Содержание

4 **Незаметный трудяга**
История развития и становления операционной системы FreeBSD

Инсталляция и настройка

10 **Постоянная прописка**
Пошаговая инсталляция ОС версии 4.10: инструкции по созданию необходимых разделов и первоначальные системные настройки

18 **Сетевой колхоз**
Организация удобного общего доступа к файлам за счет сетевой файловой системы NFS. Создание групп пользователей и работа с ними

24 **Раздача имен**
Настройка службы DNS: первичные и вторичные серверы, а также гарантированное обеспечение их безопасности

30 **Автоматический администратор**
Настройка важной службы DHCP под FreeBSD осуществляется довольно просто

34 **Калифорнийский стрелочник**
Особенности настройки маршрутизации с помощью NAT-демона, который отвечает за эти функции во FreeBSD

38 **Серверная нирвана**
Настройка и конфигурирование веб-, FTP- и почтовых серверов — как в случае с отдельными машинами, так и на одном физическом сервере

Безопасность

44 **Непреодолимый барьер**
Конфигурирование сетевого экрана потребует от вас определенных усилий, но результатом проделанной работы будет последующая гарантированная безопасность сети

50 **Парад посредников**
Использование наиболее подходящих сетевых протоколов для обмена данными внутри сети поможет укрепить ее безопасность и надежность

56 **Игра в защите**
Настройка безопасности почтовых, а также таких внутренних серверов, как веб-, FTP- и DNS-, является одной из первоочередных задач администратора

Эксплуатация

64 **Воссоздание мира**
Особенности сборки системы из исходных кодов в целях обновления системы и поддержания защиты на должном уровне

68 **Старый друг**
Основные консольные команды для типовых задач, которые пригодятся администратору любого уровня

76 **Вмешательство на расстоянии**
Удаленный доступ к серверу нужен практически любому администратору. Не лишним будет и удаленный выход в Интернет и локальную сеть

Колонка редактора

Александр Иванюк
выпускающий
редактор

Американка родом из Беркли

Все мы очень любим пользоваться благами Интернета. Кто-то обращается к информационным ресурсам Глобальной сети в силу специфики своей работы, другие не могут жить без электронной почты и тематических чатов, для третьих — это способ продвинуть свой бизнес и найти новых деловых партнеров. Все они представляют, что такое Интернет, но при этом, я более чем уверен, никто из них не задумывается, за счет чего функционирует столь мощная система. Дело в том, что тысячи тысяч серверов по всему миру работают под управлением определенной операционной системы, и, оказывается, более половины всех машин в Глобальной сети своей надежностью и безотказностью в работе обязаны сетевой операционной системе FreeBSD, которая была разработана почти три десятка лет назад в американском университете Беркли. За прошедшие годы она успела снискать себе славу надежной и нетребовательной сетевой ОС, которая к тому же распространяется совершенно бесплатно. Любой человек, ступивший на путь сетевого администратора, не хочет каждую неделю (или того чаще) копаться в своих серверах из-за возникающих сбоев, поэтому многие и выбирают FreeBSD. Понятно, что ничего не бывает даром, и эта упомянутая стабильность тоже потребует от вас определенных знаний, которыми мы и хотим с вами поделиться. Забудьте про прекрасные окошки Windows, теперь ваш лучший друг — черная консоль, вернувшаяся из времен DOS. Но не стоит бояться: как только вы найдете с ней общий язык, вы сможете забыть о том, что серверы склонны давать сбои. По крайней мере, на несколько лет, ведь именно столько уже работают без остановки несколько крупных серверов в Интернете под управлением FreeBSD.

...# Незаметный трудяга/_...

История развития

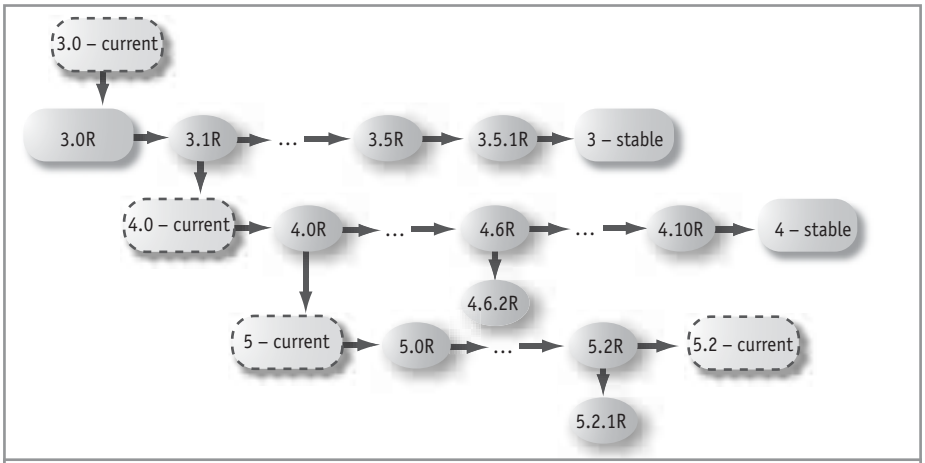
FreeBSD давно и прочно обосновалась на серверах различных сетевых служб. Эта открытая ОС регулярно возглавляет рейтинги наиболее надежных хостинг-провайдеров. Однако широкие массы пользователей до сих пор не оценили ее по достоинству.

Очень часто мы рассматриваем окружающий нас мир как двухполюсную систему, оперируя парами противоположных понятий: «да — нет», «север — юг», «хорошо — плохо» и т. д. Схожая ситуация в последнее время наблюдается и в мире операционных систем: основные игроки рынка IT-индустрии группируются вокруг двух основных «полюсов» — операционных систем Linux и Windows. Именно о них чаще всего пишут СМИ. В потоке публикаций, с энтузиазмом превозносящих один «полюс» и едко иронизирующих над «противником», читателю не всегда удается заметить других представителей рынка операционных систем. И напрасно!

Сегодня речь пойдет об одной из них — системе FreeBSD, занимающей в мире второе место по распространенности среди операционных систем с открытым кодом. В отличие от разработчиков Linux, которые временами очень агрессивно продвигают свое детище на рынок, создатели FreeBSD вовсе не пытаются убедить мир в том, что эта ОС является лучшей альтернативой другим системам. FreeBSD — своего рода компромисс между энтузиазмом и корпоративной рутиной.

История проекта FreeBSD

Своим возникновением FreeBSD во многом обязана операционной системе Unix, которая была разработана в конце 60-х — начале 70-х годов подразделением Bell Labs компании AT&T. Первые версии Unix были »



На последней версии FreeBSD — 5.2-Current — обкатываются все новейшие программные разработки

Один из основателей FreeBSD и бывший член Core Team — Джордан Хаббард

» написаны на ассемблере, но в 1973 году система была переписана с использованием разработанного Гаем Ричи языка C. Именно это дало толчок развитию Unix, упростив ее перенос на новые аппаратные платформы.

В то время компьютерный бизнес являлся монополией правительства, поэтому AT&T предложила исходные коды Unix правительственным учреждениям и университетам за сравнительно небольшую плату. Благодаря этому данная операционная система попала примерно в 80% образовательных учреждений, имевших компьютерные факультеты. Одной из первых вплотную занялась работой над Unix группа Калифорнийского университета в Беркли — Computer Systems Research Group. В 1975 году в отдел компьютерных исследований в Беркли перешел Кен Томпсон, оставивший Bell Labs. Результатом его совместной с преподователями и студентами работы стала операционная система под названием Berkley Software Distribution — BSD. Данная версия Unix оказалась настолько удачной, что в конце 70-х годов Министерство обороны США объявило, что ее подразделение ARPA будет использовать именно ту версию Unix, которая разработана в Беркли. Основными требованиями, которые предъявлялись к системе, были возможность работы в сети и высокая надежность.

В 1991 году BSD была портирована на аппаратную платформу Intel x86. Эту версию называли 386BSD. В Калифорнийском университете образовалась группа, которая стала продавать коммерческую версию BSD для платформы x86.

Возникновение FreeBSD

Собственно проект FreeBSD возник в начале 1993 года по инициативе Джордана Хаббарда, Нэйта Вильямса и Рода Граймса. В то время они выступали координаторами проекта под названием «Неофициальный »

Альтернативные ОС семейства BSD

Свободные ОС для любых нужд

Поскольку исходные коды FreeBSD открыты, время от времени у энтузиастов появлялось желание сделать собственную ОС. Зачастую выдвигаемые разработчиками концепции привлекали большое количество людей, и эти проекты выливались в зрелые операционные системы. На сегодняшний день заслуживают внимания две родственные FreeBSD ОС: NetBSD и OpenBSD. Расскажем вкратце об их истории и особенностях.

NetBSD

Начало разработки этой ОС датировано 20 апреля 1993 года, когда был выпущен релиз под номером 0.8. NetBSD, как и FreeBSD, берет начало от дистрибутива 386BSD. Однако если FreeBSD направила свои усилия на платформу i386, то NetBSD сконцентрировалась на кроссплатформенности. С тех пор эта ОС была портирована на 40 различных платформ, среди которых есть и игровые консоли, и наладонники, и компьютеры фирмы Apple, и множество других, порой весьма экзотических устройств. Еще одной задачей, поставленной разработчиками, была переносимость программ. Система бинарной эмуляции позволяет запускать программы, скомпилированные для различных Unix-подобных ОС: FreeBSD, SunOS, HP-UX и других. NetBSD характеризуется также стабильностью и быстродействием. И хотя это в первую очередь исследовательская ОС, она используется и в качестве DNS и других серверов на многих интернет-хостингах. Последний релиз этой ОС — NetBSD 1.6.2 — выпущен 1 марта 2004 года.

OpenBSD

Родословная этой ОС восходит к NetBSD 1.1. Первая версия OpenBSD появилась в 1995 году по инициативе Тео де Раадта, разработчика NetBSD, ответственного за портирование на платформу SPARC. Резиденция OpenBSD находится в Канаде, что позволило внедрить в ОС алгоритмы шифрования и технологии безопасности, запрещенные к экспорту в США: RSA, Blowfish и другие. Девиз OpenBSD — «Безопасность по умолчанию». Например, сразу после установки все сетевые порты закрыты. Способствует защищенности системы и постоянный аудит кода. Как и NetBSD, OpenBSD хорошо переносится на различные аппаратные платформы. Текущая версия этой ОС — OpenBSD 3.5.

Оконная среда KDE предоставляет пользователю удобный интерфейс и множество утилит, но требует достаточно мощного компьютера

» комплект исправлений к 386BSD» (patchkit), представлявшего собой серию исправлений и дополнений к 386BSD. По утверждению Джордана, руководитель проекта 386BSD Билл Джолиц не проявлял особого интереса к судьбе своего детища и в конце концов прекратил его поддержку. В результате Хаббард и его коллеги решили начать собственный проект: Дэвид Гринмэн дал ему имя FreeBSD.

Первым дистрибутивом, который распространялся как на CD-ROM, так и через Интернет, был FreeBSD 1.0, выпущенный в декабре 1993 года. Он был выполнен на основе 4.3BSD-Lite (Net/2) Калифорнийского университета в Беркли.

К сожалению, в 1994 году возникли проблемы с лицензированием. Компания Novell выступила инициатором судебного процесса, заявив, что ей принадлежат права на часть кода Net/2. В ответ Беркли выпустил версию 4.4BSD-Lite, которая была объявлена полностью «свободной»; всем пользователям Net/2 рекомендовалось перейти именно на эту версию. Это же касалось и FreeBSD — проекту было дано время до конца июля 1994 года для прекращения выпуска версии дистрибутива, базирующейся на основе Net/2.

FreeBSD пришлось приступить к «буквально полному изобретению себя из абсолютно новой и довольно неполной системы 4.4BSD-Lite» (цитата из Handbook — официального руководства пользователя FreeBSD). Чтобы выпустить новую версию, понадобилось почти полгода. В августе 1996 года вышла в свет версия 2.1.5, которая завоевала большую популярность среди интернет-провайдеров. Тогда же произошло и ветвление в дереве разработки: появился первый официально стабильный релиз — 2.1-Stable.

