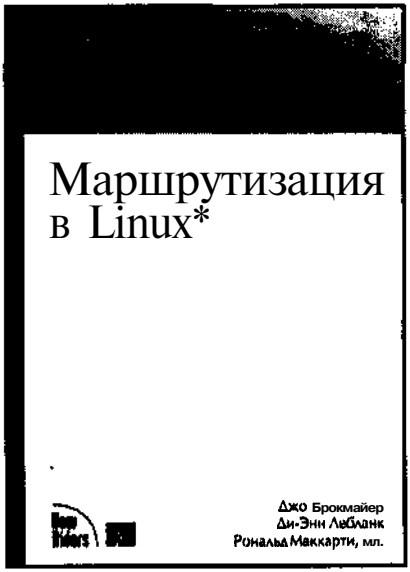
В части *Объектно-ориентированные сокеты* изложены объектно-ориентированные подходы к сетевому программированию на языках Java и C++, а также описаны достоинства и недостатки объектной технологии в целом.

В часть *Сложные сетевые методики* включены главы по технологии RPC, протоколу SSL, работе в групповом и широковещательном режимах, программированию неструктурированных сокетов, стандарту IPv6. В отдельную часть вынесены приложения, в которых содержится справочный материал, касающийся сокетов.

В приложение А включены таблицы, которые слишком велики для основных глав. В приложениях Б и В описаны системные функции работы с сокетами и функции ядра.

МАРШРУТИЗАЦИЯ В LINUX

Джо Брокмайер, Ди-Энн Лебланк, Рональд Маккарти-младший



www.williamspublishing.com

в продаже

Данная книга является учебным пособием, написанным профессионалами Linux с целью научить читателей настраивать подсистему маршрутизации в Linux. В книге излагается теория маршрутизации, рассказывается об основных протоколах и утилитах, имеющихся в распоряжении пользователей Linux, описывается, в каких ситуациях лучше всего применять те или иные средства. Книга разбита на три части. В части *Основы маршрутизации* описываются поддерживаемые в Linux протоколы одноадресной (RIP-1, RIP-2 и OSPF), многоадресной (DVMRP, MOSPF, PIM-SM и PIM-DM) и пограничной маршрутизации (EGP, BGP, BGMP и MSDP). В отдельной главе приводится сравнение стандартов IPv4 и IPv6 и используемых в них принципов классовой и бесклассовой адресации. В части *Средства и технологии маршрутизации в Linux* рассказывается о существующих в Linux демонах одноадресной и многоадресной маршрутизации, в том числе routed, gated, mrouted, pimd, pppd, rip2ad и pptpd; о различных сетевых командах, таких как ifconfig, netstat, route, arp, ping, traceroute и tcpdump; о конфигурировании ядра Linux и систем фильтрации пакетов, а также многое другое. В отдельную часть вынесены приложения, в которых содержится сводка по информационным ресурсам, касающимся маршрутизации.

Книга предназначена опытным пользователям и администраторам Linux, которых интересует не только создание простейших сетей и подсетей, но и реализация более сложных решений, связанных с различными протоколами маршрутизации.

ИСПОЛЬЗОВАНИЕ LINUX, 6-Е ИЗДАНИЕ. СПЕЦИАЛЬНОЕ ИЗДАНИЕ

Дэвид Бендел, Роберт Нейпир

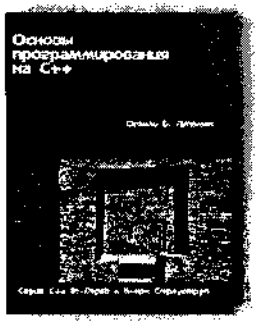
www.williamspublishing.com

в продаже

Книга содержит практические аспекты использования Linux как пользователями, имеющими только начальные навыки работы с Linux, так и опытными пользователями, желающими расширить свои познания в этой области. Кроме описания приемов работы и конфигурирования популярных графических сред KDE и GNOME, приведены сведения об общей настройке системы и ее архитектуре, различных аспектах системного администрирования Linux и работе в сетевой среде. Описание принципов работы той или иной подсистемы сопровождается рекомендациями по ее настройке и практическими примерами ее эффективного использования. Хотя изложение ориентировано на использование одного из трех наиболее популярных дистрибутивов — Caldera OpenLinux, Red Hat Linux и Debian GNU/Linux, большая часть изложенного материала применима к любому из существующих дистрибутивов Linux. Книга окажется безусловно полезной в качестве руководства для пользователей и системных администраторов Linux начального и среднего уровня, а также в качестве справочного руководства для опытных пользователей.

