DNS and BIND

Fourth Edition

Paul Albitz and Cricket Liu

O'REILLY®

DNS и BIND

Четвертое издание

Пол Альбитц и Крикет Ли

Санкт-Петербург
2002

Пол Альбитц, Крикет Ли

DNS и BIND, 4-е издание

Перевод М. Зислиса

Главный редактор	*А. Галунов*
Зав. редакцией	*Н. Макарова*
Научный редактор	*А. Маврин*
Редактор	*А. Лосев*
Корректура	*О. Маршкова*
Верстка	*Н. Гриценко*

Альбитц П., Ли К.
DNS и BIND. – Пер. с англ. – СПб: Символ-Плюс, 2002. – 696 с., ил.
ISBN 5-93286-035-9

Книга «DNS и BIND» стала библией для системных администраторов. Она уникальна по полноте изложения материала, что в сочетании с прекрасным авторским стилем делает ее незаменимой и актуальной для каждого, кто хочет наладить эффективную работу DNS. Четвертое издание включает информацию о версии 9 пакета BIND, в которой реализованы новые и очень важные механизмы, а также о версии 8, входящей в состав большинства действующих коммерческих разработок. Пакеты BIND 8 и 9 позволяют значительно повысить безопасность служб DNS.

Рассмотрены следующие темы: функциональность и принципы работы DNS; структура пространства доменных имен; установка и настройка серверов имен; применение MX-записей для маршрутизации почты; настройка узлов на работу с DNS; разделение доменов на поддомены; обеспечение безопасности DNS-сервера; новые возможности BIND 9; расширения системы безопасности DNS (DNSSEC) и подписи транзакций (TSIG); распределение нагрузки между DNS-серверами; динамические обновления, асинхронные уведомления об изменениях зон, пошаговая передача зон; устранение неполадок (применение nslookup и dig, чтение отладочного вывода); DNS-программирование с применением библиотеки DNS-клиента и модуля Perl Net::DNS.

ISBN 5-93286-035-9
ISBN 0-596-00158-4 (англ)

Издательство «Символ-Плюс». 193148, Санкт-Петербург, ул. Пинегина, 4,
тел. (812) 324-5353, edit@symbol.ru. Лицензия ЛП N 000054 от 25.12.98.
Налоговая льгота – общероссийский классификатор продукции
ОК 005-93, том 2; 953000 – книги и брошюры.
Подписано в печать 26.02.2002. Формат 70x100 1/16 . Печать офсетная.
Объем 43,5 печ. л. Тираж 3000 экз. Заказ N
Отпечатано с диапозитивов в Академической типографии «Наука» РАН
199034, Санкт-Петербург, 9 линия, 12.

Оглавление

Предисловие

Возможно, вам не так уж много известно о системе доменных имен (Domain Name System), но работая в Интернете, вы неизбежно ее используете. Всякий раз, отправляя сообщения электронной почты или исследуя просторы World Wide Web, вы полагаетесь на DNS – систему доменных имен.

Дело в том, что люди предпочитают запоминать *имена* компьютеров, а компьютерам больше нравится обращаться друг к другу по числовым адресам. В Интернете этот адрес имеет разрядность 32, то есть может быть числом от нуля до четырех с хвостиком миллиардов.[1] Компьютеры с легкостью запоминают такие вещи, потому что обладают большими объемами памяти, идеально подходящей для хранения чисел, но для людей эта задача не в пример сложнее. Попробуйте случайным образом выбрать из телефонной книги десять номеров и запомнить их. Непросто? Теперь вернитесь к началу телефонной книги и сопоставьте каждому номеру случайный код района. Примерно настолько же сложно будет запомнить 10 произвольных интернет-адресов.

Отчасти именно по этой причине необходима система доменных имен. DNS занимается двунаправленным отображением имен хостов, подходящих для запоминания людьми, и интернет-адресов, с которыми работают компьютеры. По сути дела, DNS в сети Интернет является не только средством работы с адресами, но и стандартным механизмом для предоставления и получения разнообразной информации об узлах сети. DNS нужен практически для каждой программы, обеспечивающей сетевое взаимодействие, в том числе программам для работы с электронной почтой, терминальным клиентам (например, telnet), средствам передачи файлов, таким как FTP, и, разумеется, веб-броузерам, таким как Netscape Navigator и Microsoft Internet Explorer.

Другой важной особенностью DNS является способность системы распространять информацию о хосте по всей сети Интернет. Хранение доступной информации о хосте на единственном компьютере полезно лишь для тех, кто пользуется этим компьютером. Система доменных имен обеспечивает получение информации из любой точки сети.

Более того, DNS позволяет распределять управление информацией о хостах между многочисленными серверами и организациями. Нет необ-

---

[1] А в системе IP-адресации версии 6 адреса имеют колоссальную длину – 128 бит, что позволяет охватить десятичные числа от нуля до 39-значных.

ходимости передавать данные на какой-то центральный сервер или регулярно синхронизировать свою базу данных с «основной». Достаточно убедиться, что ваш раздел, называемый *зоной*, соответствует действительности на ваших *DNS-серверах*. А они, в свою очередь, сделают информацию о зоне доступной всем остальным DNS-серверам сети.

Поскольку база данных DNS является распределенной, в системе должна быть предусмотрена возможность поиска нужной информации путем опроса множества возможных источников ее получения. Система доменных имен наделяет DNS-серверы способностью находить нужные источники информации и получать сведения по любой зоне.

Разумеется, система DNS не лишена недостатков. К примеру, в целях избыточности базы данных, система позволяет хранение зональной информации на более чем одном сервере. При этом возникает опасность десинхронизации копий зональной информации.

Но *самая большая* проблема, связанная с DNS, несмотря на широкое распространение в сети Интернет, – это реальное отсутствие хорошей документации по работе с системой. Большинство администраторов сети Интернет вынуждены обходиться лишь той документацией, которую считают достаточной поставщики используемых программ, а также тем, что им удается выудить из соответствующих списков интернет-рассылок и конференций Usenet.

Такой дефицит документации означает, что понимание предельно важной интернет-службы, одной из монументальных основ сегодняшней сети Интернет, либо передается от администратора к администратору как ревностно хранимая семейная тайна, либо постоянно изучается повторно отдельными программистами и разработчиками. Новые администраторы зон повторяют ошибки, уже бесчисленное число раз сделанные другими.

Цель этой книги – изменить сложившуюся ситуацию. Мы осознаем, что не у каждого читателя есть время и желание становиться специалистом по DNS. У большинства из вас есть достаточно других занятий, помимо управления зонами и DNS-серверами: системное администрирование, разработка сетевых инфраструктур, или разработка программного обеспечения. Заниматься исключительно DNS может только сотрудник безумно большой организации. Мы постарались представить информацию, достаточную для решения основных рабочих задач, будь то управление небольшой зоной или целой международной системой, работа с единственным сервером имен или наблюдение за сотней серверов. Извлеките из книги нужный вам минимум, и возвращайтесь к ней по мере необходимости.

DNS – это сложная тема, настолько сложная, что взяться за нее пришлось не одному, а двум авторам; и мы постарались представить систему настолько прозрачно и доступно, насколько это возможно. В первых двух главах содержится теоретический обзор и достаточный для применения объем практической информации, а в последующих гла-

вах использование системы доменных имен рассмотрено более подробно. С самого начала мы предлагаем читателям нечто вроде дорожной карты, чтобы каждый мог выбрать свой собственный путь изучения книги, соответствующий рабочим задачам или интересам.

Когда речь пойдет о программах, обеспечивающих работу DNS, мы практически целиком сконцентрируемся на инструменте под названием BIND, Berkeley Internet Name Domain, который является наиболее популярной (и наиболее нами изученной) реализацией спецификаций DNS. Мы старались представить в этой книге выжимку из нашего опыта управления и поддержки зон с помощью BIND. (Так получилось, что некоторое время одна из наших зон являлась самой большой зоной сети Интернет; правда, это было очень давно). Где это было возможно, мы включали реальные программы, используемые нами в администрировании; многие из них переписаны на языке Perl с целью достижения большей скорости работы и повышения эффективности.

Надеемся, эта книга поможет вам познакомиться с системой DNS и инструментом BIND, если вы еще новичок, лучше понять их работу, если вы уже знакомы, и приобрести ценное понимание и опыт, даже если вы уже знаете DNS и BIND, как свои пять пальцев.

Версии

Четвертое издание этой книги затрагивает новые версии BIND – 9.1.0 и 8.2.3, а также более старую версию 4.9. Несмотря на то, что на момент написания этой книги версии 9.1.0 и 8.2.3 являются наиболее свежими, они пока не получили широкого распространения в составе Unix-систем, отчасти потому, что обе версии были выпущены недавно, а многие поставщики настороженно относятся к использованию новых программ. Мы время от времени упоминаем и другие версии BIND, в частности 4.8.3, поскольку многие поставщики продолжают распространять программы, содержащие код, основанный на более старых версиях, в составе своих Unix-разработок. Если определенная возможность доступна только в версии 4.9, 8.2.3 или 9.1.0, либо если существуют различия в поведении версий, мы постараемся четко определить, что именно работает и для какой версии BIND.

В наших примерах мы очень часто прибегаем к служебной программе DNS – *nslookup*. Мы пользуемся *nslookup* из комплекта поставки BIND версии 8.2.3. Более старые версии *nslookup* обеспечивают большую часть функциональности (но не всю) *nslookup* версии 8.2.3.[1] В большинстве примеров мы использовали команды, доступные почти во всех версиях *nslookup*; случаи, когда это было невозможно, отмечены отдельно.

[1] Это верно также и для nslookup из комплекта поставки BIND версии 9.1.0. См. более подробно в главе 12 «nslookup и dig».

Что нового в четвертом издании?

Текст книги был обновлен, чтобы соответствовать наиболее поздним версиям BIND; мы также добавили в большом объеме новый материал:

- Более подробное рассмотрение динамических обновлений и механизма NOTIFY, включая и подписываемые динамические обновления (signed dynamic updates), а также описание нового для BIND 9 механизма *update-policy* – в главе 10.
- Поэтапная передача зоны – также в главе 10.
- Зоны ретрансляции, поддерживающие передачу по условию (conditional forwarding), – в главе 10.
- Прямое и обратное отображение адресов в контексте технологии IPv6 с использованием записей новых типов A6 и DNAME, а также бит-строковых меток – в конце главы 10.
- Новый механизм подтверждения подлинности транзакций – транзакционные подписи (transaction signatures, известные также как TSIG) – описан в главе 11.
- Более подробное рассмотрение вопросов обеспечения безопасности DNS-серверов – в главе 11.
- Более подробное рассмотрение работы с брандмауэрами в сети Интернет – в главе 11.
- Описаны расширения DNS, связанные с безопасностью (DNS Security Extensions или DNSSEC), представляющие собой новый механизм цифровой подписи зональных данных, – все в той же 11-ой главе.
- Раздел, посвященный совместной работе клиентов, серверов и контроллеров доменов Windows 2000 и BIND, – в главе 16.

Структура

Порядок следования глав настоящей книги приблизительно соответствует возможному развитию зоны и росту знаний ее администратора. В главах 1 и 2 обсуждается теория системы доменных имен. В главах с 3 по 6 рассматриваются вопросы, связанные с принятием решений по созданию собственных зон, а также действия администратора в случае необходимости создать зону. Следующая часть книги, главы с 7 по 11, посвящена сопровождению зон, настройке хостов для использования DNS-серверов, планированию развития зон, созданию доменов различных уровней и безопасности серверов. Наконец, главы с 12 по 16 посвящены разрешению сложностей, возникающих при работе с различными инструментами, общим проблемам и забытому искусству программирования с применением библиотек DNS-клиента.

Перечислим темы по главам:

Глава 1 «Основы» описывает исторический фон создания системы, и посвящена проблемам, приведшим к созданию DNS, а также собственно обзору теории системы доменных имен.

Глава 2 «Как работает DNS» посвящена более подробному рассмотрению теоретических основ DNS, в частности – организации пространства имен в системе DNS, доменов, зон и DNS-серверов. Там же рассматриваются такие важные понятия, как разрешение адресов и кэширование.