Версия 3.0, явившаяся логическим продолжением развития ветви 2.2, вышла в свет в октябре 1998 года. При этом очередное ветвление в дереве разработки произошло через три месяца после этого — появились ветки 4.0-Current и 3.x-Stable. В ветке 3.x-Stable было шесть релизов (точнее, релизов было пять, с 3.1 по 3.5, а шестой представлял собой некоторое обновление предыдущего и содержал исправления в области безопасности Kerberos).

Последнее ветвление в дереве проекта было выполнено 13 марта 2000 года, благодаря чему появилась ветка 4.x-Stable, являющаяся официально стабильной веткой (последний релиз из нее — 4.10-Release — датируется маем 2004 года). А вот первый релиз 5.x был анонсирован только через три года — 19 января 2003 года. Причину такой задержки можно объяснить тем, что начиная с этого релиза был взят курс на поддержку многопроцессорности, потоков в приложениях, и аппаратных платформ UltraSPARC и IA64.

В настоящий момент принято считать релизы 4.x «промышленными» (production), а 5.x — «новыми технологическими» (new technology). Со временем 5.x-Current займет место 4.x-Stable, а ветка 6.x-Current станет полигоном для обкатки нового программного обеспечения.

FreeBSD: цели, лицензии и принципы развития

«Целью проекта FreeBSD является предоставление программного обеспечения, которое может быть использовано для любых целей и без дополнительных ограничений» (цитата из руководства пользователя).

Участники проекта предпочитают использовать программное обеспечение, предоставляемое под лицензиями BSD. Это позволяет избегать дополнительных сложностей, которые могут появиться при коммерческом использовании GPL-продуктов.

Каждый проект внутри FreeBSD поддерживает публично доступное дерево исходных текстов программ. При помощи CVS (Concurrent Versions System) можно получить доступ к коду проекта, его документации и вспомогательным файлам. Это позволяет пользователям в любой момент получить копию дерева любого из проектов или системы в целом. Всех, кто в той или иной мере участвует в разработке FreeBSD, можно разделить на три категории:

► Контрибьюторы (contributors) — те, кто пишет код или документацию, но не имеет права вносить изменения непосредственно в дерево разработки. Они только предоставляют изменения и дополнения к коду, а решение об их внесении принимают коммиттеры.

► Коммиттеры (committers) — участники группы разработки, имеющие право записи в дерево CVS. Как правило, коммиттер сам решает, необходимо ли ему подтверждение от других участников проекта для внесения изменений в код. Если изменения в коде имеют далеко идущие последствия, проводится предварительное обсуждение. Возможны случаи, когда член Core Team, выполняющий функции архитектора проекта, отклоняет внесенные изменения.

► Core Team — группа людей, управляющих деятельностью разработчиков FreeBSD. Абсолютно четко их права не определены, но, как правило (хотя и не обязательно), член Core Team является также коммиттером. Правила, которыми руководствуются члены Core Team, могут меняться от проекта к проекту, но в общем и целом именно участники »

» очередь не на личных предпочтениях, а на результатах тщательного анализа поставленной задачи. Детальное сравнение характеристик FreeBSD и Windows может занять несколько страниц, поэтому кратко остановимся на основных различиях между этими системами:

► Графический интерфейс — неотъемлемая часть Windows, но FreeBSD прекрасно обходится и без него. Редактировать конфигурационные файлы при помощи окон не всегда удобно, и к тому же часть настроек при таком виде редактирования бывает недоступной. Поэтому для внесения необходимых изменений в конфигурационный файл легче воспользоваться простым текстовым редактором (это также упрощает удаленное администрирование на медленных линиях связи). Кроме этого, при отсутствии графической оболочки требования к аппаратным ресурсам сервера снижаются.

► Форматы хранения настроек системы в Windows и FreeBSD различны. Windows использует для хранения данных реестра двоичные файлы, тогда как в FreeBSD конфигурационные настройки различных служб хранятся в отдельных файлах и, как правило, в текстовом виде. Преимущества последнего метода очевидны: если файл реестра Windows разрушен, система приходит в нерабочее состояние и приходится использовать внешние инструменты для восстановления резервных копий, а в случае с FreeBSD неисправной окажется только та служба, файл которой оказался испорчен.

► Ядро операционной системы Windows является своего рода «монолитом», который невозможно модифицировать без полной замены версии операционной системы. FreeBSD позволяет компилировать новое ядро, максимально соответствующее той аппаратной платформе, на которой работает система. Помимо этого, необходимость в замене ядра может возникать в том случае, если в его коде обнаружена «дыра» в безопасности.

► Уровень подготовки пользователей Windows и FreeBSD, необходимый для решения задач системного администрирования начального уровня (например, развертывание одноранговой локальной сети), также неодинаков. Здесь Windows имеет неоспоримое преимущество перед FreeBSD, которое, однако, исчезает при решении более сложных задач.

Диски с FreeBSD выпускают множество издателей, но, в отличие от Linux, дистрибутив создается централизованно

На серверах под управлением FreeBSD работают сайты всемирно известных компаний

► Лицензия FreeBSD — еще одна причина обратить внимание именно на эту операционную систему. В отличие от Windows, при работе с которой приходится приобретать лицензии на каждое сетевое подключение, FreeBSD не налагает на своих пользователей таких ограничений.

Сравнение FreeBSD и Linux

Сторонники каждой из операционных систем могут спорить до хрипоты о преимуществах именно своей системы и недостатках другой. Для некоторых эти споры стали своеобразным «спортом». Опустим технические детали и остановимся на различиях, касающихся организационных вопросов:

► Лицензия, используемая в FreeBSD, не содержит так называемых «передающихся свобод». Она не обязывает вас открывать исходные тексты программ, если вы этого не хотите. В отличие от BSD-лицензии, GPL-лицензия требует предоставлять по первому требованию исходные тексты программ. При этом основная цель GPL-лицензии — стоять на страже интересов разработчиков, не позволяя тем, кто не вложил ни капли своего труда в создание кода, зарабатывать на нем. В этом отношении BSD-лицензия проявляет больше либерализма.

► Количество дистрибьюторов операционной системы Linux измеряется десятками. В случае с FreeBSD дистрибьютор один, и только он решает, в каком направлении необходимо развиваться. В Linux ситуация диаметрально противоположная — каждый производитель видит дальнейшее развитие системы по-своему.

► Контроль кода, поступающего в FreeBSD, осуществляет основной состав разработчиков, в то время как в Linux каждый дистрибьютор контролирует качество кода самостоятельно. Нельзя сказать, что процессы контроля полностью изолированы, но каждая из компаний, выпускающих Linux, сама решает вопрос о включении тех или иных «заплаток», которые используют их коллеги или конкуренты. Попытка создать унифицированный Linux пока не увенчалась успехом — проект United Linux выпустил первый релиз, но дальше этого дело не пошло. В принципе, вопрос о том, чья модель разработки лучше, — весьма сложный и спорный. Если рассматривать его с позиций надежности, выигрывает подход FreeBSD Team. Но с позиций быстроты разработки, модель, используемая Linux-сообществом, выглядит более предпочтительной.

Заключение

Вот уже 11 лет, как FreeBSD является заметным игроком на рынке операционных систем и уходить с него пока не собирается. Стабильность системы доказана годами эксплуатации: под управлением FreeBSD работают сайты таких компаний как Yahoo!, Netcraft, Sony Japan, Weathernews (именно в Weathernews Inc. располагаются компьютеры-рекордсмены по непрерывной работе) и многих других.

■ ■ ■ Александр Куприн

Бесплатный офисный пакет, способный работать с документами Microsoft Office, существует и для FreeBSD

С помощью Samba и административной утилиты SWAT легко получить доступ к ресурсам сети Microsoft

» этой группы определяют направление развития операционной системы FreeBSD.

Проект FreeBSD предоставляет пользователям три различных варианта системы. Версиям присваиваются номера (например, 3.5.1 или 4.10), к которым добавляется суффикс, указывающий на цели версии:

► Current — версия для разработчиков (например, FreeBSD 5.0-Current) — все новые разработки проходят тестирование именно на этой ветке;

► Release — версия для конечных пользователей (как правило, она появляется раз в три-шесть месяцев);

► Stable — версия FreeBSD, являющаяся логическим продолжением версии Release. По мере того, как в Release обнаруживаются ошибки и в дерево CVS вносятся изменения, дистрибутив переходит на стадию Stable. В настоящий момент официально стабильной веткой является 4.x.

Почему FreeBSD?

FreeBSD в роли сервера

Как и любая сетевая операционная система, FreeBSD предлагает набор программ, которые позволяют превратить компьютер в узел, предоставляющий интернет-услуги: электронная почта, веб- или FTP-сервер, сервер доменных имен, средства для маршрутизации и сетевой трансляции адресов, прокси-сервер, сетевой экран и т. п. Кроме того, свободно распространяемый пакет Samba позволяет организовать на базе FreeBSD файловый сервер и сервер печати в сети Microsoft либо создать PDC (Primary Domain Controller) для сети Windows. Возможности FreeBSD в роли сервера сети Microsoft ограничиваются только возможностями Samba. В настоящий момент идет активная разработка Samba 3. Среди наиболее значимых нововведений этого пакета следует отметить поддержку Unicode в именах файлов (это существенно упрощает хранение файлов с нелатинскими именами), идентификацию пользователей при помощи LDAP/Kerberos 5 и поддержку службы каталогов Active Directory (хотя текущая версия Samba может выступать лишь в роли члена AD).

Если добавить к этому возможность запуска на FreeBSD как коммерческих, так и свободно распространяемых версий SQL-серверов, а также работы в составе кластера, становится понятно, насколько широкий круг задач способна решать FreeBSD.

Однако большое количество доступного программного обеспечения не всегда делает операционную систему привлекательной — особенно если она выступает в роли сервера. Как упоминалось ранее, одним из основных требований к BSD была устойчивость в работе. Согласно данным Netcraft.com по состоянию на 21 июня 2004 года, из 50 наиболее продолжительно работающих в Интернете компьютеров шесть работают именно под управлением FreeBSD, причем два из них занимают первые места и работают без перезагрузки более 4,5 лет.

FreeBSD на рабочем столе

FreeBSD вполне подходит на роль рабочей станции, если речь идет о “среднестатистической” системе, которая используется для работы в Интернете и для обработки текстовой информации. Здесь, как и в большинстве Unix-подобных операционных систем, графический интерфейс представлен средой X Window System. Ядро X-сервера примитивно и не обладает большим количеством функциональных возможностей, поэтому существует специальный класс программ — диспетчеры окон (или оконные менеджеры), которые упрощают работу пользователей. Упрощенные оконные менеджеры способны работать на маломощных компьютерах, но есть и оконные среды, по удобству использования сравнимые с Windows. Это проекты KDE и GNOME.

В качестве одного из офисных пакетов в FreeBSD может использоваться OpenOffice, способный взять на себя большую часть функций Microsoft Office. Конечно, если речь идет о запуске программ, аналогов которым в FreeBSD нет (например, это активно использующиеся сегодня продукты компании 1С), приходится отказываться от намерения использовать эту операционную систему на настольном компьютере.

FreeBSD vs. Windows

Необходимо понимать, что выбор операционной системы для решения той или иной задачи должен основываться в первую »

С серверами HP на базе **процессоров Intel® Xeon™** ваш бизнес способен на большее!

Серверы HP ProLiant ML150 и DL140 разработаны с учетом требований малого и среднего бизнеса. Расширяемые модели HP ProLiant ML150 и DL140, оснащенные **процессорами Intel® Xeon™**, будут расти вместе с вашим бизнесом. Их высокая надежность и производительность — это ваша уверенность в будущих успехах. HP ProLiant ML150 — сервер начального уровня. Мощность двух **процессоров Intel® Xeon™** и возможность установки опций третьих фирм. HP ProLiant DL140 — тонкий стоечный сервер высотой 1U. Обладает гибкой конфигурацией, поддерживает до двух **процессоров Intel® Xeon™**. Выберите серверы HP ProLiant на базе **процессоров Intel® Xeon™** — позвольте передовым технологиям работать на вас!

HP PROLIANT ML150

$ 1599 Рекомендуемая розничная цена на сервер ML150 с **Процессором Intel® Xeon™** 2,80 ГГц, 256 МБ памяти, диском 36 ГБ Ultra 320 SCSI с возможностью горячей замены.