НОВИНКИ ИЗДАТЕЛЬСКОЙ ГРУППЫ ДИАЛЕКТИКА/ВИЛЬЯМС

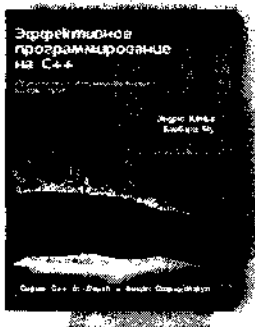
серия книг C++ In-Depth



Основы программирования на C++. Серия C++ In-Depth, т. 1

Стэнди Б. Липпман

Эта книга поможет вам быстро освоить язык C++. Обширные и сложные темы исчерпывающе представлены в ней на уровне основных концепций, которые необходимо знать каждому программисту для написания реальных программ на языке C++. Приведенные примеры и предлагаемые упражнения весьма эффективны, что поможет быстро освоить излагаемый материал. Основное внимание уделяется тем аспектам программирования на языке C++, которые будут представлять интерес для каждого программиста-практика, а обсуждаемые технологии и методы позволят найти решение для практической любой задачи, взятой из реального мира. Книга будет интересна всем, кто только планирует освоить или уже практически использует язык C++.*



Эффективное программирование на C++. Серия C++ In-Depth, т. 2

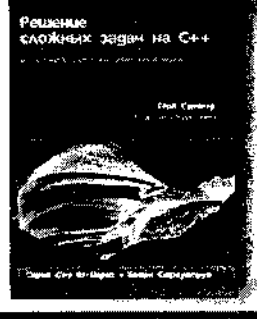
Эндрю Кёниг, Барбара Э. Му

Эта книга, в первую очередь, предназначена для тех, кому хотелось бы быстро научиться писать настоящие программы на языке C++. Зачастую новички в C++ пытаются освоить язык чисто механически, даже не попытавшись узнать, как можно эффективно применить его к решению каждодневных проблем. Цель данной книги - научить программированию на C++, а не просто изложить средства языка, поэтому она полезна не только для новичков, но идля тех, кто уже знаком с C++ и хочет использовать этот язык в более натуральном, естественном стиле.

Современное проектирование на C++. Серия C++ In-Depth» т* 3

Андрей Александреску

В книге изложена новая технология программирования, представляющая собой сплав обобщенного программирования, метапрограммирования шаблонов и объектно-ориентированного программирования на C++. Настраиваемые компоненты, созданные автором, высоко подняли уровень абстракции, наделив язык C++ чертами языка спецификации проектирования, сохранив всю его мощь. В книге изложены способы реализации основных шаблонов проектирования. Разработанные компоненты воплощены В библиотеке Loki, которую можно загрузить с Web-страницы автора. Книга предназначена для опытных программистов на C++.



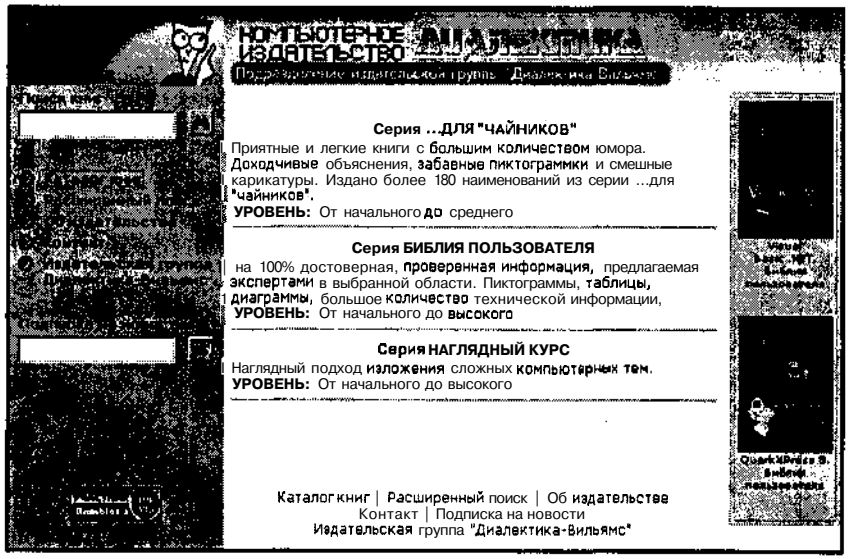
Решение сложных задач на C++. Серия C++ In-Depth, т. 4