Глава 3 «С чего начать?» посвящена вопросам получения пакета BIND в случае его отсутствия, применениям пакета, когда он уже у вас в руках, определению и выбору доменного имени, а также установления связи с организацией, которая обладает полномочиями делегировать выбранную зону.

Глава 4 «Установка BIND» – это подробное рассмотрение того, как установить два первых DNS-сервера на основе BIND, включая создание базы данных серверов, запуск и диагностику их работы.

Глава 5 «DNS и электронная почта» рассказывает о записи DNS типа MX, которая позволяет администраторам задавать альтернативные узлы, которым передается на обработку почта для определенных адресов. В этой главе описаны стратегии маршрутизации почты для различных типов сетей и узлов, включая сети с интернет-брандмауэрами и узлы, не имеющие прямого подключения к сети Интернет.

Глава 6 «Конфигурирование узлов» рассказывает о том, как настраивать клиентскую часть (*resolver*) BIND, а также об особенностях реализаций клиента – как в составе распространенных Unix-систем, так и применяемых на платформах Windows 95/NT/2000.

Глава 7 «Работа с BIND» посвящена регулярным действиям администратора, выполнение которых необходимо для поддержания устойчивой работы зон, находящихся под его началом, в частности – проверке состояния DNS-сервера и вопросов, касающихся авторитативных серверов зоны.

Глава 8 «Развитие домена» рассказывает о планировании роста и эволюции зон, включая вопросы о том, как вырасти большим, а также о планировании переездов и перебоев в работе.

Глава 9 «Материнство» – о радостях, связанных с обретением потомства. Мы расскажем, когда имеет смысл заводить детей (создавать поддомены), как их называть, *как* их заводить (!) и как присматривать за ними.

Глава 10 «Дополнительные возможности» рассказывает о параметрах настройки сервера имен, которые используются не очень часто, но могут помочь в тонкой настройке DNS-сервера и в упрощении процесса администрирования.

Глава 11 «Безопасность» посвящена обеспечению безопасности и тем настройкам DNS-сервера, которые относятся к работе с интернет-

брандмауэрами, а также двум новым технологиям DNS, связанным с безопасностью: DNS Security Extensions и подписям транзакций (Transaction Signatures).

Глава 12 «nslookup и dig» подробно рассказывает о самых популярных инструментах DNS-отладки, и содержит описания способов извлечения неявной информации из удаленных DNS-серверов.

Глава 13 «Чтение отладочного вывода BIND» – это Розеттский камень[1] отладочной информации BIND. Глава поможет разобраться в таинственной отладочной информации, создаваемой пакетом BIND, а это, в свою очередь, поможет лучше понять, как работает DNS-сервер

Глава 14 «Разрешение проблем DNS и BIND» содержит описания и способы разрешения многих распространенных проблем, связанных с использованием DNS и BIND, а также рассказывает о более редких случаях, связанных с ошибками, диагностика которых может вызывать затруднения.

Глава 15 «Программирование с использованием библиотечных функций» рассказывает о том, как использовать функции библиотеки клиента BIND для опроса DNS-серверов и получения информации в программе на языке C или Perl. Приводится исходный текст полезной (как мы надеемся) программы, которая проверяет работоспособность DNS-серверов и их авторитативность.

Глава 16 «Обо всем понемногу» посвящена незатронутым темам. Она содержит описание использования масок (wildcards) в DNS, принципов работы с хостами и сетями, не имеющими постоянного подключения к сети Интернет, кодировки сетевых имен, экспериментальных типов записей и работы с DNS в Windows 2000.

Приложение A «Формат сообщений DNS и RR-записи (resource records)» содержит предельно подробный справочник по форматам, используемым в запросах и ответах DNS, а также полный перечень определенных в настоящее время типов RR-записей.

Приложение B «Таблица совместимости BIND» – это перечисление наиболее важных особенностей самых распространенных версий BIND.

Приложение C «Сборка и установка BIND на Linux-системах» содержит пошаговые инструкции по сборке BIND версии 8.2.3 в Linux.

---

[1] Розеттский камень – черная базальтовая плита с трехъязычной надписью на египетском иероглифическом, египетском демотическом (разговорном) и древнегреческом языках, обнаруженная в 1799 г. офицером наполеоновских войск Бушаром при сооружении форта Сен-Жюльен на берегу Розеттского рукава Нила. Расшифровка иероглифического текста в 1822 г. стала началом изучения египетской иероглифической письменности. – *Примеч. ред.*

Приложение D «Домены высшего уровня» – это перечисление существующих в настоящее время доменов высшего уровня сети Интернет.

Приложение E «Настройка DNS-сервера и клиента BIND» содержит справочник по синтаксису и семантике каждого из существующих параметров настройки серверов и библиотек клиента.

Для кого эта книга

Прежде всего эта книга предназначена для системных и сетевых администраторов, которым приходится управлять зонами и одним или несколькими DNS-серверами, но она содержит материал, который будет интересен проектировщикам сетей, почтовым администраторам и многим другим людям. Не все главы одинаково интересны для столь разношерстной аудитории, и, конечно же, читателю нет смысла копаться во всех шестнадцати главах, чтобы найти интересующий его материал. Мы надеемся, что следующая карта поможет выстроить правильный путь по главам книги.

Системным администраторам, впервые столкнувшимся с вопросами сопровождения зон, следует прочесть главы 1 и 2, чтобы получить теоретическую подготовку по DNS, главу 3 – в целях получения информации о первых шагах и выборе подходящего доменного имени, главы 4 и 5 – чтобы узнать, как происходит настройка зоны «с нуля». Глава 6 объясняет, как настроить хосты для работы с новыми DNS-серверами. Несколько позже следует обратиться к главе 7, в которой объясняется, как «подкачать» объем, добавляя серверы и данные в зону. Главы 12, 13 и 14 содержат описание инструментов и методов, помогающих в устранении проблем.

Опытным администраторам будет полезно прочитать главу 6, чтобы узнать, как настраивать DNS-клиенты на различных хостах, и главу 7, чтобы получить информацию о том, как грамотно сопровождать зоны. В главе 8 содержатся инструкции, связанные с планированием роста и развития зоны, которые должны быть особенно полезны людям, занятым в администрировании больших зон. Глава 9 рассказывает о том, как можно стать родителем – то есть, о создании поддоменов, и является учебником *этикета*, обязательным к прочтению теми, кто планирует совершить этот трудный шаг. В главе 10 рассмотрены многие новые возможности BIND версий 8.2.3 и 9.1.0. Глава 11 посвящена обеспечению безопасности DNS-серверов и для опытных администраторов может представлять особенный интерес. Главы с 12 по 14 содержат описание действий на случай возникновения проблем и сопутствующих инструментов, эти главы могут оказаться занимательным чтением даже для очень опытных администраторов.

Системным администраторам сетей, не имеющих постоянного подключения к сети Интернет, рекомендуется прочесть главу 5, чтобы изучить процесс настройки маршрутизации почты в таких сетях, и

главу 11, которая содержит описание создания независимой инфраструктуры DNS.

Программистам, в целях освоения теории DNS, предлагается прочесть главы 1 и 2, а затем главу 15, в которой содержится подробное рассмотрение программирования при помощи библиотечных функций BIND.

Сетевым администраторам, которые напрямую не вовлечены в процесс сопровождения зон, рекомендуется прочесть главы 1 и 2, в целях освоения теории DNS, главу 12, чтобы научиться использовать *nslookup* и *dig*, а затем главу 14, чтобы узнать о способах разрешения возникающих сложностей.

Почтовым администраторам следует прочесть главы 1 и 2, в целях освоения теории DNS, главу 5, чтобы узнать, как сосуществуют DNS и электронная почта, и главу 12, в которой описаны инструменты *nslookup* и *dig*, – эта глава научит извлекать информацию о маршрутизации почты из пространства доменных имен.

Заинтересованные пользователи могут прочесть главы 1 и 2, в целях освоения теории DNS, а затем – любые главы, по желанию!

Мы предполагаем, что читатель знаком с основами администрирования Unix-систем, сетевым взаимодействием TCP/IP, а также программированием на уровне простых сценариев командного интерпретатора или языка Perl. При этом никаких других специальных знаний не требуется. При появлении новых терминов и понятий они насколько возможно подробно объясняются в тексте книги. По возможности мы использовали аналогии с системами Unix (и реальным миром), чтобы облегчить читателю восприятие новых для него концепций.

Примеры программ

Исходные тексты программ-примеров, приводимых в книге, доступны для загрузки по протоколу FTP по следующим адресам:

ftp://ftp.uu.net/published/oreilly/nutshell/dnsbind/dns.tar.Z
ftp://ftp.oreilly.com/published/oreilly/nutshell/dnsbind/

В обоих случаях извлечь файлы из архива можно командой:

```
% zcat dns.tar.Z | tar xf -
```

На System V – системах необходимо использовать следующую *tar*-команду:

```
% zcat dns.tar.Z | tar xof -
```

Если команда *zcat* недоступна в системе, следует использовать отдельные команды *uncompress* и *tar*.

Если не удается получить тексты примеров напрямую по сети Интернет, но существует возможность посылать и получать сообщения электронной почты, можно воспользоваться службой *ftpmail*. Чтобы получить справку по использованию службы *ftpmail*, необходимо отправить сообщение на адрес *ftpmail@online.oreilly.com*. Следует оставить пустым поле темы сообщения; тело письма должно содержать единственное слово – «help».

Как связаться с издательством O'Reilly

Комментарии и вопросы, связанные с этой книгой, можно направлять непосредственно издателю:

O'Reilly & Associates, Inc.
101 Morris Street
Sebastopol, CA 95472
(800) 998-9938 (в США или Канаде)
(707) 829-0515 (международный/местный)
(707) 829-0104 (факс)

Издательством O'Reilly создана веб-страница, посвященная этой книге, на которой доступна информация о найденных ошибках и будут появляться разнообразные дополнительные сведения. Страница доступна по адресу:

http://www.oreilly.com/catalog/dns4

Если у вас есть технический вопрос или комментарий, связанный с этой книгой, задайте его, отправив сообщение по адресу:

bookquestions@oreilly.com

На веб-сайте издательства O'Reilly доступна дополнительная информация о книгах, конференциях, программном обеспечении, источниках информации и сети O'Reilly (O'Reilly Network):

http://www.oreilly.com

Типографские соглашения

Использованы следующие соглашения по шрифту и формату для команд, инструментов и системных вызовов Unix:

- Выдержки из сценариев или конфигурационных файлов оформлены моноширинным шрифтом:

```
if test -x /usr/sbin/named -a -f /etc/named.conf
then
    /usr/sbin/named
fi
```

- Примеры диалоговых сеансов, отображающие ввод в командной строке и соответствующую реакцию системы, оформлены непропорциональным шрифтом, причем ввод пользователя отмечен жирным выделением:

```
% cat /var/run/named.pid
78
```

- Если команда должна вводиться суперпользователем (администратором системы, или пользователем root), она предваряется символом диеза (#):

```
# /usr/sbin/named
```

- Заменяемые элементы кода оформлены моноширинным курсивом.
- Имена доменов, файлов, функций, команд, названия страниц руководства Unix, фрагменты кода оформлены курсивом, если они расположены внутри параграфа.

Цитаты

Цитаты из Льюиса Кэррола в каждой из глав приводятся по версии 2.9 издания Millenium Fulcrum электронного текста «Алисы в Стране чудес» из библиотеки Проекта Гутенберга (Project Gutenberg) и по изданию 1.7 текста «Алиса в Зазеркалье». Цитаты в главах 1, 2, 5, 5, 8 и 14 из «Алисы в стране чудес», а цитаты в главах 3, 4, 7, 9, 10, 11, 12, 13, 15 и 16 – из «Алисы в зазеркалье».[1]

Благодарности

Авторы выражают благодарность Кену Стоуну (Ken Stone), Джерри МакКоллому (Jerry McCollom), Питеру Джеффу (Peter Jeffe), Хэлу Стерну (Hal Stern), Кристоферу Дарему (Christopher Durham), Биллу Уизнеру (Bill Wisner), Дэйву Керри (Dave Curry), Джеффу Окамото (Jeff Okamoto), Брэду Ноулзу (Brad Knowles), Роберту Эльцу (K. Robert Elz), а также Полу Викси (Paul Vixie) за их бесценный вклад в написание этой книги. Мы также хотели бы поблагодарить наших рецензентов – Эрика Пирса (Eric Pearce), Джека Репенинга (Jack Repenning), Эндрю Черенсона (Andrew Cherenson), Дэна Тринкла (Dan Trinkle), Билла Лефевра (Bill LeFebvre) и Джона Секреста (John Sechrest) за их критику и предложения. Без помощи этих людей эта книга была бы совсем не такой (а была бы она гораздо короче!).