- До 2 **Процессоров Intel® Xeon™ 2,40/2,80 ГГц**
- Два варианта поставки — для дисков с/без горячей замены
- Допускается установка опций третьих фирм
- Память до 12 ГБ SDRAM
- Интегрированный двухканальный контроллер SCSI Ultra 320
- Внутренние накопители до 730 ГБ ATA

HP PROLIANT DL140

$ 1475 Рекомендуемая розничная цена на сервер DL140 с **Процессором Intel® Xeon™** 2,40 ГГц, 512 МБ памяти, диском 80 ГБ Ultra ATA/100.

- До 2 **Процессоров Intel® Xeon™ 2,80/3,20 ГГц**
- Высота сервера 1U
- Расширенная защита памяти Advanced ECC
- Память до 4 ГБ SDRAM
- Внутренние накопители до 320 ГБ ATA

hp invent

До 31 декабря 2004 года, покупая серверы HP ProLiant DL140, ML150, оснащенные **процессорами Intel® Xeon™**, вы можете получить скидку 20% на Microsoft® Windows® Small Business Server 2003.

ТЕЛ.	**8 800 200 3 500 звонок бесплатный**
САЙТ	**www.hp.ru/promo/proliant&sbs**

intel inside XEON

Наши партнеры: **Волгоград:** Вкс Про (8442) 36-92-32, Телесто (8442) 34-58-90, VOGS'S (8442) 90-00-70; **Екатеринбург:** АСМ-электроника (343) 378-31-23, АСП (343) 370-67-05, Деком КС (343) 217-91-97, Диджитек (343) 377-74-07, Клосс Сервисез Корпорейшн (343) 216-17-01, Компьютер без Проблем (343) 355-30-04, Корус АКС (343) 376-23-00, Крона КС (343) 242-35-61, Новаком (343) 263-74-66, Парад-Компьютерные технологии (343) 257-52-08, Промавтоматизация 2002 (343) 257-20-88, Трилайн (343) 378-70-70, Эр-Стайл-Урал (343) 261-60-86, Ю-Ти-Ай (343) 365-81-09; **Иркутск:** Атон (3952) 51-17-45, Сайбирит (3952) 25-81-28; **Казань:** Абак-Центр (8432) 72-97-21, Форт Диалог (8432) 95-23-69, ICL-КПО ВС (8432) 73-24-43; **Кемерово:** Кузбасский Компьютерный центр (3842) 58-10-23; **Киров:** Вит (8332) 64-04-10, Находка-Киров (8332) 57-71-15; **Красноярск:** Синтез-Н (3912) 55-55-19; **Краснодар:** Бизнес Компьютер Центр – Юг (8612) 64-04-50, Интеркрайт (8612) 60-56-75; **Нижний Новгород:** Вист-НН (8312) 17-45-07, ЛИК-Н (8312) 34-27-70; **Новороссийск:** Эльдорадо Новороссийск (8617) 63-01-26; **Новокузнецк:** Бит-Тех (3843) 74-07-70; **Новосибирск:** Интерлинк (3832) 34-44-44, Кардинал (3832) 10-62-02, Нонолет (3832) 35-65-35, НЭТА (3832) 10-65-04, Сибвэй (3832) 17-38-17; Сибкон (3832) 23-23-92; **Омск:** Сибирский компьютер (3812) 30-66-93, Сибирский медведь (3812) 30-12-65; **Пермь:** НПО Индукция (3422) 69-35-43, НЭТА (3422) 12-01-91; **Ростов-на-Дону:** Информатика (8632) 99-01-01, Технополис (8632) 61-86-17, Форте (8632) 67-68-10, CREDITCARD (8632) 64-47-33, R-Style (8632) 52-48-13; **Сургут:** Технотрейд (3462) 24-19-99; **Самара:** Железная логика (8462) 79-02-25, Киберкуб (8462) 42-50-23, КоссПлюс (8462) 51-96-00, Крафт-С (8462) 41-24-12, МДСконтрол (8462) 24-01-12, НПП Бинар (8462) 70-50-45, Прагма (8462) 70-17-01; **Саратов:** Современные технологии (8452) 45-00-45, Техносерв (8452) 28-36-09; **Таганрог:** Стинс-Таганрог (8634) 31-11-00; **Томск:** Интант (3822) 56-16-01; **Тольятти:** Артэк (8482) 70-60-70, СофтЭкс (8482) 42-07-59, Спайс (8482) 22-86-15, ТопсБИ (8482) 42-09-09; **Тюмень:** Арсенал+ (3452) 46-47-74, CAT LTD (3452) 41-16-63; **Ульяновск:** Апрель (8422) 31-83-72, Симбирск М+ (8422) 42-00-03; **Уфа:** Банкос (3472) 79-81-00, Бизнес-софт (3472) 77-14-70, Грит (3472) 51-69-99: **Челябинск:** Алиас (3512) 37-88-96, Астра СТ (3512) 63-00-75, НТЦ Логис (3512) 41-01-81, Электронные микросистемы (3512) 60-56-70; **Якутск:** Эльф-95 (4112) 45-73-33.

Постоянная прописка/_

Пошаговые рекомендации

Установка — один из важнейших этапов в жизненном цикле любой операционной системы. От того, насколько правильно она спланирована, зависит очень многое. В первую очередь это относится к разметке жесткого диска — если она выполнена грамотно, в дальнейшем у вас не должно возникнуть особых проблем.

Перед тем как приступать к инсталляции, давайте определимся: где, собственно, можно достать дистрибутив свободно распространяемой ОС FreeBSD. Способов существует как минимум два.

1. Интернет. Для этого вам понадобится канал достаточно большой пропускной способности. Чтобы не перегружать сеть, старайтесь использовать зеркала сайта проекта FreeBSD, которые расположены ближе всего к вам. Список зеркал вы найдете в приложении A «Руководства FreeBSD» (далее Handbook): http://www.freebsd.org/doc/ru_RU.KOI8-R/books/handbook/index.html Если вы скачаете ISO-образ, не забудьте проверить его контрольную сумму (MD5). Подобные программы существуют как для Unix, так и для DOS/Windows; они позволяют убедиться в том, что данные были приняты без искажений (случайных или преднамеренных). При этом информацию о контрольных суммах разумнее брать из первоисточника: ftp://ftp.freebsd.org/pub/FreeBSD/releases/i386/ISO-IMAGES/4.10/CHECKSUMS.MD5.

2. Онлайн-поставщики. Если для вас это окажется удобнее или выгоднее, вы можете заказать FreeBSD в интернет-магазине, специализирующемся на продаже свободного ПО. Наиболее известны среди таких магазинов на территории СНГ российский LinuxCenter.RU и украинский Lafox.NET. Если их услуги вас чем-то не устраивают, обратитесь за советом к Google или любой другой поисковой системе, набрав соответствующий запрос.

Предварительная подготовка

Сбор информации

Прежде чем приступить к установке, необходимо собрать информацию об аппарат- »

» ной конфигурации компьютера. Как минимум, вам понадобятся сведения:

► О сетевой карте (производитель, чипсет). Если это ISA-карта, необходимо выяснить номер порта ввода-вывода и номер прерывания, которые она использует.

► О модеме: если он встроенный, нужно уточнить, какое прерывание он использует; если же модем подключается к последовательному порту, необходимо знать номер порта (COM1 или COM2).

► О параметрах сетевого подключения (IP-адрес и имя вашей машины, маска подсети, IP-адрес DNS-сервера и шлюза по умолчанию, имя домена).

Поскольку в данной статье мы будем рассматривать установку FreeBSD в качестве сервера, необходимости в таких жизненно важных данных для настройки графической подсистемы, как информация о видеокарте и модели монитора, у нас не возникнет. Возможно, в дальнейшем понадобится указать параметры подключения мыши (PS/2, Serial, Bus), а в случае с подключением к последовательному порту — номер порта. Однако для большинства моделей распознавание происходит автоматически.

Если вы не уверены в том, что FreeBSD поддерживает оборудование, установленное на вашем ПК, обращайтесь к списку совместимого оборудования для релиза 4.10 (ftp://ftp.freebsd.org/pub/FreeBSD/releases/i386/4.10-RELEASE/HARDWARE.HTM).

Подготовка дискового пространства

Если вы производите установку FreeBSD на «чистый» диск, этот раздел можно пропустить. В противном случае вам необходимо решить проблему выделения свободного дискового пространства. Запомните главное условие — для установки FreeBSD необходим свободный первичный раздел.

У персональных компьютеров первый сектор жесткого диска называется главной загрузочной записью (MBR — Master Boot Record). Он состоит из двух частей — кода загрузчика ОС и таблицы разделов (partition table). При этом таблица разделов позволяет разместить всего четыре записи с информацией о разметке. Чтобы обойти подобное ограничение, используются так называемые расширенные (extended) разделы. Это своего рода «матрешки» — их первый сектор содержит свою таблицу разделов. Ограничение одной таблицы четырьмя записями сохраняется, поэтому в том случае, если используется более четырех разделов, программа разметки дисков формирует цепочку из расширенных разделов, вложенных друг в друга. Разделы, информация о геометрии которых располагается непосредственно в таблице разделов MBR, называют первичными (primary). Их максимальное количество равно количеству записей в таблице (четыре).

Если на вашем компьютере установлена ОС Windows, то прежде чем воспользоваться одной из программ переразметки дисков, нужно выполнить следующие действия:

► Проверить целостность файловой системы — для этого воспользуйтесь программой Scandisk («Мой компьютер» → «Свойства диска» → «Сервис» → «Проверка диска»). Эта операция необходима для устранения цепочки потерянных кластеров и, при необходимости, проверки поверхности диска на наличие сбойных блоков.

► Создать резервные копии важных данных и скопировать их на внешние носители. Существует вероятность того, что компьютер «зависнет» или окажется обесточен в тот момент, когда будет производиться изменение размера диска. Поэтому нет смысла рисковать данными, потеря которых для вас нежелательна.

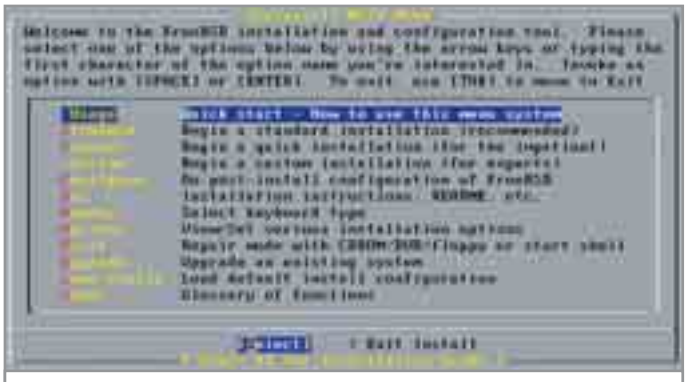
► Дефрагментировать пространство логического диска при помощи программы Defrag («Мой компьютер» → «Свойства диска» → «Сервис» → «Дефрагментация диска»). Эта операция позволит «подвинуть» файлы к началу раздела, тем самым предоставив больше свободного места для раздела под FreeBSD.

Самым простым будет вариант использования программ с графической оболочкой. В среде Windows среди них наиболее известны Partition Magic и Acronis Partition Expert.

FIPS

В том случае, если вы хотите изменить размер только FAT- и первичного раздела и не имеете возможности использовать PM или APE, можно прибегнуть к услугам программы FIPS, которая прекрасно справляется с функциями изменения геометрии FAT-разделов без потери данных. FIPS вы найдете в подкаталоге tools загрузочного диска FreeBSD: tools/fips.exe. Учитывая, что FIPS должна иметь полный доступ к диску, вам придется перегрузиться в однопользовательскую среду MS-DOS. По сравнению с аналогичными программами, работающими в консоли, FIPS частично автоматизирует процесс переразметки, что облегчает работу с ней.