Герб Саттер

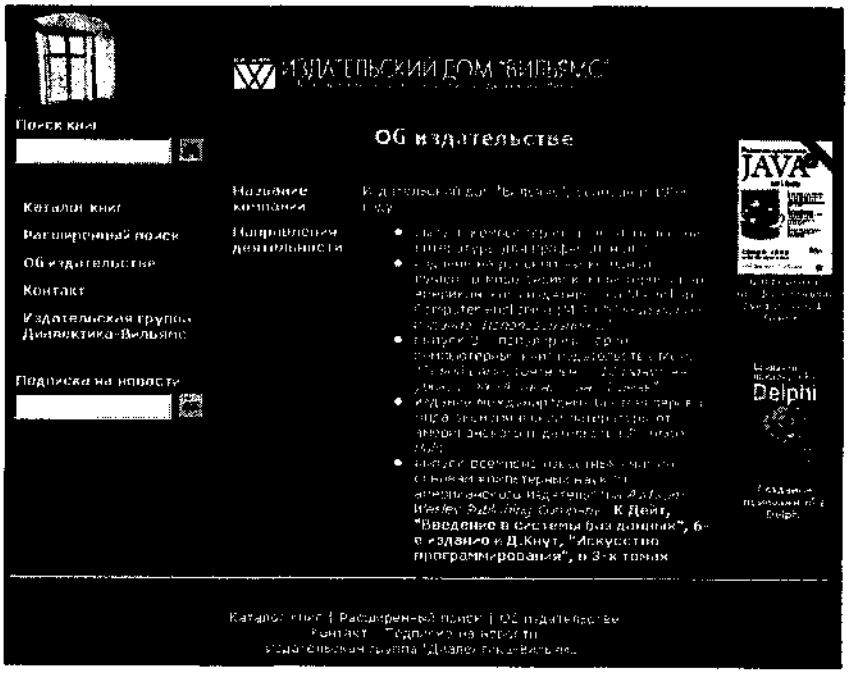
В данном издании объединены две широко известные профессионалам в области программирования на C++ книги Герба Саттера *Exceptional C++* и *More Exceptional C++* , входящие в серию книг C++ *In-Depth,* редактором которой является Бьерн Страуструп, создатель языка C++. Материал этой книги составляют переработанные задачи серии *Guru of the Week,* рассчитанные на читателя с достаточно глубоким знанием C++, однако книга будет полезна каждому, кто хочет углубить свои знания в этой области.

www.williamspublishing.com

www.dialektika.com



www.williamspublishing.com



Cisco SYSTEMS

СЕТЕВЫЕ СРЕДСТВА

РОДЕРИК В. Сми

В настоящее время число сетевых приложений постоянно увеличивается, и они используются для решения все более важных задач. Конфигурация серверов, устанавливаемая по умолчанию, далеко не всегда обеспечивает выполнение всех программ, поэтому системный администратор должен уметь конфигурировать сетевые средства с учетом специфики работы приложений. Книга *Сетевые средства Linux* поможет вам овладеть знаниями, необходимыми для того, чтобы настроить серверы для решения текущих задач, повысить эффективность и безопасность их работы.

Книга начинается с вводной части, в которой описываются низкоуровневая конфигурация системы и настройка базовых сетевых средств. Часть II посвящена серверам, используемым для обеспечения работы локальной сети. В этой части рассматриваются серверы DHCP, Kerberos, Samba, истинного времени, инструменты создания резервных копий и другие подобные средства. В части III описаны Internet-серверы DNS, SMTP (sendmail, Postfix и Exim), HTTP (Apache) и FTP. В части IV обсуждаются вопросы защиты сетей, в частности формирование поддерева enroot, создание брандмауэров с помощью утилиты iptables и организация VPN.

В процессе изложения материала а тор обращает внимание читателей особенности инсталляции и настро ки сетевых средств в системах Calde OpenLinux, Debian GNU/Linu Mandrake, Red Hat, Slackware, SuS и TurboLinux.

Особое внимание в книге уделяет< следующим вопросам.

- Установка базовой конфигураци основных серверов
- Оптимизация сетевых функци Linux
- Применение расширенных сетевь средств
- Решение поставленных задач с ис пользованием различных режиме работы программ
- Устранение всех факторов, которы могут представлять угрозу для систем и противоречат политике провайдера
- Обеспечение защиты данных

Книгу *Сетевые средства Linu* можно рассматривать как учебно и справочное пособие. Она поможе вам изучить вопросы, связанны с организацией сетей на базе Linux и повысить свою квалификацию ка системного администратора

Родерик В. Смит — квалифицированный системный администратор работающий в основном в системе Linux. Он является автором нескольки книг, в том числе *Broadband Internet Connections* (Addison-Wesley, 2002), *Linu Samba Server Administration* (Sybex, 2001), *The Multi-Boot Configuration Handboo* (Que, 2000), *Linux: Networking for your Office* (SAMS, 2000). Родерик получи степень доктора в области когнитивной психологии в Tufts University.

ISBN 5-8459-0426-9

ВИЛЬЯМС
www.williamspublishing.com
ADDISON WESLEY http://ww w.awl.com/cseng/