За второе издание этой книги авторы выражают благодарность безупречной команде рецензентов: Дэйву Бэрру (Dave Barr), Найджелу

[1] В русском издании для цитат используется перевод Нины Демуровой (М.:, ПРЕССА, 1992). – *Примеч. ред.*

Кэмпбеллу (Nigel Campbell), Биллу Лефевру, Майку Миллигану (Mike Milligan) и Дэну Тринклу.

За третье издание книги авторы отдают честь команде мечты технических рецензентов: Бобу Хэлли (Bob Halley), Барри Марголину (Barry Margolin) и Полу Викси.

Долг благодарности за четвертое издание причитается Кевину Данлэпу (Kevin Dunlap), Эдварду Льюису (Edward Lewis) и Брайану Веллингтону (Brian Wellington), первоклассной команде рецензентов.

Крикет хотел бы отдельно поблагодарить своего бывшего руководителя, Рика Норденстена (Rick Nordensten), образцового современного высокопроизводительного менеджера, под присмотром которого была написана первая версия этой книги; своих соседей, которые терпели его эпизодическую раздражительность в течение многих месяцев, и, конечно же, свою жену Пэйдж за постоянную поддержку и за то, что она мирилась с непрекращающимся, даже во время ее сна, стуком клавиш. Что касается второго издания, Крикет хотел бы добавить слова благодарности в адрес своих бывших руководителей Регины Кершнер (Regina Kershner) и Пола Клоуда (Paul Klouda) за их поддержку работы Крикета с сетью Интернет. За помощь в работе над третьим изданием Крикет считает своим долгом поблагодарить своего партнера, Мэтта Ларсона (Matt Larson), который участвовал в разработке Acme Razor; за четвертое он благодарит своих преданных пушистиков Дакоту и Энни – за их поцелуи и участие, а также замечательного Уолтера Б. (Walter B), который время от времени заглядывал в кабинет и проверял, как у Папы дела. Пол благодарит свою жену Катерину за ее терпение, за многочисленные разборы полетов и за доказательство того, что она в свободное время может гораздо быстрее сшить стеганое одеяло, чем ее муж напишет свою половину книги.

Мы хотим сказать спасибо ребятам из O'Reilly & Associates, за их тяжелый труд и терпение. В особенности этой благодарности заслуживают наши редакторы – Майк Лукидес (Mike Loukides) (издания с первого по третье) и Дебра Кэмерон (Debra Cameron) (четвертое издание), а также огромное количество других людей, которые работали над различными изданиями этой книги: Нэнси Котари (Nancy Kotary), Элли Фонтэйн Мэйден (Ellie Fountain Maden), Роберт Романо (Robert Romano), Стивен Абрамс (Steven Abrams), Кишмет МакДоноу-Чен (Kismet McDonough-Chan), Сет Мэйслин (Seth Maislin), Элли Катлер (Ellie Cutler), Майк Сьерра (Mike Sierra), Ленни Мельнер (Lenny Muellner), Крис Райли (Chris Reilley), Эмили Куилл (Emily Quill), Анна-Мария Вадува (Anne-Marie Vaduva) и Брэнда Миллер (Brenda Miller). Также спасибо Джерри Пику (Jerry Peek) за самую разнообразную поддержку, и Тиму О'Рейли за то, что он вдохновил нас на написание этой книги.

И спасибо Эди за сверчка[1] на обложке!

[1] Cricket (фамилия одного из авторов) переводится с англ. как «сверчок». – *Примеч. перев.*

9

Материнство

- *Когда заводить детей*
- *Сколько детей?*
- *Какие имена давать детям*
- *Заводим детей: создание поддоменов*
- *Поддомены in-addr.arpa*
- *Заботливые родители*
- *Как справиться с переходом к поддоменам*
- *Жизнь родителя*

А знаешь, как Дина умывала своих котят? Одной лапой она хватала бедняжку за ухо и прижимала к полу, а другой терла ей всю мордочку, начиная с носа, против шерсти. Как я уже сказал, в это время она трудилась над Снежинкой, а та лежала смирно, не сопротивлялась, да еще пыталась мурлыкать – видно, понимала, что все это делается для ее же блага.

Когда домен достигает определенных размеров, администратор приходит к мысли, что управление сегментами домена имеет смысл распределить между различными объектами организации. Домен придется разделить на поддомены. Поддомены будут детьми существующего домена в пространстве имен; домен будет их родителем. Если администратор делегирует ответственность за поддомены другим организациям, каждый из них становится отдельной зоной. Мы будем называть сопровождение поддоменов-детей – *родительскими заботами*.

Заботливый родитель начинает с разумного разделения домена, выбора подходящих имен поддоменов и делегирования поддоменов с целью создания новых зон. Ответственный родитель старается изо всех сил, чтобы сохранить хорошие отношения между зоной и детьми, он следит за тем, чтобы делегирование от родителя ребенку было актуальным и корректным.

Заботливый администратор – ключевая фигура для процветания сети, особенно когда служба имен становится незаменимой для связи между площадками. Некорректное делегирование DNS-серверам порожденной зоны может сделать узлы этой зоны попросту недоступными, а потеря связи с DNS-серверами родительской зоны может отрезать поддомен от узлов, расположенных за пределами зоны.

В этой главе мы приводим свое мнение о том, когда следует создавать поддомены, и чуть более подробно рассматриваем процесс создания и делегирования. Помимо этого мы затронем сохранение отношений родителя и ребенка, и наконец минимизацию неудобств и проблем в процессе разбиения крупных доменов на более мелкие поддомены.

Когда заводить детей

У нас и в мыслях нет *учить* читателей, когда следует заводить детей, но мы возьмем на себя смелость предложить некоторые соображения на эту тему. Кое-кто, возможно, найдет вескую причину для создания поддоменов, которой нет в этом списке, но вот наиболее распространенные:

- Необходимость делегировать или разделить управление доменом между несколькими организациями.
- Излишнее укрупнение домена – разделение облегчит сопровождение и сократит нагрузку на авторитативные DNS-серверы.
- Необходимость различать принадлежность узлов различным организациям путем включения их в соответствующие поддомены.

Решив обзавестись детьми, мы – само собой – должны спросить себя: а сколько нужно детей?

Сколько детей?

Конечно же, нельзя просто сказать: «Хочу создать четыре поддомена». Определение числа доменов связано с их предназначением. К примеру, если у компании есть четыре местных офиса, то можно создать для каждого по отдельному поддомену.

Следует ли создавать отдельный домен для каждой площадки, для каждого отдела, для каждого факультета? Масштабируемость DNS создает определенную свободу выбора. Можно создать несколько крупных поддоменов или много маленьких. В любом варианте приходится идти на определенные компромиссы.

Делегирование нескольких крупных поддоменов – задача для родителя не очень трудоемкая, поскольку не так много информации о делегировании, которую необходимо сопровождать. Однако при этом приходится работать с крупными поддоменами, которые требуют больше памяти и большей скорости работы DNS-серверов, а также не позволяют разделять работу между бо́льшим числом администраторов. Если создать для каждой из существующих площадок отдельный поддомен, автономные или неродственные группы узлов этой площадки будут вынуждены проживать в одной зоне – с централизованным администрированием.

Многочисленные делегированные поддомены могут стать головной болью для администратора родительской зоны. Сопровождение информации о делегировании связано с отслеживанием узлов и работающих на них DNS-серверов, а также с вопросами авторитативности этих серверов. Каждый раз при появлении нового DNS-сервера в поддомене или при изменении адреса DNS-сервера эта информация изменяется. Если все поддомены находятся в ведении различных людей, увеличивается число людей, которых необходимо обучать, увеличивается число связей, поддерживаемых администратором родительской зоны, увеличивается организационная нагрузка в целом. С другой стороны, поддомены, поскольку они мельче, становятся проще в сопровождении, происходит рассредоточение административных прав, и данным каждой зоны в этом случае уделяется больше внимания.

Учитывая преимущества и недостатки обоих вариантов, может быть нелегко сделать выбор. Хотя в действительности в каждой организации наверняка существует какое-то естественное деление. Некоторые компании управляют компьютерами и сетями на уровне площадки, в других существуют децентрализованные, автономные рабочие группы, которые занимаются всеми вопросами самостоятельно. Вот несколько основных правил, которые помогут определиться с разделением пространства имен:

- Не пытайтесь запихнуть организацию в неестественную или неудобную структуру. Попытки поместить 50 независимых, ничем не связанных штатов США в четыре региональных поддомена могут уменьшить трудозатраты (администратора родительской зоны), но навряд ли положительно повлияют на репутацию. Децентрализация и автономная работа требует существования многих зон – такова государственная политика DNS.
- Структура домена должна отражать структуру организации, в особенности *опорную* структуру. Если факультеты сами создают свои сети, распределяют свои IP-адреса и занимаются сопровождением своих узлов, на факультеты и должна быть возложена ответственность за сопровождение поддоменов.
- Если нет полной уверенности относительно того, какой вид должно иметь пространство имен, постарайтесь создать нормативные принципы для случаев, когда группа в пределах организации желает выделиться в отдельный поддомен (скажем, минимальное число узлов для выделения в поддомен, уровень поддержки, который должна обеспечивать эта группа) – после чего позволить пространству имен органично расти – по мере необходимости.

Какие имена давать детям

Определившись с числом поддоменов и сущностями, которым поддомены соответствуют, следует подобрать удачные имена. Не очень веж-

ливо давать доменам имен в одностороннем порядке; лучше всего привлечь к решению этого вопроса администраторов будущих поддоменов и их подданных. Более того, можно предоставить им полную свободу в этом вопросе.

Однако такая политика может приводить к осложнениям. Наиболее эффективно использовать согласованную систему именования для всех поддоменов. Это облегчает угадывание и запоминание имен поддоменов, а также поиск узлов и пользователей в различных поддоменах – как для пользователей поддомена, так и для пользователей из внешнего мира.

Если имена придумываются местными властями, это может привести к хаосу с именами. Некоторым захочется зарегистрировать «географические» имена, другие будут настаивать на именах, отражающих название организации, с которой связан домен. Одни будут заниматься сокращением, прочие станут использовать полные имена.

Таким образом, перед тем как начать давать имена поддоменам, будет логично создать правила именования. Вот некоторые соображения из нашего собственного опыта:

- В динамичной компании имена организаций могут часто меняться. В этом случае привязка поддоменов к объектам компании может привести к катастрофе. В этом месяце группа «Относительно современные технологии» выглядит достаточно стабильно, а следующем может произойти поглощение этой группы организацией «Сомнительные компьютерные системы», а в следующем квартале обе они будут приобретены немецким конгломератом. Между тем, вы уже привязаны к конкретным узлам в поддоменах, имена которых более не имеют смысла.
- Географические имена более стабильны, но иногда не столь прозрачны. Администратору, быть может, известно, что его любимое «Бизнес-подразделение компьютерного евангелизма» находится в Паукипси (Poughkeepsie) или Уокигане (Waukegan), но люди, не работающие в компании, могут понятия не иметь, где это, и испытывать сложности при написании подобных названий.
- Не приносите читабельность в жертву удобству. Двухбуквенные имена поддоменов, конечно, проще набирать, но они совершенно ни о чем не говорят. Какой смысл сокращать «Italy» (Италию) до букв «it» и путать с Организацией информационных технологий (IT), когда ценой всего лишь трех дополнительных букв можно уничтожить всякую двусмысленность и пользоваться полным названием?
- Очень многие компании используют загадочные, неудобные имена. Общее правило примерно таково: чем больше компания, тем хуже поддаются расшифровке доменные имена. Не поддавайтесь тенденции – делайте имена поддоменов самоочевидными!