Итак, вот что необходимо сделать при работе с FIPS: перейдите в подкаталог, где расположена программа, и запустите ее. Вы увидите приглашение. Нажмите любую клавишу. »



Внешний вид sysinstall отличается простотой и строгостью

Набор опций, который можно настроить до начала установки

» Если в вашем компьютере установлено более одного жесткого диска, FIPS предложит выбрать жесткий диск (цифры 1, 2, 3 и т. д.). Программа покажет вам содержимое таблицы разделов. Если разделов несколько, выберите порядковый номер того из них, размер которого вы хотите изменить (от 1 до 4.).

После этого FIPS покажет информацию о разделе и предложит сохранить данные корневого и загрузочного разделов на дискете. Рекомендуется сделать резервную копию.

Затем программа вычислит минимальный размер, который будет занимать "сжимаемый" раздел (насколько сильно можно его сжать, зависит от того, выполнялась ли перед этим дефрагментация). Теперь, используя клавиши управления курсором Left и Right, выберите размер для нового раздела. Перед вами будет пара чисел: слева — новый размер старого раздела, справа — размер создаваемого раздела. С помощью клавиши Enter подтвердите выбор.

Программа отобразит информацию в таблице разделов с учетом изменений. Если вы согласны, нажмите C, а затем Enter; если же нет — R, и вы вернетесь к пункту 3. В результате будет создан первичный FAT-раздел, который в дальнейшем при установке можно удалить и разместить на его месте FreeBSD.

Важно не записывать ничего на жесткий диск до перезагрузки системы: не забывайте, что DOS «не знает», что таблица разделов изменилась, поэтому сразу по окончании работы с FIPS перезагрузитесь. После этого запустите FIPS с опцией -t, что позволит убедиться в том, что разметка прошла без ошибок. Если ошибки обнаружены, воспользуйтесь утилитой restorrb.exe, которая находится в том же подкаталоге, что и FIPS.

Возможно, вы обратили внимание, что пока не было сказано ни слова о том, сколько дискового пространства требуется для FreeBSD? Минимально необходимое место на диске составляет около 100 Мбайт. Однако если вы планируете использовать порты, только для дерева портов вам понадобится около 300 Мбайт на диске. Чтобы не ломать голову над тем, как разместить все файловое хозяйство, исходите из расчета 1-1,5 Гбайт (и более) на диске плюс место для работы тех служб, которые вы собираетесь использовать: почтовый, веб- и файловый серверы, прокси-сервер и т. п. Конечно, можно не устанавливать дерево портов и сборку пакетов производить самостоятельно, но лучше доверить эту миссию специалистам, уже выполнившим большую часть работы за вас. К этому можно добавить, что минимальные требования к процессору — машина уровня 386, объем оперативной памяти от 16 Мбайт (при меньшем объеме не будет работать программа установки sysinstall). Как видите, требования очень демократичные.

Подготовка загрузочных дисков

Первый компакт-диск FreeBSD является загрузочным. Если ваш компьютер не оснащен CD-приводом, или BIOS не поддерживает загрузку с компакт-диска, создайте загрузочные дискеты. Образы загрузочных дискет располагаются в подкаталоге floppies. Для записи образов на дискеты в среде DOS/Windows используйте утилиту fdimage.exe из подкаталога Tools\. Вам необходимо создать две дискеты, используя образы kern.flp и msfroot.flp (x — имя CD-привода в среде DOS/Windows):

```
cd x:\tools
fdimage.exe \floppies\kern.flp a:
fdimage.exe \floppies\msfroot.flp a:
```

Для записи образов на дискету в Unix используется утилита dd. Формат вызова утилиты для FreeBSD и для Linux отличаются только именованием устройства флоппи-привода.

```
dd if=kern.flp of=/dev/rfd0
```

и, для сравнения,

```
dd if=kern.flp of=/dev/fd0
```

Обратите внимание, что fdimage.exe при записи использует прямой доступ к флоппи-диску, поэтому необходимо либо перегрузиться в режим эмуляции MS-DOS, либо запускать программу, используя права администратора. Выбор способа загрузки за вами. Перед началом следующего этапа место на диске для установки FreeBSD должно быть уже освобождено.

Начальная загрузка

На начальном этапе загрузки на экран выводится масса служебных сообщений. Детальный разбор каждого из них в объеме одной статьи невозможен. Основным показателем того, что все в порядке, является так называемая «волшебная палочка». Пока она вращается, можно считать, что все хорошо — процесс загрузки идет. Как только она остановилась, это верный признак того, что система «зависла».

В случае успешной загрузки система выдаст следующее сообщение: »

Disklabel Editor занимается разметкой дискового пространства непосредственно внутри слайса

Выбор дистрибутивных наборов определяет список пакетов, которые будут установлены на жесткий диск

```
Hit [Enter] to boot immediately,
      or any other key for command prompt.
Booting [kernel] in 9 seconds...
```

Для продолжения нажмите клавишу Enter. Если же происходит загрузка с дискет, вам будет предложено сперва сменить дискету с kern.flp на mfsroot.flp и только затем перейти к загрузке ядра:

```
Please insert MFS root floppy
      and press enter:
```

Следующий этап — настройка параметров загружаемого ядра «Kernel configuration menu». Большинству пользователей менять ничего не нужно (пункт «Skip kernel configuration...») Однако существует вероятность того, что в вашем оборудовании имеются устройства, конфликтующие между собой. В таком случае на этапе загрузки необходимо отключить драйвер одного из них, или всех, если в них нет необходимости при установке системы. Для этого воспользуйтесь вторым пунктом меню «Start kernel configuration in full-screen visual mode». За более детальной информацией обращайтесь к Handbook, раздел 2.3.2.

Общие сведения о sysinstall

Когда ядро завершит процесс загрузки, управление перейдет к программе sysinstall. Если до этого вам приходилось сталкиваться только с установкой операционных систем семейства Windows, возможно, вы будете разочарованы, не встретив привычного графического интерфейса. Поверьте, что его удобство — лишь мнимое. Программу sysinstall можно запускать и после окончания установки: она может быть использована для добавления или удаления программных компонентов, а также для обновления ОС (ищите программу в подкаталоге /stand).

Несколько слов нужно сказать и об управлении. Клавиши Up и Down используются для навигации по списку элементов меню. Клавиши Left и Right для выбора кнопок управления (Select, OK, Exit и т. п.) Клавиша Space используется для установки/сброса флажков или выбора элемента меню, рядом с которым стоят квадратные скобки. Для выбора активной кнопки управления используется клавиша Enter. В некоторых меню она может быть использована вместо клавиши пробела. Начальные символы большинства пунктов меню выделены другим цветом; чтобы активизировать такой пункт, достаточно нажать соответствующую клавишу в сочетании с Alt (это правило распространяется только на ту часть списка, которая помещается на экране). Клавиша Tab работает так же, как курсорные клавиши Left/Right, и используется для перехода между полями при вводе данных (данные о пользователе и т. п.). К сожалению sysinstall имеет только англоязычный интерфейс (соответственно, и доступная в момент установки документация представлена на английском языке).

Рассмотрим кратко основные пункты меню, доступные на начальном этапе установки:

- Usage — описание использования системы меню (навигация, горячие клавиши и т. п.).
- Standard — начать установку в стандартной конфигурации (рекомендуется).
- Express — начать быструю установку (для нетерпеливых; рекомендуется быть внимательным с этим пунктом).
- Custom — режим установки для экспертов.
- Configure — выполнить пост-инсталляционную настройку FreeBSD.
- Doc — инструкции по установке и т. п.
- Keymap — выбор раскладки клавиатуры.
- Options — просмотр/выбор различных опций перед началом инсталляции.
- Fixit — режим ремонта, запуск оболочки на четвертой консоли (shell) .
- Upgrade — обновление существующей операционной системы.
- Load Config — загрузка конфигурации установки по умолчанию.
- Index — глоссарий функций.

При установке новой системы вам понадобятся пункты Standard, Configure, Keymap и Options. Использование остальных пунктов опционально.

Установка: стандартный режим

Прежде чем перейти непосредственно к установке, необходимо выбрать раскладку клавиатуры (пункт меню «Keymap»). Нас интересует «Russia KOI8-R». В качестве переключателя между языками используется клавиша CapsLock.

Следующим пунктом остановки будет раздел «Options», позволяющий выполнить предварительную настройку системы.

Большинство пользователей устраивают настройки параметров, заданные по умолчанию. Если же вы хотите изменить один из пунктов, используйте для этого клавишу Space. Краткое описание того или иного пункта вы будете видеть внизу экрана. Понадобится ли вам использование сетевой файловой системы (NFS) или нет, решать вам. В том случае, если для вашей машины настроена динамическая IP-адресация, не

Окно выбора интерфейса для настройки доступных сетевых устройств предлагает вам шесть различных вариантов

Настройка консоли необходима для последующего корректного отображения кириллицы на экране

» забудьте включить пункт DHCP. Здесь же можно указать источник установки — CD/DVD, DOS-раздел, FTP-, HTTP-, NFS-сервер, файловая система FreeBSD или накопитель на магнитной ленте. Пункт «Use defaults» позволит произвести сброс тех изменений, которые вы внесли, на значения по умолчанию. Выход — с помощью клавиши Q.

Разметка диска

Рассмотрим пример установки в стандартном режиме. Выберите пункт «Standard». Прежде всего вам необходимо создать раздел для размещения на нем FreeBSD. К этому моменту вы должны были решить данную проблему при помощи одной из программ переразметки дисков. Программа fdisk немного похожа на свой аналог в DOS/Windows или Linux — cfdisk.

Прежде чем перейти к созданию разделов, нужно сказать несколько слов об организации дискового пространства во FreeBSD. На вершине этой иерархии располагается слайс (slice). На каждом жестком диске может быть создано до четырех слайсов (по количеству записей в таблице разделов). Слайс можно сравнить с расширенным разделом, но сходство между ними ограничивается тем, что внутри обеих структур создаются подразделы. Разделы, располагаемые внутри слайса, обозначаются буквой от A до H и могут содержать только одну файловую систему. Для разделов с A по D существуют определенные соглашения (они не обязательны, но желательны к исполнению):

- A содержит корневую файловую систему;
- B содержит раздел подкачки;
- C создается такого же размера, что и весь слайс, — это позволяет утилитам, которым необходимо работать над всем слайсом (например, сканеру сбойных блоков), работать с разделом C (как правило, в создании подобной файловой системы особой необходимости нет);
- D создавался для специальных целей, но в настоящий момент не используется. Есть вероятность того, что некоторые утилиты могут некорректно работать при попытке обращения к D, поэтому sysinstall обычно не создает подобный раздел.

Полное имя раздела состоит из имени дискового устройства, номера слайса и буквенного обозначения раздела. В итоге оно выглядит примерно следующим образом: ad0s2a — корневой раздел на слайсе номер 2 ведущего (master) ATA-диска, расположенного на первом контроллере. Ниже даны коды дисковых устройств:

- ad ATAPI (IDE) диск;
- da SCSI диск;
- acd ATAPI (IDE) CDROM;
- cd SCSI CDROM;
- fd флоппи-диск.

Вернемся к программе fdisk. Если вы использовали FIPS для переразметки, удалите созданный программой раздел (клавиша D). Теперь необходимо создать слайс. Если у вас имеется «чистый» диск, или вы хотите удалить все разделы и поверх разместить FreeBSD, нажмите A. Для создания нового слайса на свободном месте (не забудьте на него переместиться!) нажмите C. В секторах задается максимально доступный размер; если необходимо, измените его. Второй указанный параметр — это десятичная однобайтовая сигнатура раздела (для слайса это 165). Не забудьте сделать слайс загрузочным (S). Если вы ошиблись, можно выполнить откат (клавиша U); в противном случае сохраняйте внесенные изменения (Q) и переходите к формированию подразделов.

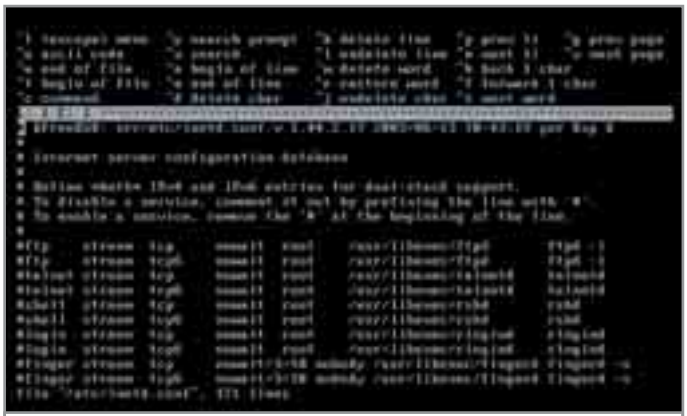
Теперь необходимо настроить загрузчик. Вам предлагается три варианта:

- BootMgr — используйте этот вариант, если на машине еще не был установлен никакой менеджер загрузки операционных систем. В таком случае им станет boot-менеджер FreeBSD.
- Standard — в этом случае в MBR записывается стандартный загрузчик, запускающий систему из активного раздела жесткого диска (который необходимо было выбрать на предыдущем этапе).
- None — содержимое MBR не меняется. Такой вариант используется в том случае, если на диске уже установлен другой загрузчик (LILO, GRUB, NTLoader, System Commander и т. п.).

Disklabel Editor

Создание подразделов выполняется при помощи программы Disklabel Editor. Операционной системе для работы необходимо, как минимум, два раздела: "/" (корневой) и swap-раздел. Однако для повышения эффективности работы с диском рекомендуется формировать файловые системы на нескольких разделах. Если вы затрудняетесь решить, сколько места отвести под каждый из разделов, воспользуйтесь функцией автоматической разметки слайса (клавиша A). »

Настройка сетевого интерфейса не вызовет у вас трудностей, если вы знакомы с основами теории TCP/IP-сетей



Редактирование файла /etc/inetd.conf позволяет настроить список сетевых сервисов, доступных через суперсервер inetd

» Если для /home не выделяется отдельного раздела то во FreeBSD, в отличие от Linux, используется свободное место на разделе /usr (опять же, если такой раздел создан): создается символическая ссылка /home, указывающая на /usr/home. Это удобно, так как позволяет снизить нагрузку на корневую файловую систему.

Следующее, на что необходимо обратить внимание, — режим Soft Updates. Эта опция способна значительно ускорить работу дисковых операций. Признаком того, что она включена, служит наличие значка +S рядом с типом файловой системы. Soft Updates существенно увеличивает скорость создания и удаления файлов путем использования кеширования. Хотя Handbook рекомендует использовать Soft Updates на всех дисках, но sysinstall при автоматической разметке слайса не устанавливает подобного режима для корневого раздела. Связано это с тем, что при использовании подобной технологии перед записью на жесткий диск могут наблюдаться задержки в несколько секунд (иногда до минуты). Если в этот момент система зависнет, можно потерять часть еще не записанных данных.

Если вам необходимо удалить раздел, используйте D; если же вы хотите удалить раздел и отдать освободившееся пространство соседнему разделу, стоящему в списке выше, используйте R. В любом случае отмену внесенных изменений можно выполнить при помощи U.

Начинающие администраторы часто затрудняются при выборе размеров разделов и их количества. Возможно, следующие советы помогут им более точно определить, что конкретно им нужно:

► Для корневого раздела желательно создавать отдельную файловую систему — это позволит монтировать ее в режиме «только для чтения» и снизит риск ее повреждения в случае сбоя.

► Создайте отдельный раздел для /tmp — здесь, как правило, хранится множество мелких файлов. Очень важно, чтобы раздел, отводимый под /tmp, имел достаточный размер и не переполнялся в процессе работы: это может привести к сбою как отдельных процессов, так и системы в целом. Любителям открывать и просматривать большие архивы с помощью Midnight Commander следует помнить, что распаковка архивов происходит в подкаталоге /tmp. Важную роль играют параметры создаваемой файловой системы (ключи -b и -f) , которые можно изменить в случае необходимости (Newfs Opts, горячая клавиша N).

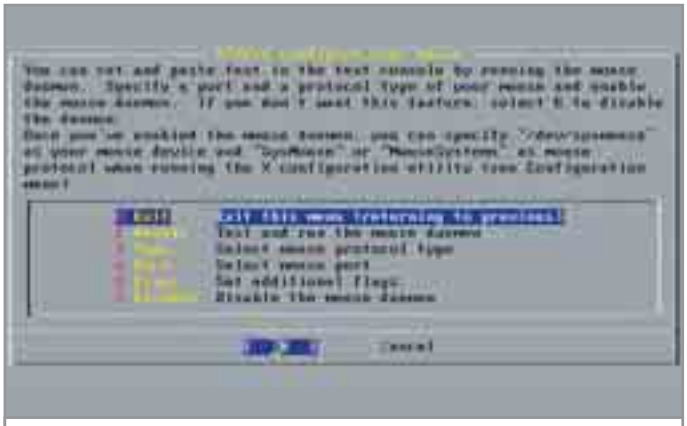
► По умолчанию используется пара значений: -b 16384 -f 2048. Для /tmp данное значение можно уменьшить вдвое, а для разделов, где будут храниться файлы большого размера, напротив — увеличить в четыре раза (-b 65535 -f 8192). Если вы затрудняетесь точно определить размер блока файловой системы, оставляйте значение по умолчанию.

► Желательно создавать отдельный раздел для /var. Если на машине будет работать почтовый сервер или другая сетевая служба с большой нагрузкой, имеет смысл создать отдельный раздел для каталога, где будут храниться рабочие файлы сервиса: например, /var/spool/mail для почтового сервера или /var/spool/squid — для кеша прокси-сервера. Если есть возможность, разместите подобный раздел на отдельном жестком диске, что также увеличит быстродействие.

► В режиме автоматической разметки слайса sysinstall выделяет недостаточно места под файловую систему /var. Для рабочей станции такие величины приемлемы, а вот при использовании FreeBSD в качестве сервера — нет.

► Если вы не можете точно сказать, насколько активно будет заполняться дисковое пространство, оставьте место для резерва — в дальнейшем, используя growfs, можно будет расширить размеры отдельных файловых систем.

► Размер области подкачки зависит от объема оперативной памяти. Если последний составляет до 32 Мбайт, то размер рассчитывается в пропорции 1:2; при объеме оперативной памяти от 32 до 128 Мбайт — как 1:1.5, от 128 до 256 Мбайт как 1:1. При больших объемах оперативной памяти (свыше 512 Мбайт) нет смысла создавать SWAP-раздел размером более 256 Мбайт: вряд ли он будет использован полностью. Однако помните, что приведенные закономерности получены опытным путем. Кроме того, старайтесь придерживаться следующих правил: если в системе несколько жестких дисков, располагайте раздел подкачки на самом быстром из них; по возможности располагайте раздел SWAP-раздел ближе к началу диска; его рекомендуется размещать между двумя разделами (например, корневым и /var) — теоретически это должно снизить »



Если sysinstall не смогла самостоятельно определить тип вашей мыши, укажите данные вручную

Внешний вид программы fdisk интуитивно понятен

» нагрузку на диск (хотя подобное возможно скорее всего на SCSI-дисках). Для завершения разметки нажмите [Q].

Выбор дистрибутивных наборов

Теперь вам необходимо выбрать, какой именно дистрибутивный набор установить. Новичкам, если позволяет свободное место на диске, рекомендуется выбирать вариант All. Если на компьютере должна быть установлена графическая среда, не забудьте выбрать X-User. Если вы планируете собирать новое ядро или модернизировать систему посредством CVSup (системы, позволяющей автоматически поддерживать FreeBSD в актуальном состоянии и вносить изменения; если вы не установили CVSup сразу, это можно сделать в дальнейшем при помощи порта cvsupit), вам понадобится один из наборов для разработчика — Developer и/или Kern-Developer. Если вы устанавливаете серверную версию FreeBSD и уверены, что X Window не понадобится, выберите вариант Developer. Обратите внимание, что пункты меню являются зависимыми: если вы выбрали Developer, автоматически будет выбран набор пакетов и Kern-Developer.

Если после этого будет выбран пункт Minimal, то установка других наборов будет отменена. Если вы четко представляете, что именно вам необходимо, выбирайте пункт Custom и отмечайте те наборы пакетов, которые считаете нужным использовать.

При выборе любого набора (кроме Custom) вам будет предложено установить дерево портов, которое занимает примерно 300 Мбайт. Набор портов обеспечивает простую установку множества дополнительных программ. Если у вас достаточно места на диске, обязательно соглашайтесь. Благодаря набору портов, в вашем распоряжении окажется около 10,5 тыс. программ, которые вы сможете без труда установить и использовать. Особенностью установки из портов является сборка из исходных текстов.

Выбор источника установки

Здесь вам необходимо указать источник установки — CD/DVD, DOS-раздел, FTP-, HTTP-, NFS-сервер, файловую систему FreeBSD или накопитель на магнитной ленте. По умолчанию предлагается CD/DVD.

После того как источник установки выбран, sysinstall предупреждает, что это последний шанс исправить изменения, внесенные в дисковую структуру. Выберите «No», и вы вернетесь к главному меню sysinstall; иначе начнется процесс установки.

Настройка после инсталляции

Конфигурирование сети

Первое, что необходимо настроить после установки, — параметры сетевого соединения. Перед вами появится окно с запросом о том, будут ли конфигурироваться сетевые устройства — Ethernet или SLIP/PPP.

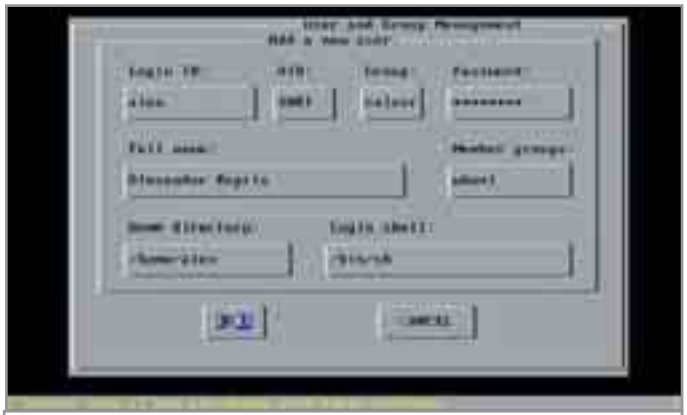
Выберите интерфейс, который вы хотите настроить. Sysinstall предложит вам использовать IP версии 6. Скорее всего, в вашей сети данная версия протокола еще не используется, поэтому откажитесь. Следующий вопрос будет касаться типа адресации, используемой на вашем компьютере, — динамической или статической. Если в сети работает DHCP-сервер и в нем присутствуют данные о вашей машине, ответьте утвердительно; в противном случае настройте параметры подключения вручную.

Необходимо указать следующие данные:

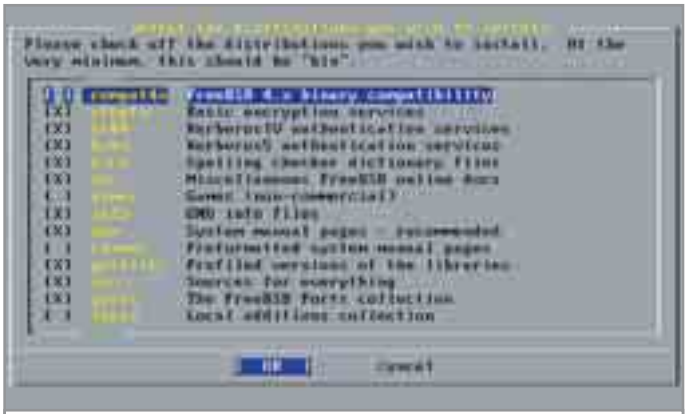
- Host — полное имя узла (например, vm.home);
- Domain — имя домена, в котором он расположен (например, home);
- IPv4 Gateway — IP-адрес шлюза по умолчанию (например, 192.168.0.1);
- Name server — IP-адрес DNS-сервера (например, 192.168.0.1);
- IPv4 address — IP-адрес узла (например, 192.168.0.4);
- Netmask — маску подсети;
- Extra options — дополнительные опции, которые используются утилитой ifconfig при активации сетевого интерфейса.