- Не используйте имена существующих или зарезервированных доменов высшего уровня в качестве имен поддоменов. Использование двухбуквенных кодов стран в качестве имен международных поддоменов или использование имени вроде *net* для организации, деятельность которой связана с сетями, может показаться неплохой идеей, но также может привести и к неприятностям. Если дать поддомену отдела коммуникаций имя *com* это может затруднить взаимодействие с узлами домена высшего уровня *com*. Представим, что администратор поддомена *com* назвал новую рабочую станцию Sun именем *sun*, а новый HP 9000 – *hp* (налицо некоторые проблемы с воображением). Пользователи домена, отправляющие почтовые сообщения своим друзьям из доменов *sun.com* и *hp.com*, могут обнаружить, что их письма неожиданно попали в поддомен *com*[1], поскольку доменное имя родительской зоны может присутствовать в списке поиска одного из узлов.

Заводим детей: создание поддоменов

Когда имена придуманы, остается только создать поддомены, что довольно просто. Но прежде необходимо решить, сколько автономии дать новым поддоменам. Странно, что приходится об этом думать *до* создания поддоменов...

До сих пор мы предполагали, что после создания поддомен будет делегирован другой организации, таким образом превращаясь в самостоятельную зону. Но всегда ли это так? Необязательно.

Принимая решение о делегировании поддомена, следует тщательно обдумать вопрос сопровождения машин и сетей этого поддомена. Нет смысла делегировать поддомен объекту, который не сопровождает собственные узлы и сети. К примеру, в крупной корпорации отдел кадров, видимо, не занимается сопровождением компьютеров; скорее всего, за это отвечает отдел информационных систем управления или отдел информационных технологий. Поэтому при создании поддомена для отдела кадров делегирование управления этим поддоменом – по всей видимости, потеря времени и сил.

Создание поддоменов в родительской зоне

Итак, можно создать поддомен, но не делегировать его. Как это сделать? С помощью RR-записей, относящихся к поддомену в родительской зоне. Предположим, в зоне *movie.edu* существует узел *brazil*, на

[1] Вообще говоря, эта проблема существует не во всех почтовых программах, но она присутствует в некоторых весьма распространенных версиях *sendmail*. Все зависит от используемой формы канонизации, как мы уже говорили в разделе «Электронная почта» главы 6 «Конфигурирование узлов».

котором хранится полная база данных по служащим и студентам. Чтобы поместить *brazil* в домен *personnel.movie.edu*, можно создать соответствующие записи в файле *db.movie.edu*.

Вот фрагмент *db.movie.edu*:

```
brazil.personnel      IN  A      192.253.253.10
                      IN  MX     10 brazil.personnel.movie.edu.
                      IN  MX     100 postmanrings2x.movie.edu.
employeedb.personnel  IN  CNAME  brazil.personnel.movie.edu.
db.personnel          IN  CNAME  brazil.personnel.movie.edu.
```

Теперь пользователи могут соединяться с узлом *db.personnel.movie.edu* для получения доступа к базе данных по служащим. Мы могли бы сделать это изменение особенно удобным для работников отдела кадров, добавив *personnel.movie.edu* в списки поиска их рабочих станций; это позволит набирать *telnet db* для получения доступа к узлу базы данных.

Мы можем упростить жизнь и себе, используя директиву $ORIGIN для изменения суффикса по умолчанию на *personnel.movie.edu*.

Фрагмент файла *db.movie.edu*:

```
$ORIGIN personnel.movie.edu.
brazil     IN A     192.253.253.10
           IN MX    10 brazil.personnel.movie.edu.
           IN MX    100 postmanrings2x.movie.edu.
employeedb IN CNAME brazil.personnel.movie.edu.
db         IN CNAME brazil.personnel.movie.edu.
```

Если бы записей было больше, мы могли бы вынести их в отдельный файл и включать его в *db.movie.edu* с помощью директивы $INCLUDE (в то же время меняя локально суффикс по умолчанию).

Обратили внимание, что SOA-запись для *personnel.movie.edu* отсутствует? В ней нет необходимости, поскольку SOA-запись *movie.edu* сообщает о начале авторитативности для всей зоны *movie.edu*. Поскольку нет делегирования *personnel.movie.edu*, поддомен является частью зоны *movie.edu*.

Создание и делегирование поддомена

Когда администратор принимает решение делегировать поддомены – дать детям путевку в жизнь, – требуется несколько иной подход. Поскольку мы находимся в процессе демонстрации, читатели могут к нам присоединяться.

Необходимо создать в зоне *movie.edu* поддомен для лаборатории специальных эффектов. Мы выбрали имя *fx.movie.edu* – краткое, прозрачное, недвусмысленное. Поскольку мы делегируем *fx.movie.edu* администраторам лаборатории, поддомен будет являться самостоятельной зоной. Узлы *bladerunner* и *outland*, расположенные в лаборатории, бу-

дут выступать в качестве DNS-серверов этой зоны (причем *bladerunner* будет первичным мастер-сервером DNS). Мы решили, в целях повышения надежности, установить два DNS-сервера для этой зоны – если выйдет из строя единственный DNS-сервер *fx.movie.edu*, вся лаборатория специальных эффектов будет, по сути дела, отрезана от внешнего мира. Но поскольку в лаборатории не так уж и много узлов, двух серверов будет вполне достаточно.

Лаборатория специальных эффектов расположена в новой сети *movie.edu* – 192.253.254/24 network.

Вот фрагмент файла */etc/hosts*:

```
192.253.254.1 movie-gw.movie.edu movie-gw
# fx: первичный
192.253.254.2 bladerunner.fx.movie.edu bladerunner br
# fx: вторичный
192.253.254.3 outland.fx.movie.edu outland
192.253.254.4 starwars.fx.movie.edu starwars
192.253.254.5 empire.fx.movie.edu empire
192.253.254.6 jedi.fx.movie.edu jedi
```

Сначала создадим файл данных зоны, который содержит записи для всех узлов *fx.movie.edu*.

Содержимое файла *db.fx.movie.edu*:

```
$TTL 1d
@  IN  SOA  bladerunner.fx.movie.edu. hostmaster.fx.movie.edu. (
            1       ; порядковый номер
            3h      ; обновление
            1h      ; повторение
            1w      ; устаревание
            1h )    ; отрицательное TTL

   IN  NS  bladerunner
   IN  NS  outland

; MX-записи fx.movie.edu
   IN  MX  10 starwars
   IN  MX  100 wormhole.movie.edu.

; узел starwars обрабатывает почту узла bladerunner
; wormhole - почтовый концентратор movie.edu

bladerunner  IN  A   192.253.254.2
             IN  MX  10 starwars
             IN  MX  100 wormhole.movie.edu.

br           IN    CNAME    bladerunner

outland      IN  A   192.253.254.3
             IN  MX  10 starwars
             IN  MX  100 wormhole.movie.edu.
```

```
starwars     IN  A   192.253.254.4
             IN  MX  10 starwars
             IN  MX  100 wormhole.movie.edu.

empire       IN  A   192.253.254.5
             IN  MX  10 starwars
             IN  MX  100 wormhole.movie.edu.

jedi         IN  A   192.253.254.6
             IN  MX  10 starwars
             IN  MX  100 wormhole.movie.edu.
```

Теперь следует создать файл *db.192.253.254*:

```
$TTL 1d
@  IN  SOA  bladerunner.fx.movie.edu. hostmaster.fx.movie.edu. (
             1       ; порядковый номер
             3h      ; обновление
             1h      ; повторение
             1w      ; устаревание
             1h )    ; отрицательное TTL

       IN    NS    bladerunner.fx.movie.edu.
       IN    NS    outland.fx.movie.edu.
1      IN    PTR   movie-gw.movie.edu.
2      IN    PTR   bladerunner.fx.movie.edu.
3      IN    PTR   outland.fx.movie.edu.
4      IN    PTR   starwars.fx.movie.edu.
5      IN    PTR   empire.fx.movie.edu.
6      IN    PTR   jedi.fx.movie.edu.
```

Обратите внимание, что PTR-запись для *1.254.253.192.in-addr.arpa* указывает на имя *movie-gw.movie.edu*. Это сделано умышленно. Этот маршрутизатор связан и с другими сетями *movie.edu* и в действительности не принадлежит *fx.movie.edu*; к тому же, не существует правила, по которому все PTR-записи в *254.253.192.in-addr.arpa* должны отображать адреса в единственную зону – хотя все они должны соответствовать каноническим именам узлов.

Затем мы создаем файл *named.conf* для первичного мастер-сервера DNS:

```
options {
                directory "/var/named";
};

zone "0.0.127.in-addr.arpa" {
               type master;
               file "db.127.0.0";
};

zone "fx.movie.edu" {
               type master;
               file "db.fx.movie.edu";
};
```

```
zone "254.253.192.in-addr.arpa" {
                type master;
                file "db.192.253.254";
};

zone "." {
                type hint;
                file "db.cache";
};
```

Аналогичный файл *named.boot* для BIND 4:

```
directory     /var/named

primary       0.0.127.in-addr.arpa      db.127.0.0  ; loopback
primary       fx.movie.edu              db.fx.movie.edu
primary       254.253.192.in-addr.arpa  db.192.253.254

cache         .                         db.cache
```

Разумеется, если бы мы пользовались программой *h2n*, то могли бы просто выполнить команду:

```
% h2n -d fx.movie.edu -n 192.253.254 -s bladerunner -s outland \
-u hostmaster.fx.movie.edu -m 10:starwars -m 100:wormhole.movie.edu
```

и сэкономить время. В результате мы получили бы – по сути дела – такие же файлы *db.fx.movie.edu*, *db.192.253.254* и *named.boot*.

Теперь следует произвести настройку DNS-клиента узла *bladerunner*. Может оказаться, что нет необходимости создавать файл *resolv.conf*. Если мы установим значение *hostname* для узла *bladerunner* в новое доменное имя этого узла, *bladerunner.fx.movie.edu*, клиент сможет извлечь локальное доменное имя из абсолютного.