По окончании ввода данных система предложит вам активировать интерфейс. Затем вам придется ответить на ряд вопросов относительно тех функций, которые будет выполнять FreeBSD в сети:

- Будет ли компьютер шлюзом в сеть? При этом неважно, будет ли он шлюзом в локальную сеть или Интернет.
- Будут ли доступны сетевые сервисы? Программа установки предлагает сконфигурировать супер-сервер inetd, который контролирует попытки подключения к другим серверным программам компьютера и координирует сетевой трафик. Список сетевых служб перечислен в /etc/initd.conf. Вам будет предоставлена возможность от- »



Создание учетной записи непривилегированного пользователя не вызовет у вас особых проблем



Список пакетов, доступных на диске FreeBSD, — любой из них может быть установлен без соединения с Интернетом

» редактировать файл под свои нужды. Выход из редактора: Ctrl+[] и A.

► Будет ли разрешен анонимный FTP-доступ к компьютеру? Рекомендуется ответить «No». Вы всегда сможете сконфигурировать анонимный доступ по FTP позднее.

► Будет ли компьютер выступать в роли NFS-сервера? На ваше усмотрение.

► Является ли ваш узел NFS-клиентом? Если не знаете, отвечайте «No».

► Требует ли компьютер повышенных мер безопасности? Если ваш узел не требует максимального уровня защиты, выбирайте «No». Позднее вы сможете настроить функции защиты с учетом своих потребностей.

Если потребуется внести изменения в настройки, обращайтесь к уже известному файлу /etc/rc.conf.

Настройка консоли

Пришло время настроить консоль. Отвечайте утвердительно на вопрос о том, требуется ли вам настройка системной консоли.

Как минимум, необходимо настроить следующие параметры:

► Font — наш выбор «IBM 866». Хотя это шрифт в кодировке CP866, при правильном параметре Screenmap он будет корректно отображать кириллицу в локали KOI8-R.

► Keymap — выберите раскладку клавиатуры «Russia KOI8-R»

► Screenmap — для корректного отображения кириллических символов в консоли, укажите «KOI8-R to IBM866».

Для ускорения работы клавиатуры в пункте Repeat выберите параметр Fast. Выбирать хранитель экрана или нет (пункт «Saver»), решайте сами.

Часовая зона и совместимость с Linux

Следующим шагом необходимо настроить часовой пояс. На вопрос о том, настроены ли CMOS-часы на UTC, отвечайте «No». Затем выберите регион, страну и часовой пояс (или конкретный город, к которому идет привязка часового пояса).

Следующий запрос просит уточнить, есть ли необходимость в установке пакетов бинарной совместимости с Linux. В случае положительного ответа sysinstall установит в /usr набор разделяемых библиотек и других программ, необходимых для последующего запуска Linux-приложений.

Настройка мыши

Вопрос, который задается в этом случае, может в первый момент поставить вас в тупик: «Есть ли в системе не-USB-мышь?» Если у вас USB-мышь, выбирайте «No», иначе «Yes». Чтобы проверить, правильно ли sysinstall определила тип используемой мыши, выберите пункт Enable. Если по какой-то причине этого не произошло, укажите тип мыши и способ ее подключения вручную (пункты «Type» и «Port»).

Установка дополнительных программ

Ответьте «Yes» на вопрос, хотите ли вы просмотреть набор пакетов FreeBSD, доступный на компакт-диске. Пакеты разбиты на категории: если вы знаете, что именно ищете, то найдете это без труда.

Добавление пользователей

Теперь необходимо создать в системе учетную запись непривилегированного пользователя. Сперва создайте группу для такого пользователя (пункт «Group»). Например, localusr. Свободный GID (групповой идентификатор) система выделит сама. Затем введите данные пользователя (пункт «User»): логин, пароль, имя пользователя, группу, к которой он принадлежит. Если существует набор групп, членом которых пользователь должен быть, добавьте их в поле Member Groups (например, группа wheel). В этом случае пользователь сможет при необходимости получать привилегии пользователя root. Последнее, что вам необходимо ввести, — пароль для администратора системы. В отличие от пароля локальных пользователей, вы вводите его дважды.

На запрос о переходе в конфигурационное меню ответьте «No», после чего вы попадете обратно в основное меню sysinstall. Выберите пункт Exit Install и нажмите Enter. После этого перезагрузите систему.

А дальше?

Есть вопрос, который способен породить «бурю в стакане воды» среди фанатов командной строки, но его часто можно услышать от новичков: «Как запустить Midnight Commander?» Ответ на него прост — собрать из портов. Предлагаемый ниже мини-курс «молодого бойца» покажет вам, как это сделать.

Прежде всего необходимо донастроить работу с кириллицей. В файле /etc/ttys, описывающем типы терминалов, неверно указан тип терминала для локальных консолей — cons25. Из-за этого некорректно отображается псевдографика. Отредактируйте файл /etc/ttys, заменив cons25 на cons25r:

```
ee /etc/ttys
```

Теперь выполните следующие команды:

```
cd /usr/ports/misc/mc
make install
```

Сперва будет предпринята попытка обнаружить необходимый файл в /usr/ports/distfiles. В том случае, если файл не будет найден, программа установки попытается подключиться к Интернету и скачать требуемый файл оттуда. После того как будут получены все файлы (а в случае с mc их оказалось несколько -make, gettext, libiconv и т. д.), будет выполнена сборка и установка mc в /usr/local.

На сайте FreeBSD по адресу http://www.freebsd.org/ports/index.html находится удобная система поиска портов, которая поможет вам отыскать необходимый пакет.

Последнее, о чем хотелось бы здесь сказать, — это выбор локали для непривилегированного пользователя. Чтобы изменить значение по умолчанию, необходимо отредактировать файл ~/.login_conf, добавив следующие строки:

```
me:\
   :charset=KOI8-R:\
   :lang=ru_RU.KOI8-R:
```

Вот, в принципе и все. Теперь вы можете двигаться дальше, настраивая систему под свои нужды. Как настроить самые важные для нормального функционирования системы службы будет подробно рассказано в последующих статьях.

■ ■ ■ Александр Куприн

Сетевой колхоз

Общий доступ к файлам

Рано или поздно у любого пользователя возникает необходимость поделиться своими файлами с другими клиентами локальной сети. Можно, конечно, организовать файловый обмен посредством WWW- и FTP-протоколов (и вы наверняка это уже сделали). Но существуют гораздо более удобные способы. Итак, встречайте — NFS.

Сетевая файловая система (NFS — Network File System) обеспечивает совместное использование файлов в ОС Unix. В таких операционных системах, как Windows или Mac OS, есть свои механизмы совместного использования файлов, позволяющие подключенным к сети компьютерам обращаться к файлам на удаленных машинах так, как если бы они находились на их собственном диске. Система NFS обеспечивает те же преимущества, а кроме того — ряд возможностей, отсутствующих в других протоколах совместного использования файлов.

Немного теории

ОС Windows использует для совместного доступа к файлам протокол NetBIOS, а Mac OS — протокол AppleTalk. Оба этих протокола двухточечные: каждая система сообщает о своем присутствии в сети широковещательной рассылкой, и все машины могут динамически монтировать ресурсы друг друга, предоставленные для общего доступа. Система NFS отличается от них тем, что использует протокол типа «клиент-сервер», явно выделяя серверы, предоставляющие ресурсы в совместный доступ. Эти ресурсы в свою »

» очередь могут быть смонтированы удаленными клиентами NFS. Таким образом, объем передаваемой по сети информации уменьшается за счет отсутствия многочисленных ненужных запросов и ответов на них. Кроме того, сервер явно определяет, какие клиенты могут к нему подключаться, в зависимости от имени хоста или IP-адреса. Еще одно полезное свойство системы NFS состоит в том, что она не зависит от широковещательной рассылки в локальной сети, использующейся для выявления серверов. Поэтому ее можно использовать по Интернету точно так же, как и в локальной сети. Помимо этого, NFS отслеживает целостность передаваемых данных, уменьшая вероятность их потери.

Для операционной системы FreeBSD NFS — такая же файловая система, как и любая другая. Общий ресурс NFS вы можете смонтировать по сети точно так же, как дискету или раздел жесткого диска. Общие ресурсы могут даже автоматически монтироваться при обращении к ним, если клиентская система настроена соответствующим образом (об этом мы расскажем немного позже).

FreeBSD можно сконфигурировать для работы в качестве сервера NFS, клиента NFS или и того, и другого одновременно.

Конфигурирование сервера NFS

Настройка FreeBSD для работы в качестве сервера NFS требует добавления всего одной строки в файл /etc/rc.conf:

```
nfs_server_enable = "YES"
```

Убедитесь, что параметр portmap_enable имеет значение "YES" (как это было установлено по умолчанию, если вы его не изменили). Демон portmap необходим для функционирования системы NFS, поскольку NFS-серверу требуется механизм сообщения клиентам о том, к какому именно порту подключаться.

После установки этих опций и перезагрузки ОС FreeBSD предоставит через NFS общие ресурсы, указанные в файле /etc/exports. В нем должны быть перечислены каталоги, которые необходимо отдать для общего доступа через NFS, а также пользователи и хосты, которые будут иметь право доступа к ним. Если файл /etc/exports не существует или недоступен для чтения при запуске сети, система NFS не запустится.

Полный формат файла /etc/exports описывается на страницах справочного руководства man exports. Строка экспорта должна состоять из одного или нескольких имен каталогов, которые экспортируются (предоставляются для общего доступа), опций экспорта и необязательного списка хостов (задаваемых IP-адресом, именем сети, сетевой группой или по имени), которым разрешается использовать соответствующие каталоги. Например, следующая строка предоставляет в общий доступ каталог /home и все его подкаталоги для любого подключившегося хоста:

Еще один способ сконфигурировать систему как NFS-клиент: использовать универсальную утилиту sysinstall

```
/home    -alldirs
```

Учтите, что опция –alldirs может указываться только в том случае, если общий ресурс является точкой монтирования физической файловой системы (например, /usr или /home). В противном случае доступ к ресурсу предоставлен не будет.

Общий ресурс, доступ к которому (только для чтения) могут получить три указанных хоста, можно задать следующим образом:

```
/usr  -ro –alldirs office.domain.ru man-
      agers.domain.ru 192.168.0.16
```

После внесения изменений в файл /etc/exports необходимо перезапустить систему NFS. Для этого нужно указать идентификатор соответствующего процесса, содержащийся в файле /var/run/mountd.pid, например, с помощью команды:

```
# kill –HUP `cat /var/run/mountd.pid`
```

Для получения списка всех имеющихся общих ресурсов и прав доступа к ним можно использовать команду showmount. Проверить, правильно ли настроен файл /etc/exports, можно следующим образом:

```
# showmount –e
Exports list on localhost:
/usr              Everyone
/home/lena        192.168.0.47
/home/vasya       192.168.0.49
/                 192.168.0.1
```

»

Автоматизация доступа

Работа с группой

Чтобы более гибко управлять доступом к ресурсам NFS, в файле /etc/netgroup можно указать группы хостов. Группа задается следующим образом:

```
имя_группы (хост, пользователь, домен)
(хост, пользователь, домен) …
```

Например, для создания группы friends, содержащей три определенных хоста (с именами friend1, friend2, friend3), необходимо добавить следующую строку:

```
friends (friend1,,) (friend2,,) (friend3,,)
```

Группа на базе имен пользователей может иметь такой вид:

```
office (,kirill,) (,lena,) (,vasya,)
```

Затем можно использовать любое из этих имен групп из файла /etc/netgroup вместо имен хостов в файле /etc/exports, чтобы выделить общий ресурс NFS только членам нужной группы.

Конфигурирование клиента NFS

Если вы собираетесь монтировать общие ресурсы NFS с других серверов, необходимо сконфигурировать вашу систему в качестве клиента. С технической точки зрения это совсем не обязательно — можно монтировать общий ресурс NFS примитивным способом без всяких предварительных настроек. Однако конфигурирование системы в качестве клиента NFS предоставляет дополнительные возможности и гарантирует быструю и надежную работу.

Для настройки клиента NFS включите в файл /etc/rc.conf следующую строку:

```
nfs_client_enable = "YES"
```

Эта установка включает демон ввода/вывода NFS, nfsiod, помогающий ускорить выполнение клиентских запросов и настраивающий несколько параметров ядра, чтобы уменьшить время доступа. Он не обязателен для функционирования клиента NFS, но ускоряет работу посредством проведения асинхронных операций чтения/записи. «Опережающее чтение» и «запись с задержкой» выполняются в фоновом режиме, что избавляет от необходимости ожидать завершения каждого последовательного шага процесса.