Теперь мы запускаем процесс *named* на узле *bladerunner* и проверяем log-файл *syslog* на наличие ошибок. Если *named* стартовал нормально, а в log-файле *syslog* нет ошибок, требующих немедленного исправления, используем *nslookup* для поиска данных для нескольких узлов в зонах *fx.movie.edu* и *254.253.192.in-addr.arpa* :

```
Default Server:  bladerunner.fx.movie.edu
Address:  192.253.254.2

> jedi
Server:  bladerunner.fx.movie.edu
Address:  192.253.254.2

Name:    jedi.fx.movie.edu
Address:  192.253.253.6

> set type=mx
> empire
Server:  bladerunner.fx.movie.edu
Address:  192.253.254.2
```

```
empire.fx.movie.edu     preference = 10,
                        mail exchanger = starwars.fx.movie.edu
empire.fx.movie.edu     preference = 100,
                        mail exchanger = wormhole.movie.edu
fx.movie.edu    nameserver = outland.fx.movie.edu
fx.movie.edu    nameserver = bladerunner.fx.movie.edu
starwars.fx.movie.edu   internet address = 192.253.254.4
wormhole.movie.edu      internet address = 192.249.249.1
wormhole.movie.edu      internet address = 192.253.253.1
bladerunner.fx.movie.edu        internet address = 192.253.254.2
outland.fx.movie.edu    internet address = 192.253.254.3
> ls -d fx.movie.edu
 [bladerunner.fx.movie.edu]
$ORIGIN fx.movie.edu.
@                       1D IN SOA       bladerunner hostmaster (
                                        1               ; порядковый номер
                                        3H              ; обновление
                                        1H              ; повторение
                                        1W              ; устаревание
                                        1H )            ; минимум

                        1D IN NS        bladerunner
                        1D IN NS        outland
                        1D IN MX        10 starwars
                        1D IN MX        100 wormhole.movie.edu.
bladerunner             1D IN A         192.253.254.2
                        1D IN MX        10 starwars
                        1D IN MX        100 wormhole.movie.edu.
br                      1D IN CNAME     bladerunner
empire                  1D IN A         192.253.254.5
                        1D IN MX        10 starwars
                        1D IN MX        100 wormhole.movie.edu.
jedi                    1D IN A         192.253.254.6
                        1D IN MX        10 starwars
                        1D IN MX        100 wormhole.movie.edu.
outland                 1D IN A         192.253.254.3
                        1D IN MX        10 starwars
                        1D IN MX        100 wormhole.movie.edu.
starwars                1D IN A         192.253.254.4
                        1D IN MX        10 starwars
                        1D IN MX        100 wormhole.movie.edu.
@                       1D IN SOA       bladerunner hostmaster (
                                        1               ; порядковый номер
                                        3H              ; обновление
                                        1H              ; повторение
                                        1W              ; устаревание
                                        1H )            ; минимум
> set type=ptr
> 192.253.254.3
Server:  bladerunner.fx.movie.edu
Address:  192.253.254.2
```

```
3.254.253.192.in-addr.arpa      name = outland.fx.movie.edu

> ls -d 254.253.192.in-addr.arpa.
 [bladerunner.fx.movie.edu]
$ORIGIN 254.253.192.in-addr.arpa.
@          1D IN SOA     bladerunner.fx.movie.edu. hostmaster.fx.movie.edu. (
                           1              ; порядковый номер
                           3H             ; обновление
                           1H             ; повторение
                           1W             ; устаревание
                           1H )           ; минимум

            1D IN NS      bladerunner.fx.movie.edu.
            1D IN NS      outland.fx.movie.edu.
1           1D IN PTR     movie-gw.movie.edu.
2           1D IN PTR     bladerunner.fx.movie.edu.
3           1D IN PTR     outland.fx.movie.edu.
4           1D IN PTR     starwars.fx.movie.edu.
5           1D IN PTR     empire.fx.movie.edu.
6           1D IN PTR     jedi.fx.movie.edu.
@          1D IN SOA     bladerunner.fx.movie.edu. hostmaster.fx.movie.edu. (
                           1              ; порядковый номер
                           3H             ; обновление
                           1H             ; повторение
                           1W             ; устаревание
                           1H )           ; минимум
> exit
```

Вывод выглядит нормально, поэтому можно приступать к установке вторичного DNS-сервера для зоны *fx.movie.edu*, а затем переходить к делегированию *fx.movie.edu*.

Вторичный DNS–сервер fx.movie.edu

Вторичный DNS-сервер для зоны *fx.movie.edu* устанавливается без сложностей: следует скопировать файлы *named.conf*, *db.127.0.0* и *db.cache* с узла *bladerunner*, а затем отредактировать *named.conf* и *db.127.0.0* в соответствии с инструкциями, приводимыми в главе 4 «Установка BIND».

Содержимое файла *named.conf*:

```
options {
                directory "/var/named";
};

zone "0.0.127.in-addr.arpa" {
                type master;
                file "db.127.0.0";
};

zone "fx.movie.edu" {
                type slave;
```

```
                masters { 192.253.254.2; };
                file "bak.fx.movie.edu";
};

zone "254.253.192.in-addr.arpa" {
                type slave;
                masters { 192.253.254.2; };
                file "bak.192.253.254";
};

zone "." {
                type hint;
                file "db.cache";
};
```

Эквивалентный файл *named.boot*:

```
directory  /var/named

primary    0.0.127.in-addr.arpa      db.127.0.0
secondary  fx.movie.edu              192.253.254.2  bak.fx.movie.edu
secondary  254.253.192.in-addr.arpa  192.253.254.2  bak.192.253.254

cache      .                         db.cache
```

Как и в случае с узлом *bladerunner*, на узле *outland* не нужен файл *resolv.conf* – если значение *hostname* установлено в *outland.fx.movie.edu*.

И снова – запускаем *named* и сверяемся с log-файлом *syslog* на предмет наличия в нем сообщений об ошибках. Если ошибок нет, можно переходить к поиску записей в *fx.movie.edu*.

Первичный DNS-сервер movie.edu

Осталось только делегировать поддомен *fx.movie.edu* новым DNS-серверам *fx.movie.edu*, работающим на узлах *bladerunner* и *outland*. Добавляем соответствующие NS-записи в файл *db.movie.edu*.

Фрагмент файла *db.movie.edu*:

```
fx      86400    IN    NS    bladerunner.fx.movie.edu.
        86400    IN    NS    outland.fx.movie.edu.
```

Документ RFC 1034 утверждает, что доменные имена в правой части записей в приводимых двух строках (*bladerunner.fx.movie.edu* и *outland.fx.movie.edu*) должны являться каноническими доменными именами DNS-серверов. Удаленный DNS-сервер, использующий для навигации информацию о делегировании, ожидает найти одну или несколько адресных записей, связанных с таким доменным именем, а не запись псевдонима (CNAME). Вообще говоря, этот RFC-документ распространяет данное ограничение на любой тип записей, включающий доменное имя в качестве значения – это значение должно являться каноническим доменным именем.

Но этих двух записей недостаточно. Видите, в чем проблема? Как может DNS-сервер за пределами *fx.movie.edu* производить поиск информации из *fx.movie.edu*? Ведь DNS-сервер *movie.edu* направит внешний сервер к DNS-серверам, авторитативным для зоны *fx.movie.edu*? Разумеется, но NS-записи в файле *db.movie.edu* содержат только *имена* DNS-серверов зоны *fx.movie.edu*. Серверу-иностранцу понадобятся IP-адреса DNS-серверов *fx.movie.edu*, чтобы послать им запросы. Кто может дать ему эти адреса? Только DNS-серверы зоны *fx.movie.edu*. Что было раньше – курица или яйцо?

Вот решение: адреса DNS-серверов *fx.movie.edu* следует включить в файл данных зоны *movie.edu*. Такая информация не принадлежит, строго говоря, зоне *movie.edu*, но она необходима, чтобы работало делегирование для *fx.movie.edu*. Разумеется, если DNS-серверы *fx.movie.edu* находились бы не в пределах *fx.movie.edu*, эта информация – называемая связующими записями (glue records) – не потребовалась бы. Сервер-иностранец нашел бы требуемые адреса, сделав несколько запросов к другим DNS-серверам.

Итак, в комплекте со связующими записями, фрагмент файла *db.movie.edu* выглядит следующим образом:

```
fx      86400    IN    NS    bladerunner.fx.movie.edu.
        86400    IN    NS    outland.fx.movie.edu.
bladerunner.fx.movie.edu.  86400  IN  A  192.253.254.2
outland.fx.movie.edu.      86400  IN  A  192.253.254.3
```

Не следует включать в файл лишние связующие записи. Более старые версии (до 4.9) загружают эти записи в кэш и выдают их в ответах другим DNS-серверам. Если DNS-сервер из адресной записи сменил IP-адрес, а администратор забыл обновить связующие записи, то DNS-сервер будет возвращать устаревшую адресную информацию, что приведет к понижению скорости разрешения для DNS-серверов, интересующихся данными из делегированной зоны, или просто лишит их возможности получать эти данные.

DNS-сервер BIND версии 4.9 или более поздней автоматически игнорирует связующие записи, которые не являются строго необходимыми, и заносит в log-файл *syslog* сообщение о том, что записи были проигнорированы. Так, если бы у нас была NS-запись для *movie.edu*, указывающая на внешний DNS-сервер, *ns-1.isp.net*, и мы сделали бы ошибку, включив адресную запись для этого сервера в файл *db.movie.edu* на первичном мастер-сервере DNS зоны *movie.edu*, то увидели бы примерно такое сообщение в выводе *syslog*:

```
Aug  9 14:23:41 terminator named[19626]: dns_master_load: db.movie.edu:55:
ignoring out-of-zone data
```

Если бы мы работали с DNS-сервером версии более ранней, чем 4.9, и он, по ошибке, включил бы ненужные связующие записи в данные

при передаче данных DNS-серверу более поздней версии, мы могли бы увидеть примерно следующее сообщение в резервной копии файла данных зоны:

```
; Ignoring info about ns-1.isp.net, not in zone movie.edu
; ns-1.isp.net 258983  IN      A       10.1.2.3
```

Обратите внимание, что лишняя адресная запись закомментирована.

И помните, что связующие записи должны быть актуальными. Если *bladerunner* обзаводится новым сетевым интерфейсом – а значит, и новым IP-адресом – следует добавить еще одну адресную запись в связующие данные.

Мы могли бы также создать псевдонимы для любых узлов, осуществляющих переезд из *movie.edu* в *fx.movie.edu*. К примеру, если необходимо переместить *plan9.movie.edu* (сервер, хранящий важную библиотеку свободно доступных алгоритмов специальных эффектов) в зону *fx.movie.edu*, следует создать псевдоним в *movie.edu*, связывающий старое доменное имя с новым:

```
plan9           IN      CNAME   plan9.fx.movie.edu.
```

Это позволит пользователям вне зоны *movie.edu* получать доступ к узлу *plan9*, используя даже прежнее доменное имя *plan9.movie.edu*.

Не следует размещать какую-либо информацию о доменных именах зоны *fx.movie.edu* в файле *db.movie.edu*. Псевдоним *plan9* в действительности принадлежит зоне *movie.edu* (запись связана с именем *plan9.movie.edu*), так что ему место в файле *db.movie.edu*. С другой стороны, псевдоним *p9.fx.movie.edu* для узла *plan9.fx.movie.edu* принадлежит зоне *fx.movie.edu* и должен содержаться в файле *db.fx.movie.edu*. Случись администратору поместить в файл данных зоны «чужую» запись, DNS-сервер BIND 4.9 или более поздней версии проигнорирует ее, как показано в примере с ненужными связующими записями. Более старый DNS-сервер может загрузить запись в кэш и даже поместить ее в свои авторитативные данные, но поскольку такое поведение приводит к непредсказуемым результатам и не реализовано в более новых версиях BIND, лучше всего придерживаться правильного способа, даже если этот способ не навязывается программным пакетом.

Делегирование зоны in-addr.arpa

Мы чуть не забыли делегировать зону 254.253.192.in-addr.arpa! Здесь сложностей чуть больше, чем при делегировании *fx.movie.edu*, поскольку администрирование родительской зоны нам недоступно.

Во-первых, необходимо узнать, в какую родительскую зону входит *254.253.192.in-addr.arpa* и кто занимается сопровождением этой зоны. Получение этой информации может быть связано с сыскным делом, но мы уже рассматривали этот вопрос в главе 3 «С чего начать?».

Выясняется, что родительской зоной для *254.253.192.in-addr.arpa* является *in-addr.arpa*. Если подумать, в этом есть определенный смысл. Администраторы *in-addr.arpa* не видят особого смысла в делегировании зон *253.192.in-addr.arpa* и *192.in-addr.arpa* отдельным инстанциям, поскольку сети вроде 192.253.253/24 и 192.253.254/24 не имеют ничего общего, если 192/8 или 192.253/16 не является одним большим CIDR-блоком. Ими могут заведовать совершенно не связанные между собой организации.

Возможно читатели помнят (из главы 3), что сопровождением зоны *in-addr.arpa* занимается организация ARIN (American Registry of Internet Numbers). Если бы вы этого не помнили, то могли бы воспользоваться программой *nslookup* для поиска контактного адреса из SOA-записи зоны *in-addr.arpa*, как мы демонстрировали в той же самой главе. Нам осталось только воспользоваться веб-интерфейсом «Modify Tool» (Инструмент регистрации изменений) по адресу *http://www.arin.net/cgi-bin/amt.pl*, чтобы запросить регистрацию зоны обратного отображения.

Еще один вторичный DNS-сервер movie.edu

Если лаборатория специальных эффектов станет достаточно большой, возможно будет иметь смысл разместить вторичный DNS-сервер *movie.edu* в сети 192.253.254/24. В этом случае разрешение большей доли DNS-запросов, создаваемых узлами *fx.movie.edu*, будет происходить локально. Кажется логичным сделать один из существующих DNS-серверов зоны *fx.movie.edu* вторичным сервером *movie.edu* – так мы сможем более полно использовать уже существующий сервер, вместо того чтобы создавать еще один.

Мы решили сделать вторичным DNS-сервером *movie.edu* узел *bladerunner*. Это не помешает работе узла *bladerunner* в качестве первичного мастер-сервера DNS зоны *fx.movie.edu*. Единственный DNS-сервер, если снабдить его достаточным количеством памяти, может быть авторитативным буквально для тысяч зон. Один и тот же DNS-сервер может выступать для одних зон в качестве первичного, а для других в качестве вторичного.[1]

Изменения в настройке минимальны: к файлу *named.conf* узла *bladerunner* добавляется один оператор, который сообщает серверу *named*, что следует загружать зону *movie.edu* с IP-адреса первичного мастер-сервера DNS *movie.edu*, который называется *terminator.movie.edu*.

Содержимое файла *named.conf*:

```
options {
```

[1] При этом очевидно, что сервер имен не может быть первичным мастером и вторичным для одной и той же зоны. DNS-сервер либо получает данные для зоны от другого сервера (и тогда он является вторичным), либо из локального файла данных зоны (и тогда он является первичным).

```
    directory "/var/named";
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "db.127.0.0";
};

zone "fx.movie.edu" {
    type master;
    file "db.fx.movie.edu";
};

zone "254.253.192.in-addr.arpa" {
    type master;
    file "db.192.253.254";
};

zone "movie.edu" {
    type slave;
    masters { 192.249.249.3; };
    file "bak.movie.edu";
};

zone "." {
    type hint;
    file "db.cache";
};
```

Если используется DNS-сервер BIND 4, то содержимое файла *named.boot*:

```
directory     /var/named

primary       0.0.127.in-addr.arpa      db.127.0.0  ; обратная связь
primary       fx.movie.edu              db.fx.movie.edu
primary       254.253.192.in-addr.arpa  db.192.253.254
secondary     movie.edu                 192.249.249.3     bak.movie.edu

cache         .                         db.cache
```

Поддомены доменов in-addr.arpa

Делить на поддомены и делегировать можно не только домены прямого отображения. Если сегмент пространства имен *in-addr.arpa* достаточно крупный, может возникнуть необходимость в его разделении. Обычно домен, соответствующий номеру сети, делится на поддомены, соответствующие подсетям. Механизм работы зависит от типа существующей сети и маски подсети.

Формирование подсети на границе октета

Поскольку в Университете кинематографии всего три сети /24 (класса C), по одной на сегмент, нет особой необходимости в формировании подсетей. Однако наш университет-побратим, Altered State, имеет сеть класса B, 172.20/16. Сеть этого университета разделена на подсети между третьим и четвертым октетом IP-адреса; то есть маска подсети – 255.255.255.0. Этот университет уже создал несколько поддоменов в своем домене *altered.edu*, в частности *fx.altered.edu* (признаемся, мы просто следовали их примеру), *makeup.altered.edu* и *foley.altered.edu*. Поскольку каждый из этих факультетов имеет собственную подсеть (факультет Spesial Effects – подсеть 172.20.2/24, факультет Makeup – 172.20.15/24, а Foley – 172.20.25/24), они хотели бы соответствующим образом разделить и пространство имен *in-addr.arpa*.

Делегирование поддоменов *in-addr.arpa* ничем не отличается от делегирования поддоменов доменов прямого отображения. В файле данных зоны *db.172.20* университету Altered State понадобится создать примерно такие записи:

```
2     86400    IN    NS    gump.fx.altered.edu.
2     86400    IN    NS    toystory.fx.altered.edu.
15    86400    IN    NS    prettywoman.makeup.altered.edu.
15    86400    IN    NS    priscilla.makeup.altered.edu.
25    86400    IN    NS    blowup.foley.altered.edu.
25    86400    IN    NS    muppetmovie.foley.altered.edu.
```

делегируя поддомены, соответствующие отдельным подсетям, DNS-серверам в различных поддоменах.

Несколько важных замечаний: администраторы Altered State могли использовать в поле имени владельца записи только третий октет подсети, поскольку суффикс по умолчанию для этого файла – *20.172.in-addr.arpa*. При этом им пришлось использовать в правой части NS-записей абсолютные доменные имена, чтобы избежать добавления к ним суффикса по умолчанию. И они *не* добавили связующие записи, поскольку имена DNS-серверов, которым делегируется зона, не заканчиваются доменным именем этой зоны.

Формирование подсети не на границе октета

Что делать с сетями, подсети в которых формируются не на границах октетов, как в случае сети /24 (сети класса C)? В таких случаях можно производить делегирование по границам подсетей. Это приводит к одной из двух ситуаций: существует несколько подсетей в каждой зоне *in-addr.arpa* либо существует много зон *in-addr.arpa* для каждой подсети. Оба варианта достаточно неприятные.

Сети классов A и B

Возьмем случай сети /8 (сеть класса A) – 15/8, подсети в которой формируются маской 255.255.248.0 (13-битное поле подсети и 11-битное поле узла, 8192 подсети по 2048 узлов). В таком случае, скажем, сеть 15.1.200.0 охватывает диапазон адресов с 15.1.200.0 по 15.1.207.255. Следовательно, делегирование для одного этого поддомена в *db.15*, файле данных зоны для *15.in-addr.arpa*, может выглядеть следующим образом:

```
200.1.15.in-addr.arpa.    86400    IN    NS    ns-1.cns.hp.com.
200.1.15.in-addr.arpa.    86400    IN    NS    ns-2.cns.hp.com.
201.1.15.in-addr.arpa.    86400    IN    NS    ns-1.cns.hp.com.
201.1.15.in-addr.arpa.    86400    IN    NS    ns-2.cns.hp.com.
202.1.15.in-addr.arpa.    86400    IN    NS    ns-1.cns.hp.com.
202.1.15.in-addr.arpa.    86400    IN    NS    ns-2.cns.hp.com.
203.1.15.in-addr.arpa.    86400    IN    NS    ns-1.cns.hp.com.
203.1.15.in-addr.arpa.    86400    IN    NS    ns-2.cns.hp.com.
204.1.15.in-addr.arpa.    86400    IN    NS    ns-1.cns.hp.com.
204.1.15.in-addr.arpa.    86400    IN    NS    ns-2.cns.hp.com.
205.1.15.in-addr.arpa.    86400    IN    NS    ns-1.cns.hp.com.
205.1.15.in-addr.arpa.    86400    IN    NS    ns-2.cns.hp.com.
206.1.15.in-addr.arpa.    86400    IN    NS    ns-1.cns.hp.com.
206.1.15.in-addr.arpa.    86400    IN    NS    ns-2.cns.hp.com.
207.1.15.in-addr.arpa.    86400    IN    NS    ns-1.cns.hp.com.
207.1.15.in-addr.arpa.    86400    IN    NS    ns-2.cns.hp.com.
```

Для одной подсети – довольно объемная информация о делегировании!

К счастью, начиная с версии 8.2 серверы BIND реализуют поддержку директивы $GENERATE. $GENERATE позволяет создавать группу RR-записей, которые отличаются только численным итератором. К примеру, 16 только что перечисленных NS-записей можно создать следующей парой директив $GENERATE:

```
$GENERATE 200-207 $.1.15.in-addr.arpa.  86400  IN  NS  ns-1.cns.hp.com.
$GENERATE 200-207 $.1.15.in-addr.arpa.  86400  IN  NS  ns-2.cns.hp.com.
```

Синтаксис оператора довольно прост: DNS-сервер создает набор записей из оператора $GENERATE, заменяя символ доллара ($) поочередно числами из диапазона, определенного в первом поле оператора.

Сети класса C

Рассмотрим случай сети /24 (класса C), например 192.253.254/24, в которой формирование сетей производится на основе маски 255.255.255.192. В данном случае существует единственная зона *in-addr.arpa*, *254.253.192.in-addr.arpa*, которая соответствует подсетям 192.253.254.0/26, 192.253.254.64/26, 192.253.254.128/26 и 192.253.254.192/26. Это может быть проблемой, если необходимо разрешить различным организациям сопровождение информации обрат-

ного отображения для этих сетей. Проблема решается одним из трех некрасивых способов.

Решение 1

Первое решение: администрировать зону *254.253.192.in-addr.arpa* в качестве единой сущности, даже не думая о делегировании. Для этого требуется сотрудничество администраторов четырех подсетей либо применение инструмента вроде Webmin (*http://www.webmin.com/webmin*), который позволит каждому из администраторов сопровождать собственный раздел данных.

Решение 2

Второе решение: произвести делегирование на *четвертом* октете. Это даже хуже, чем делегирование для сети /8, которое мы уже продемонстрировали. Нужна хотя бы пара NS-записей на каждый IP-адрес в файле *db.192.253.254*. Это выглядит примерно так:

```
1.254.253.192.in-addr.arpa.    86400    IN    NS    ns1.foo.com.
1.254.253.192.in-addr.arpa.    86400    IN    NS    ns2.foo.com.

2.254.253.192.in-addr.arpa.    86400    IN    NS    ns1.foo.com.
2.254.253.192.in-addr.arpa.    86400    IN    NS    ns2.foo.com.

...

65.254.253.192.in-addr.arpa.    86400    IN    NS    relay.bar.com.
65.254.253.192.in-addr.arpa.    86400    IN    NS    gw.bar.com.

66.254.253.192.in-addr.arpa.    86400    IN    NS    relay.bar.com.
66.254.253.192.in-addr.arpa.    86400    IN    NS    gw.bar.com.

...

129.254.253.192.in-addr.arpa.    86400    IN    NS    mail.baz.com.
129.254.253.192.in-addr.arpa.    86400    IN    NS    www.baz.com.

130.254.253.192.in-addr.arpa.    86400    IN    NS    mail.baz.com.
130.254.253.192.in-addr.arpa.    86400    IN    NS    www.baz.com.
```

и так далее, вплоть до *254.254.253.192.in-addr.arpa*.

Можно существенно сократить объем, воспользовавшись оператором $GENERATE:

```
$GENERATE 0-63 $.254.253.192.in-addr.arpa  86400  IN  NS  ns1.foo.com.
$GENERATE 0-63 $.254.253.192.in-addr.arpa  86400  IN  NS  ns2.foo.com.

$GENERATE 64-127 $.254.253.192.in-addr.arpa.  86400  IN  NS  relay.bar.com.
$GENERATE 64-127 $.254.253.192.in-addr.arpa.  86400  IN  NS  gw.bar.com.

$GENERATE 128-191 $.254.253.192.in-addr.arpa.  86400  IN  NS  mail.baz.com.
$GENERATE 128-191 $.254.253.192.in-addr.arpa.  86400  IN  NS  www.baz.com.
```

Конечно, подразумевается, что в файле *named.conf* на узле *ns1.foo.com* присутствует следующий фрагмент:

```
zone "1.254.253.192.in-addr.arpa" {
                type master;
                file "db.192.253.254.1";
};

zone "2.254.253.192.in-addr.arpa" {
                type master;
                file "db.192.253.254.2";
};
```

Если же на *ns1.foo.com* используется BIND 4, то следующие инструкции присутствуют в файле *named.boot*:

```
primary    1.254.253.192.in-addr.arpa    db.192.253.254.1
primary    2.254.253.192.in-addr.arpa    db.192.253.254.2
```

а в файле *db.192.253.254.1* – одна-единственная PTR-запись:

```
$TTL 1d
@    IN    SOA    ns1.foo.com.    root.ns1.foo.com.    (
                        1        ; Порядковый номер
                        3h       ; Обновление
                        1h       ; Повторение
                        1w       ; Устаревание
                        1h       ; Отрицательное TTL

     IN    NS    ns1.foo.com.
     IN    NS    ns2.foo.com.

     IN    PTR    thereitis.foo.com.
```

Обратите внимание, что эта PTR-запись связана с доменным именем зоны, поскольку доменное имя зоны соответствует всего лишь одному IP-адресу. Теперь, получая запрос PTR-записи для имени *1.254.253.192.in-addr.arpa*, DNS-сервер зоны *254.253.192.in-addr.arpa* направляет клиент к серверам *ns1.foo.com* и *ns2.foo.com*, которые возвращают именно эту, единственную в зоне, PTR-запись.