Для запуска демона без перезагрузки выполните команду:

```
# nfsiod –n 4
```

Монтирование удаленных файловых систем

Монтирование общего ресурса NFS выполняется с помощью команды mount_nfs, которая является сокращенным вариантом стандартной команды mount –t nfs. Как правило, этой команде передается два аргумента — имя хоста и имя общего ресурса в виде комбинированной строки, а также локальная точка монтирования:

```
# mount_nfs office:/home /home2
```

```
# df
Filesystem      1K-blocks    Used      Avail      Capacity  Mounted on
/dev/ad0s1a     992239       54353     858597     6%        /
/dev/ad0s1f     26704179     4872963   19694882   20%       /home
...
procfs          4            4         0          100%      /proc
office:/home    9924475      1642343   7488174    18%       /home2
```

Таблица 1. Результат выполнения команды df

При успешном монтировании на экран не выдается никаких сообщений. Проверить, успешно ли произошло монтирование, можно командой df (см. таблицу 1)

Файловая система будет оставаться смонтированной до тех пор, пока не будет явно демонтирована с помощью команды umount:

```
# umount /home2
```

Как и для других типов файловых систем, можно добавить описание монтируемых ресурсов NFS в файл /etc/fstab, что впоследствии упростит сам процесс монтирования (таблица 2). При наличии такой записи можно смонтировать файловую систему NFS простой командой:

```
# mount /home2
```

Более подробную информацию об опциях монтирования можно получить в справочной системе с помощью команды

```
# man mount_nfs.
```

Автоматическое монтирование

Демон автоматического монтирования amd делает работу с общими ресурсами NFS еще более удобной. Он позволяет монтировать их (а на самом деле — и все типы файловых систем) динамически при переходе в необходимый каталог, не вводя при этом никаких команд монтирования.

ОС FreeBSD обеспечивает простой способ настройки этого демона. Добавьте следующую строку в файл /etc/rc.conf:

```
amd_enable="YES"
```

При перезагрузке системы демон amd запустится с опциями, которые задаются параметром amd_flags. Стандартное значение этих опций обеспечивает автоматическое монтирование по имени всего содержимого каталогов /host или /net, которые автоматически будут созданы демоном amd. Демон можно также запустить вручную — с помощью команды:

```
# amd –a /.amd_mnt –l syslog /host
    /etc/amd.map /net /etc/amd.map
```

При работающем демоне amd перейдите с помощью команды cd в каталог /host и просмотрите его содержимое. Каталог пуст:

```
# cd /host
# ls
#
```

Однако попытайтесь получить каталог по имени, как если бы там уже была директория, имя которой совпадает с именем одного из серверов NFS в сети:

```
# ls office
# home
```

В каталоге /host действительно появился каталог office, а в нем — подкаталог home, содержащий то же самое, что и office:/home. Он только что автоматически смонтировался в каталог /host при первом обращении к нему.

Для еще большего удобства можно создать ссылку на нужный каталог:

```
# ln –s /host/office/home /home2
```

С этого момента при переходе в каталог /home2 общий ресурс office:/home будет монтироваться автоматически, и вы получите доступ к нужным файлам. Неиспользуемый общий ресурс будет автоматически демонтирован, что также удобно. »

# Device	Mountpoint	Fstype	Options	Dump	Pass#
office:/home	/home2	nfs	rw,-T,-i,noauto	0	0

Таблица 2. Описание монтируемых ресурсов NFS

» В случае необходимости можно создавать намного более сложные карты монтирования для демона amd, задавая записи в файле /etc/amd.conf. Обратите внимание: после установки ОС FreeBSD этого файла не существует — его нужно создать самостоятельно следующим образом:

```
#touch /etc/amd.conf
```

Подробные сведения о его формате и предоставляемых им возможностях можно посмотреть на страницах справочного руководства man amd.conf.

Взаимодействие с Windows-сетью

А что делать, если большинство пользователей вашей локальной сети работают на компьютерах под управлением ОС Windows? Система NFS — отличное решение проблемы совместного использования файлов Unix-машинами, однако она мало распространена в большинстве пользовательских операционных систем. ОС Windows поддерживает ее только с помощью приложений сторонних производителей. Поэтому при включении компьютера под управлением ОС FreeBSD в существующую сеть необходимо, чтобы ОС FreeBSD поддерживала те же методы совместного использования файлов, что и Windows.

Подобные средства совместного использования файлов изначально не встроены в FreeBSD. Однако дополнительный пакет под названием Samba предоставит вашей машине под управлением этой ОС возможность работать в качестве файл-сервера Windows и участвовать в совместном использовании файлов с реальными клиентами Windows.

Введение в систему Samba

Система Samba — это некоммерческий проект с открытым исходным кодом, который позволит вашей системе FreeBSD пользоваться всеми преимуществами совместного доступа к файлам Windows, включая появление машины в списке ресурсов сети, защиту подключений на основе доменов NT и регистрацию пользователей, а также поддержку сетевых служб печати и других удобств.

Установка и конфигурирование системы Samba

Система Samba доступна среди портированных приложений в каталоге /usr/ports/net/samba или в виде пакетов на сайте производителя. Рекомендуется установить приложение из системы портов, предварительно обновив ее следующим образом:

```
# cd /usr/ports/net/samba
# make install clean
```

В состав пакета входит единственный конфигурационный файл — smb.conf.default, который для работы необходимо скопировать в smb.conf. Сценарий запуска /usr/local/etc/rc.d/samba.sh.sample также необходимо переименовать в samba.sh.

В самом простом случае запуска системы Samba необходимо только отредактировать файл smb.conf, изменив строку рабочей группы в соответствии с именем рабочей группы или домена, в который должна входить машина:

Ресурсы Windows-сети можно просматривать файловым менеджером Nautilus

```
# workgroup = Имя домена или рабочей
      группы NT, например, WORKGROUP
workgroup = MY_WORKGROUP
```

Теперь при перезагрузке системы Samba будет запускаться автоматически.

Интернет-SWAT

Основной файл конфигурации системы Samba — /usr/local/etc/smb.conf, в котором можно устанавливать десятки различных параметров и задавать общие ресурсы. Каждая опция неплохо описана в комментариях в файле примера smb.conf.default; однако с ходу разобраться в содержимом этого файла непросто: опций для установки там очень много (все они детально описаны на страницах справочного руководства man smb.conf) и между ними есть масса тонких различий.

Существует также альтернативный метод формирования и настройки файла smb.conf. Речь идет о системе SWAT (Samba Web Administration Tools), которая »

Команда разработчиков некоммерческого пакета Samba

С помощью веб-интерфейса программы SWAT Samba легко настроить, но необходимо помнить о безопасности

» входит в состав портированного пакета Samba и позволяет конфигурировать его через веб-браузер. В результате существенно упрощается работа с файлом конфигурации, снижается вероятность появления в нем ошибок. Недостаток же этой системы, к сожалению, органически присущ всем веб-приложениям — это существенная угроза защите. Система SWAT аутентифицирует пользователей с помощью базы данных пользователей системы FreeBSD, хранящейся в /etc/master.passwd, и посылает эти данные по сети в явном виде, где они могут быть перехвачены злоумышленником. Снизить риск можно несколькими способами.

► Обращайтесь к системе SWAT только с локального хоста (localhost). Это предотвратит пересылку информации по сети.

► Работайте только под защитой брандмауэра, запрещающего или ограничивающего передачу информации извне.

► По умолчанию файл smb.conf принадлежит пользователю root, поэтому браузер должен регистрироваться в системе SWAT, передавая пароль пользователя root, который посылается по сети в явном (нешифрованном) виде вместе с каждым HTTP-запросом к SWAT. Никогда не делайте этого в сети, в которой может находиться потенциальный злоумышленник.

► Создайте фиктивного пользователя (например, smbowner) и сделайте его владельцем файла smb.conf (с помощью команды chown). Работая с системой SWAT, регистрируйтесь в качестве этого пользователя, а не как пользователь root. Не используйте это имя пользователя для решения других задач на сервере, не давайте ему никаких привилегий, запретите доступ к командному интерпретатору и не создавайте ему домашний каталог.

Поддержку SWAT можно включить, добавив следующую строку в файл /etc/services:

```
swat    901/tcp
```

Затем добавьте следующую строку в файл /etc/inetd.conf:

```
swat stream tcp nowait root
      /usr/local/sbin/swat swat
```

И наконец перезапустите демон inetd:

```
# killall –HUP inetd
# inetd
```

Теперь можно обращаться к системе SWAT по адресу URL http://localhost:901. Она запросит имя пользователя и пароль.

Система SWAT позволяет изменять предоставляемые для общего доступа ресурсы и принтеры, а также глобальные настройки Samba. С ее помощью вы можете также узнавать о текущем состояние сервера и управлять пользователями системы.

Если же вы предпочитаете делать все собственными руками, можете внести изменения в файл конфигурации Samba посредством прямого редактирования файла smb.conf в любимом текстовом редакторе.

Предоставление каталогов для общего доступа

Немало примеров конфигурирования каталогов общего доступа можно найти в файле smb.conf.default. Чтобы задействовать их, внесите соответствующие изменения (сняв комментарии на нужных строках) в файл smb.conf, а затем перезапустите сервер Samba:

```
# /usr/local/etc/rc.d/samba.sh stop
# /usr/local/etc/rc.d/samba.sh start
```

Перезапустить Samba можно также через веб-интерфейс системы SWAT.

Чтобы предоставить какой-либо каталог в общее пользование, нужно определить его как общий ресурс:

```
[public]
comment =  Общие файлы
path       =  /usr/local/share/samba-public
public     =  yes
writeable =  yes
printable =  no
write list =  @users
```

При наличии таких строк клиент будет видеть в сетевом окружении ресурс public вашего компьютера. Однако пока пользователь не будет аутентифицирован и не окажется членом Unix-группы users, файлы ресурса будут доступны ему только для чтения.

По умолчанию определяется и включается общий ресурс [homes] — это специальный встроенный ресурс, обеспечивающий доступ к домашнему каталогу каждого пользователя, определенного на сервере Samba:

```
[homes]
      comment    =  Домашние каталоги
      browseable =  no
      writeable  =  yes
```

Этот ресурс установлен как «не просматриваемый», но если клиент подключается от имени пользователя, имеющего домашний каталог на сервере Samba, то каталог появится как один из общедоступных ресурсов. Домашние каталоги других пользователей не будут видны.

Совместная печать

Как и [home], [printers] — специальный общий ресурс, немного отличающийся от остальных. В ОС FreeBSD все принтеры, определенные в файле /etc/printcap, доступны всем пользователям. По умолчанию общий ресурс [printers] настроен так:

```
[printers]
      comment    =  Samba-принтер
      path       =  /var/spool/samba
      browseable =  no
# Установите public =yes, чтобы разре-
      шить печать пользователю guest
      guest ok   =  no
      writeable  =  no
      printable  =  yes
```

»

» Управление доступом

В системе Samba имеется два популярных способа управления доступом — на уровне пользователей и на уровне общих ресурсов. Стандартное управление доступом происходит на уровне пользователей и задается опцией security в файле smb.conf:

```
security = user
```

При таком управлении доступом клиент при начале соединения предоставляет серверу пару из имени пользователя и пароля. Если сервер примет клиента, ему будут доступны все общие ресурсы.

При управлении доступом на уровне общих ресурсов клиент может подключаться к серверу Samba без всякой аутентификации. Клиенту может быть отказано в доступе, только если его IP-адрес не указан в файле smb.conf (в строке hosts allow). При таком способе управления доступом клиент свободно может получить только те общие ресурсы, которые помечены параметром public=yes, но домашние каталоги пользователей по-прежнему будут защищены именем пользователя и паролем.

Подробности об организации защиты общих ресурсов вы можете прочесть в файле документации /usr/local/share/doc/samba/textdocs/security_level.txt.