Решение 3

И наконец, существует более умный способ, избавляющий нас от необходимости сопровождать отдельный файл данных зоны для каждого IP-адреса.[1] Организация, отвечающая за всю сеть /24, создает CNAME-записи для каждого из доменных имен зоны; эти CNAME-записи указывают на доменные имена в новых поддоменах, которые, в свою оче-

[1] Впервые мы увидели эту идею в конференции *comp.protocols.tcp-ip.domains* в изложении Глена Херрмансфельдта (Glen Herrmansfeldt) из КалТех. В настоящее время способ стандартизирован в документе RFC 2317.

редь, делегируются различным DNS-серверам. Новые поддомены могут иметь практически любые имена, например *0-63*, *64-127*, *128-191* и *192-255*, четко показывающие диапазоны адресов, для которых производится обратное отображение в каждом из доменов. При этом каждый поддомен содержит только PTR-записи для указанного диапазона.

Фрагмент файла *db.192.253.254*:

```
1.254.253.192.in-addr.arpa.  IN  CNAME  1.0-63.254.253.192.in-addr.arpa.
2.254.253.192.in-addr.arpa.  IN  CNAME  2.0-63.254.253.192.in-addr.arpa.

...

0-63.254.253.192.in-addr.arpa.    86400    IN    NS    ns1.foo.com.
0-63.254.253.192.in-addr.arpa.    86400    IN    NS    ns2.foo.com.

65.254.253.192.in-addr.arpa. IN  CNAME 65.64-127.254.253.192.in-addr.arpa.
66.254.253.192.in-addr.arpa. IN  CNAME 66.64-127.254.253.192.in-addr.arpa.

...

64-127.254.253.192.in-addr.arpa.    86400    IN    NS    relay.bar.com.
64-127.254.253.192.in-addr.arpa.    86400    IN    NS    gw.bar.com.

129.254.253.192.in-addr.arpa.  IN  CNAME  129.128-191.254.253.192.in-addr.
arpa.
130.254.253.192.in-addr.arpa.  IN  CNAME  130.128-191.254.253.192.in-addr.
arpa.

...

128-191.254.253.192.in-addr.arpa.    86400    IN    NS    mail.baz.com.
128-191.254.253.192.in-addr.arpa.    86400    IN    NS    www.baz.com.
```

И опять можно использовать $GENERATE для сокращения:

```
$GENERATE 1-63 $ IN CNAME $.0-63.254.253.192.in-addr.arpa.

0-63.254.253.192.in-addr.arpa.    86400    IN    NS    ns1.foo.com.
0-63.254.253.192.in-addr.arpa.    86400    IN    NS    ns2.foo.com.

$GENERATE 65-127 $ IN CNAME $.64-127.254.253.192.in-addr.arpa.

64-127.254.253.192.in-addr.arpa.    86400    IN    NS    relay.bar.com.
64-127.254.253.192.in-addr.arpa.    86400    IN    NS    gw.bar.com.
```

Файл данных для зоны *0-63.254.253.192.in-addr.arpa* (*db.192.253.254.0-63*) вполне может содержать только PTR-записи для IP-адресов с 192.253.254.1 по 192.253.254.63.

Фрагмент файла *db.192.253.254.0-63*:

```
$TTL 1d
@    IN    SOA    ns1.foo.com.    root.ns1.foo.com.    (
                          1       ; Порядковый номер
                          3h      ; Обновление
                          1h      ; Повторение
```

```
                                1w      ; Устаревание
                                1h )    ; Отрицательное TTL

        IN      NS      ns1.foo.com.
        IN      NS      ns2.foo.com.

1       IN      PTR     thereitis.foo.com.
2       IN      PTR     setter.foo.com.
3       IN      PTR     mouse.foo.com.
...
```

Работа этого варианта не очень прозрачна, поэтому рассмотрим процесс несколько подробнее. Клиент запрашивает у локального DNS-сервера PTR-запись для *1.254.253.192.in-addr.arpa*. Локальный DNS-сервер в итоге добирается до DNS-сервера *254.253.192.in-addr.arpa*, который возвращает CNAME-запись, сообщающую, что *1.254.253.192.in-addr.arpa* в действительности является всего лишь псевдонимом для *1.0-63.254.253.192.in-addr.arpa* и что PTR-запись связана с последним именем. Ответ также содержит NS-записи, которые говорят локальному DNS-серверу, что авторитативными серверами для *0-63.254.253.192.in-addr.arpa* являются *ns1.foo.com* и *ns2.foo.com*. Локальный DNS-сервер посылает запрос PTR-записи для *1.0-63.254.253.192.in-addr.arpa* – DNS-серверу *ns1.foo.com* или *ns2.foo.com*, и получает искомое.

Заботливые родители

Теперь, разобравшись с делегированием DNS-серверам *fx.movie.edu*, мы – как заботливые родители – должны проверить делегирование с помощью программы *host*. Как? Мы еще не поделились с читателями программой *host*? Версия *host* для Unix-систем доступна для анонимного FTP-копирования с сервера *ftp.nikhef.nl* (имя файла – */pub/network/host.tar.Z*).

Чтобы собрать программу *host*, следует сначала распаковать архив:

```
% zcat host.tar.Z | tar -xvf -
```

А затем выполнить команду:

```
% make
```

host облегчает проверку делегирования. С помощью этого инструмента можно производить поиск NS-записей для зоны, используя DNS-серверы зоны-родителя. Если с этими записями все в порядке, можно использовать *host* для посылки запросов каждому из DNS-серверов, перечисленных для SOA-записи зоны. Запросы нерекурсивны, поэтому DNS-сервер, к которому произошло обращение, не контактирует с другими DNS-серверами в поисках SOA-записи. Когда DNS-сервер возвращает ответ, *host* проверяет ответ на предмет наличия установленного

бита *aa* – authoritative answer (авторитативный ответ). В случае положительного результата DNS-сервер проверяет ответное сообщение на присутствие собственно ответа. При выполнении этой пары условий DNS-сервер помечается как авторитативный для зоны. В противном случае сервер не является авторитативным, и *host* сообщает об ошибке.

Почему столько суматохи вокруг некорректного делегирования? Дело в том, что оно может замедлять разрешение имен или приводить к распространению устаревшей информации о корневых DNS-серверах. Когда сервер получает запрос данных из зоны, для которой он не является авторитативным, то прилагает все усилия, чтобы сообщить клиенту полезную информацию по теме. Эта «полезная информация» содержится в NS-записях ближайшей зоны-предка, которая известна DNS-серверу. (Мы вкратце поговорили об этом в главе 8 «Развитие домена», когда обсуждали, почему не следует регистрировать DNS-сервер, специализирующийся на кэшировании.)

Допустим, один из DNS-серверов *fx.movie.edu* по ошибке получает итеративный запрос адреса *carrie.horror.movie.edu*. Серверу ничего не известно о зоне *horror.movie.edu* (хотя возможно существуют какие-то данные в кэше), но, вполне вероятно, он кэшировал NS-записи для *movie.edu*, поскольку эти записи содержат информацию о родительской зоне. И эти записи DNS-сервер возвращает автору запроса.

В описанном сценарии возвращаемые NS-записи могут помочь DNS-серверу, от которого исходил запрос, найти ответ. Однако в сети Интернет правда жизни заключается в том, что не все администраторы следят за тем, чтобы файлы корневых указателей соответствовали реальности. Если один из наших DNS-серверов использует для навигации некорректную информацию о делегировании и запросит у удаленного DNS-сервера данные, которых у того нет, может произойти следующее:

```
% nslookup
Default Server:  terminator.movie.edu
Address:  192.249.249.3
> set type=ns
> .
Server:  terminator.movie.edu
Address:  192.249.249.3

Non-authoritative answer:
 (root)  nameserver = D.ROOT-SERVERS.NET
 (root)  nameserver = E.ROOT-SERVERS.NET
 (root)  nameserver = I.ROOT-SERVERS.NET
 (root)  nameserver = F.ROOT-SERVERS.NET
 (root)  nameserver = G.ROOT-SERVERS.NET
 (root)  nameserver = A.ROOT-SERVERS.NET
 (root)  nameserver = H.ROOT-SERVERS.NET
 (root)  nameserver = B.ROOT-SERVERS.NET
 (root)  nameserver = C.ROOT-SERVERS.NET
 (root)  nameserver = A.ISI.EDU          – Эти три сервера
```

```
(root)  nameserver = SRI-NIC.ARPA        — уже не являются
(root)  nameserver = GUNTER-ADAM.ARPA    — корневыми
```

Удаленный сервер попытался «помочь» нашему DNS-серверу, послав ему текущий перечень корневых серверов. К сожалению, удаленный DNS-сервер не шел в ногу со временем и поэтому вернул неправильные NS-записи. Наш локальный DNS-сервер, не имея лучшей альтернативы, кэшировал эти данные.

DNS-серверы BIND версии 4.9 и более поздних не поддаются на подобные провокации.

Запросы к некорректно настроеным DNS-серверам *in-addr.arpa* зачастую приводят к получению некорректных NS-записей для корневых DNS-серверов, поскольку зоны *in-addr.arpa* и *arpa* являются ближайшими предками большинства поддоменов *in-addr.arpa*, а DNS-серверы очень редко кэшируют NS-записи *in-addr.arpa* и *arpa*. (Корневые DNS-серверы редко возвращают эти записи, так как делегируют напрямую поддоменам более низких уровней.) Если DNS-сервер кэшировал некорректные NS-записи для корневых DNS-серверов, это может пагубно повлиять на разрешение имен.

Эти NS-записи могут привести к тому, что наш DNS-сервер будет посылать запрос корневому DNS-серверу, у которого изменился IP-адрес, или корневому DNS-серверу, который более не существует. Если у вас действительно тяжелый день, некорректные NS-записи корневых серверов могут указывать на существующий, не-корневой DNS-сервер, расположенный близко к вашей сети. Несмотря на то, что этот сервер не способен возвращать авторитативные ответы, ваш сервер будет оказывать ему предпочтение, исходя из высокой скорости реакции.

Используем host

Если наша лекция убедила читателей в важности соблюдения корректности делегирования, они, вероятно, не будут против узнать, как можно использовать *host*, чтобы не попасть в ряды злодеев.

Первый шаг: используем *host* для поиска NS-записей нашей зоны с помощью DNS-сервера родительской зоны и убедимся, что все в порядке. Следующая команда производит проверку для NS-записей *fx.movie.edu* с помощью одного из DNS-серверов зоны *movie.edu*:

```
% host -t ns fx.movie.edu. terminator.movie.edu.
```

Если все в порядке, NS-записи отображаются в выводе программы:

```
fx.movie.edu    NS    bladerunner.fx.movie.edu
fx.movie.edu    NS    outland.fx.movie.edu
```

Мы видим, что NS-записи, делегирующие зону *fx.movie.edu*, верны.

Теперь мы используем *host* для «SOA-теста» и запросим у каждого из DNS-серверов *fx.movie.edu* SOA-запись. Параллельно мы увидим, возвращается ли авторитативный ответ:

```
% host -C fx.movie.edu.
```

Обычно это приводит к отображению уже показанных NS-записей наряду с содержимым SOA-записи зоны *fx.movie.edu*:

```
fx.movie.edu       NS      bladerunner.fx.movie.edu
bladerunner.fx.movie.edu   hostmaster.fx.movie.edu  (1 10800 3600 608400 3600)
fx.movie.edu       NS      outland.fx.movie.edu
bladerunner.fx.movie.edu   hostmaster.fx.movie.edu  (1 10800 3600 608400 3600)
```

Если один из DNS-серверов *fx.movie.edu* – скажем, *outland* – был настроен неправильно, мы можем увидеть следующее:

```
fx.movie.edu   NS   bladerunner.fx.movie.edu
fx.movie.edu   NS   outland.fx.movie.edu
fx.movie.edu SOA record currently not present at outland.fx.movie.edu
fx.movie.edu has lame delegation to outland.fx.movie.edu
```

Смысл сообщения заключается в том, что DNS-сервер на узле *outland* работает, но не является авторитативным для зоны *fx.movie.edu*.

Если бы один из DNS-серверов *fx.movie.edu* не работал вовсе, мы увидели бы такое сообщение:

```
fx.movie.edu   NS   bladerunner.fx.movie.edu
bladerunner.fx.movie.edu   hostmaster.fx.movie.edu  (1 10800 3600 608400 3600)
fx.movie.edu   NS   outland.fx.movie.edu
fx.movie.edu SOA record not found at outland.fx.movie.edu, try again
```

В этом случае сообщение *try again* (повторите попытку) говорит о том, что программа *host* отправила серверу *outland* запрос, но не получила ответа за приемлемое время.

Разумеется, мы могли бы произвести проверку делегирования *fx.movie.edu* с помощью *nslookup*, но удобные ключи командной строки *host* позволяют решить эту задачу с особенной легкостью.

Управление делегированием

Если лаборатория специальных эффектов укрупнится, может оказаться, что необходимо увеличить число DNS-серверов. Мы уже описывали установку новых DNS-серверов в главе 8 и даже упомянули, какую информацию следует посылать администратору родительской зоны. Но мы не сказали, какие обязанности возлагаются на родителя.

Оказывается, что работа родителя в этом случае не очень сложна, особенно, если администраторы поддоменов присылают полную инфор-

мацию. Предположим, что произошло расширение лаборатории специальных эффектов в новую сеть, 192.254.20/24. В этой сети проживает стадо новых графических рабочих станций повышенной производительности. Одна из них, *alien.fx.movie.edu*, будет выступать в качестве нового DNS-сервера этой сети.

Администраторы зоны *fx.movie.edu* (которая была делегирована ребятам из лаборатории) посылают администраторам родительской зоны (то есть нам) короткое уведомление:

```
Привет!

Мы только что настроили alien.fx.movie.edu (192.254.20.3) в качестве
DNS-сервера для fx.movie.edu. Пожалуйста, обновите информацию
о делегировании. NS-записи, которые необходимо добавить, прилагаются.

Thanks,

Arty Segue
ajs@fx.movie.edu

----- cut here -----

fx.movie.edu.   86400  IN  NS  bladerunner.fx.movie.edu.
fx.movie.edu.   86400  IN  NS  outland.fx.movie.edu.
fx.movie.edu.   86400  IN  NS  alien.fx.movie.edu.

bladerunner.fx.movie.edu.  86400  IN  A  192.253.254.2
outland.fx.movie.edu.      86400  IN  A  192.253.254.3
alien.fx.movie.edu.        86400  IN  A  192.254.20.3
```

Наша задача – задача администраторов *movie.edu* довольно проста: необходимо добавить NS- и A-записи в файл *db.movie.edu*.

Что делать в случае, если мы используем программу *h2n* для создания данных DNS-сервера? Мы можем поместить информацию о делегировании в файл *spcl.movie*, который *h2n* включает с помощью директивы $INCLUDE в конец создаваемого файла *db.movie*.

Последнее действие администратора зоны *fx.movie.edu* – послать аналогичное уведомление по адресу *noc@netsol.com* (администратору зоны *in-addr.arpa*), запросив делегирование поддомена *20.254.192.in-addr.arpa* DNS-серверам *alien.fx.movie.edu*, *bladerunner.fx.movie.edu* и *outland.fx.movie.edu*.

Заглушки: еще один способ управления делегированием

Если вы работаете с DNS-сервером BIND версии 4.9 или более поздней, существует возможность избежать ручного сопровождения информации о делегировании. В DNS-серверах BIND начиная с версии 4.9 присутствует реализация экспериментального механизма *зон-заглушек*, который и позволяет DNS-серверу самостоятельно отслеживать изменения в информации, связанной с делегированием.

DNS-сервер, реализующий функциональность заглушки для зоны, выполняет дискретные запросы SOA- и NS-записей зоны, а также необходимых связующих записей. DNS-сервер использует полученные NS-записи для делегирования зоны, а SOA-записи определяют частоту выполнения подобных запросов. В этом случае, когда администраторы поддомена вносят изменения в данные DNS-серверов поддомена, они просто обновляют NS-записи, а авторитативные DNS-серверы родительской зоны запрашивают обновленные записи в пределах интервала обновления.

На DNS-серверах *movie.edu* мы добавили бы следующий оператор в файл *named.conf*:

```
zone "fx.movie.edu" {
                type stub;
                masters { 192.253.254.2; };
                file "stub.fx.movie.edu";
};
```

В случае DNS-сервера BIND 4.9 использовали бы такую инструкцию:

```
stub     fx.movie.edu     192.253.254.2     stub.fx.movie.edu
```

Обратите внимание, что следует настроить подобным образом все DNS-серверы *movie.edu*, поскольку при изменении информации о делегировании *fx.movie.edu* не изменяется порядковый номер зоны *movie.edu*.[1] Если все DNS-серверы *movie.edu* работают с зоной-заглушкой поддомена, они синхронизируются.

Как справиться с переходом к поддоменам

Мы не станем обманывать читателей – пример с поддоменом *fx.movie.edu* был не очень жизненным. Основная причина – волшебное появление узлов лаборатории специальных эффектов. В реальной жизни лаборатория началась бы с нескольких узлов, которые входили бы в зону *movie.edu*. После получения щедрого пожертвования, гранта NSF или корпоративного подарка лаборатория может немного подрасти и купить еще несколько машин. Рано или поздно в лаборатории будет достаточно узлов, чтобы гарантировать создание нового поддомена. Однако к тому моменту многие из узлов обретут известность под своими именами в домене *movie.edu*.

Мы коротко упоминали использование CNAME-записей в родительской зоне (в примере с узлом *plan9.movie.edu*), которые позволили бы пользователям безболезненно привыкнуть к переезду узла в другой домен. Но представьте себе, что целая сеть или подсеть переезжает в другой домен!

[1] Серверы имен BIND 9 обычно не передают NS-записи в родительскую зону, чтобы они не включались в данные при передаче зоны.

Стратегия, которую мы рекомендуем, связана с использованием CNAME-записей в аналогичном стиле, но в гораздо больших масштабах. Используя инструмент вроде *h2n*, можно создавать CNAME-записи для большого числа узлов сразу. Это позволяет пользователям продолжать пользоваться прежними доменными именами любых из переехавших узлов. Однако, когда они попытаются соединиться с любым из этих узлов по протоколу telnet или FTP (или еще какому-то), то получат сообщение, что подключились к узлу *fx.movie.edu*:

```
% telnet plan9
Trying...
Connected to plan9.fx.movie.edu.
Escape character is '^]'.

HP-UX plan9.fx.movie.edu A.09.05 C 9000/735 (ttyu1)

login:
```

Конечно, многие пользователи не замечают столь тонкой разницы, поэтому придется заняться рекламной деятельностью и уведомить народ о новостях.

На узлах *fx.movie.edu*, работающих со старыми версиями программы *sendmail*, необходимо настроить *sendmail* на прием почтовых сообщений для новых доменных имен. Современные версии *sendmail* производят канонизацию имен узлов из заголовков сообщений с помощью DNS-сервера, прежде чем посылать сообщения. Канонизация превратит псевдоним из *movie.edu* в каноническое имя из *fx.movie.edu*. Однако если на принимающем узле работает старая версия *sendmail*, в которой жестко закодировано доменное имя локального узла, придется изменить это имя на новое вручную. Это обычно требует внесения простых изменений в класс *w* или файловый класс *w* в файле *sendmail.cf*; более подробная информация содержится в разделе «MX-алгоритм» главы 5 «DNS и электронная почта».

Как создать все эти псевдонимы? Достаточно просто сказать *h2n*, что следует создать псевдонимы для узлов в сетях *fx.movie.edu* (192.253.254/24 и 192.254.20/24) и задать (в файле */etc/hosts*) новые доменные имена для этих узлов. К примеру, используя таблицу узлов *fx.movie.edu*, мы можем с легкостью создать псевдонимы в *movie.edu* для всех узлов *fx.movie.edu*.

Фрагмент файла */etc/hosts*:

```
192.253.254.1 movie-gw.movie.edu movie-gw
# fx: первичный
192.253.254.2 bladerunner.fx.movie.edu bladerunner br
# fx: вторичный
192.253.254.3 outland.fx.movie.edu outland
192.253.254.4 starwars.fx.movie.edu starwars
192.253.254.5 empire.fx.movie.edu empire
192.253.254.6 jedi.fx.movie.edu jedi
192.254.20.3  alien.fx.movie.edu alien
```

Ключ *-c* программы *h2n* в качестве аргумента принимает доменное имя зоны. Когда *h2n* обнаруживает узел из этой зоны в сети, для которой производится генерация данных, то создает псевдонимы в текущей зоне (которая указывается с помощью ключа *-d*). Поэтому, выполнив команду:

```
% h2n -d movie.edu -n 192.253.254 -n 192.254.20 \
-c fx.movie.edu -f options
```

(где файл *options* содержит прочие ключи командной строки для создания данных, относящихся к прочим сетям *movie.edu*), мы можем создать в *movie.edu* псевдонимы для всех узлов *fx.movie.edu*.

Удаление псевдонимов из родительской зоны

Псевдонимы в родительской зоне полезны в плане минимизации отрицательных последствий при перемещении узлов, но они, по существу, являются костылями. Как и любые костыли, они ограничивают свободу движений. Они замусоривают пространство имен родительской зоны, и это при том, что одна из причин создания поддоменов – разгрузка и уменьшение зоны. Помимо этого, псевдонимы исключают использование в родительской зоне узлов с именами, совпадающими с именами узлов поддомена.

После тактичной паузы, о наличии которой следует обязательно проинформировать пользователей, все псевдонимы должны быть удалены, за исключением псевдонимов для широко известных в сети Интернет узлов. Во время этой паузы пользователи могут привыкнуть к новым доменными именам, отредактировать сценарии, файлы *.rhosts* и т. д. Пусть вас ничто не введет в заблуждение – сохранение жизни псевдонимам в родительской зоне противоречит самой идее DNS, поскольку псевдонимы мешают администраторам родительской зоны и администраторам поддоменов производить автономное именование узлов.

Возможно, придется оставить CNAME-записи для широко известных узлов Интернет или центральных ресурсов сети нетронутыми, из-за возможных последствий утраты связи. С другой стороны, прежде чем производить перевод широко известного узла или важного ресурса в поддомен, стоит как следует подумать – быть может, этот узел следует оставить в родительской зоне.

h2n предоставляет простой способ удалить псевдонимы, столь же просто созданные с помощью ключа *-c*, даже если записи для узлов поддомена подмешаны в таблицу узлов или в ту же сеть, что узлы других зон. Ключ *-e* принимает доменное имя зоны в качестве аргумента и предписывает *h2n* исключить (*e* от слова *exclude*) все записи, содержащие указанное доменное имя для всех сетей, для которых будет происходить создание данных. К примеру, следующая команда удалит все ранее созданные записи CNAME для узлов *fx.movie.edu*, но при этом

все равно создаст адресную запись для узла *movie-gw.movie.edu* (принадлежащего сети 192.253.254/24):

```
% h2n -d movie.edu -n 192.253.254 -n 192.254.20 \
-e fx.movie.edu -f options
```

Жизнь родителя

Всю эту информацию о жизни родителя нелегко переварить за один прием, поэтому мы резюмируем основные пункты нашей дискуссии. Жизненный цикл типичного родителя выглядит примерно так:

1. Единственная зона, которая содержит все узлы.
2. Зона разбивается на поддомены, некоторые из которых входят в ту же зону, что и родитель – при необходимости. Для широко известных узлов, совершивших переезд, в родительской зоне могут быть созданы CNAME-записи.
3. После тактичной паузы все существующие CNAME-записи удаляются.
4. Администратор обновляет информацию о делегировании поддоменов вручную либо при помощи зон-заглушек и периодически проверяет работоспособность делегирования.

Теперь, когда мы рассказали все о родителях и детях, можно переходить к разговору о более серьезных возможностях DNS-серверов. Некоторые из них могут пригодиться при воспитании детей.