Гостевой пользователь

Доступ к некоторым службам Samba, в частности, к службе печати, имеет смысл предоставить любому пользователю, независимо от аутентификации. Для этого нужно использовать так называемую гостевую учетную запись для пользователя, которому необходим доступ только к одной конкретной службе. Назначение «гостевых пользователей» рекомендуется в основном для серверов Samba, работающих с защитой на уровне ресурсов, поскольку доступ гостя к каждому ресурсу предоставляется или запрещается отдельно.

Чтобы разрешить работу такому пользователю, раскомментируйте строку guest account в файле smb.conf:

```
guest account = pcguest
```

Теперь необходимо добавить в систему учетную запись pcguest с помощью команды adduser.

Файловая система smbfs

Совместное использование файлов по протоколу SMB может быть двусторонним. Удаленные общие ресурсы SMB можно монтировать так же, как и любую другую файловую систему. Речь идет о файловой системе smbfs, доступной в портированных приложениях в каталоге /usr/ports/net/smbfs.

Чтобы смонтировать файловую систему SMB с помощью smbfs, используйте команду mount_smbfs с рядом простых опций. Опция -I задает имя хоста или IP-адрес, а два оставшихся аргумента — имя удаленного общего ресурса (в формате //пользователь@<имя NETBIOS>/<имя ресурса>) и локальную точку монтирования. Полезен также ключ -E, указывающий кодировку, например koi8-ru:cp866. Для монтирования общего ресурса public с Windows-машины office в каталог /mnt /public команда будет выглядеть так:

```
# mount_smbfs –I 192.168.0.13
    //guest@office/public /mnt/public
```

Будет запрошен пароль. Используйте пустой пароль, если ресурс открыт для общего доступа, или введите соответствующий пароль, если ресурс защищен.

Аналогично предыдущим примерам можно добавить общий ресурс SMB в файл /etc/fstab с помощью следующей строки:

```
//guest@office/public /smb/public
    smbfs rw,noauto 0 0
```

Сценарий /usr/local/etc/rc.d/smbfs.sh будет монтировать этот общий ресурс при загрузке системы FreeBSD.

Окна в сеть

Еще более простой способ получить доступ к ресурсам Windows-сети — использовать возможности оболочек GNOME или KDE. Эти графические среды уже имеют встроенный Samba-клиент. Все, что остается сделать, — набрать в адресной строке файлового менеджера такую строку:

Связка из демона LISa и Konqueror позволяет организовать в KDE подобие «Сетевого окружения» Windows

```
smb://Имя_Ресурса
```

В появившемся окне требуется ввести имя пользователя, пароль и домен Windows. Если никаких ограничений на доступ к ресурсу нет, имя пользователя должно быть guest, а остальные поля можно не заполнять. Файловый менеджер Nautilus 2.6.1 прекрасно отображает русские названия файлов и папок. Однако у такого способа есть и недостаток, заключающийся в том что ресурс не монтируется непосредственно в файловую систему FreeBSD. Поэтому чтобы прикладные программы могли открыть файлы с сетевых ресурсов, предварительно их придется скопировать на локальный компьютер. Нельзя не упомянуть также о том, что в стандартном для графической среды KDE браузере Konqueror существует аналог «Сетевого окружения» — Lan Browser. Для его функционирования необходимо дополнительно установить и настроить демон LISa. Скачать исходные коды и более подробно ознакомиться с конфигурированием демона можно на странице проекта http://lisa-home.sourceforge.net.

Как видите, работать в FreeBSD с сетевыми ресурсами очень легко. Будь то специфичная для Unix-подобных систем NFS или Windows-сеть — в любом случае простые методы настройки и наличие большого объема справочной информации помогут вам сделать систему еще удобнее и проще в использовании.

■ ■ ■ Александр Соловков

Раздача имен

Система именования серверов

Глобальная сеть Интернет подразделяется на домены, каждый из которых обслуживает различные группы пользователей. Управление этими доменами осуществляется с помощью DNS-сервера, получившего название корневого сервера имен. Такие серверы есть на каждом уровне сети и заканчиваются они локальным DNS-сервером.

Когда набирается адрес, локальный DNS-сервер просматривает свою базу данных и кеширует требуемую информацию. Если она не содержит IP-адреса, он передает запрос корневому серверу имен, а тот возвращает адрес соответствующего сервера имен. Локальный DNS-сервер, в свою очередь, обращается с запросом к серверу имен в поисках адреса сервера на следующем уровне, и далее процесс повторяется. Например, если вы хотите обратиться на узел http://www.mail.ru, ваш DNS-сервер обращается к серверу домена .ru в поисках адреса сервера имен mail в данном домене. Локальный DNS-сервер использует адрес, полученный по этому запросу, для обращения к серверу mail.ru в поисках адреса хоста.

DNS (Domain Name Service) — это часть семейства протоколов и утилит TCP/IP (слово «domain» в названии протокола относится к доменам в Интернете, а не к доменной модели NT). Существует много версий DNS-серверов, работающих на разных операционных системах. В этой статье будет рассмотрена версия, построенная на Unix.

Пространство доменных имен реализовано в виде распределенной базы данных, включающей в себя DNS-серверы и DNS-клиенты (resolver), объединенные общим протоколом запросов к базе, и обмена информацией между серверами. Информация, индексированная доменным именем, хранится в записях ресурсов RR (Resource Records). Запись ресурса имеет класс (в настоящее время используются записи Интернета — IN), тип записи (определяет характер хранимой информации) и собственно информацию. В частности, для каждого ресурса хранится максимально допустимое время кеширования полученной информации TTL (Time To Live). Совокупность запи- »

» сей ресурсов, имеющих совпадающие доменное имя, класс и тип, называется набором записей ресурсов (RRset).

Основным типом хранимой информации являются IP-адреса. Доменному имени может соответствовать несколько IP-адресов (несколько сетевых интерфейсов на компьютере); одному адресу может соответствовать несколько имен (синонимы). Порядок выдачи записей при запросе не обязан соответствовать порядку записей при описании зоны.

Уполномоченный (authoritative) сервер обладает полной информацией об определенной зоне. Адреса уполномоченных серверов зоны (домена, объемлющего зону) указываются в информации о родительском домене. Уполномоченные серверы делятся на первичные (primary master) и вторичные (secondary master, slave). Первый загружает данные зоны из локального источника (обычно из файла). Второй получает данные зоны от другого уполномоченного сервера (обычно, хотя и не обязательно, от первичного сервера). Этот процесс называется передачей зоны (zone transfer). При недоступности исходного уполномоченного сервера вторичный может загружать зону из резервной копии, предусмотрительно сохраненной в файле.

Наличие нескольких уполномоченных серверов позволяет разделить нагрузку и обеспечить защиту от сбоев. DNS-сервер (процесс) может быть уполномоченным сразу для нескольких зон или ни для одной (кеширующий сервер). При этом для одних зон он может быть первичным, а для других — вторичным. Уполномоченный сервер, указанный в родительском домене (при делегировании зоны), но не описанный в записи самой зоны, называется скрытым (stealth) уполномоченным сервером. Скрытым может быть и первичный сервер (hidden primary). Такой вариант используется тогда, когда первичный сервер находится за сетевым экраном. Неверная настройка, при которой уполномоченный сервер, указанный в родительском домене (при делегировании зоны), отказывается признавать себя уполномоченным, называется «некорректным делегированием» (lame delegation, lame servers).

Простейший сервер имен

На протяжении долгого времени самым популярным методом разрешения имен компьютеров и IP-адресов в Интернете являются коды BIND. Организация Internet Software Consortium (ISC) разрабатывает и сопровождает исходные коды, которые можно бесплатно получить на сайте ftp://ftp.isc.org. Всего имеется три ветви: BIND 4, BIND 8 и BIND 9 (правда, некоторое время назад ISC прекратила поддержку первого из них). Одни из последних версий — BIND 4.9.8, BIND 8.3.3 и BIND 9.2.3.

Кеширующий DNS-сервер служит для запоминания запросов, уходящих в глобальную сеть. Это значительно уменьшает время ожидания ответа при следующем запросе, особенно если соединение медленное или внутренняя сеть обширна. Например, пользователь 1 отправил запрос на определение IP-адреса хоста www.mail.ru. В этом случае кеширующий DNS-сервер определил адрес по алгоритму, описанному выше, и запомнил его. Через некоторое время пользователь 2 тоже отправил запрос на определение того же IP-адреса; теперь DNS-сервер вместо того, чтобы повторять алгоритм, просто достает запись из своего кеша, чем сильно снижает нагрузку на внутреннюю и внешнюю сеть, а также другие серверы. Сначала нужно установить сервер:

```
# cd /usr/ports/dns/bind9;
      (устанавливаем из портов)
# make && make install
```

Далее нужно соответствующим образом отредактировать файл named.conf, располагающийся в каталоге /etc/namedb:

```
options {
      directory "/var/named"; (рабочий каталог с конфигурационными файлами)
      // query-source port 53; (для работы через firewall надо раскомментировать).
};
zone "." {
      type hint;
      file "root.hints";
};
zone "0.0.127.in-addr.arpa" {
      type master;
      file "zone/127.0.0";
};
```

Файл, названный /var/named/root.hints, должен находиться в указанной директории. Он описывает имена корневых серверов имен по всему миру. Время от времени их список изменяется; обновлять его можно с помощью программы dig.

Следующий раздел в named.conf — обратная зона. Просто создайте файл под названием 127.0.0 в поддиректории zone:

```
@ IN SOA ns.myname.ru. hostmaster.
      ns.myname.ru (
1     ; Serial
8H    ; Refresh
2H    ; Retry
1W    ; Expire
1D)   ; Minimum TTL
IN    NS  ns.myname.ru.
1     IN  PTR       localhost.
```

Теперь нам необходимо, чтобы файл /etc/resolv.conf выглядел так:

```
search name.myname.ru myname.ru
nameserver 127.0.0.1
```

Далее следует запуск named:

```
# named
```

В логах (/var/log/messages) должно быть примерно следующее:

```
# tail -f /var/log/messages
May 15 13:26:17 myname named[456]:
      Ready to answer queries
```

На этом настройку простого DNS-сервера можно считать завершенной.

Настройка первичного DNS-сервера

Любой человек может зарегистрировать в Интернете свой домен и даже не один. Независимо от места проживания, вы можете »

» обратиться в ближайший офис ISP и, заплатив $20 за год, воспользоваться услугами регистрации доменных имен. После этого вы зададитесь вопросом: как сделать так, чтобы этот домен полноценно работал в Интернете? Есть два пути: можно обратиться с просьбой о поддержке и размещении домена в организацию, где он был зарегистрирован, или любую организацию, которая занимается подобными услугами; а можно разместить домен на своей технической площадке (для этого потребуется выделенная линия в Интернет и отдельный компьютер с установленной операционной системой, в нашем случае — FreeBSD).

Допустим, что выполнены все требования для самостоятельной поддержки домена. Необходимо выбрать версию программы BIND (на данный момент стабильной версией является BIND-9.2.3). Ее можно найти здесь: ftp://ftp.isc.org/isc/bind9/9.2.3/bind-9.2.3.tar.gz. Полученный файл распаковываем (# tar -zxvf bind-9.2.3.tar.gz) и приступаем к конфигурации (#./configure --prefix=/usr/local/named (директория установки)). Команда with-openssl используется, если вы применяете криптографические методы шифрования данных. Также можно воспользоваться дополнительными опциями, посмотрев их командой: #./configure --help. Далее: #make ; make install. И снова редактируем файл named.conf , но синтаксис будет отличаться от предыдущего:

```
options {
        directory "/var/named"; (рабочий каталог с конфигурационными файлами)
        // query-source port 53; (для работы через firewall надо раскомментировать).
};
zone "." {
        type hint;
        file "root.hints";
};
zone "myname" {
        type master;
        file "zone/myname.dns";
};
zone "85.85.85.in-addr.arpa" {
        type master;
        file "zone/85.85.85"; };
zone "0.0.127.in-addr.arpa" {
        type master;
        file "zone/127.0.0";
};
```

Далее необходимо отредактировать файл зоны (давайте назовем его /etc/namedb/zone/ myname.dns):

```
@ IN SOA ns.myname.ru;
        (уполномоченный сервер)
hostmaster.ns.myname.ru; (адрес
        электронной почты администратора)
(
2004011501; (номер версии)
8H    ; (интервал обновления зоны)
2H    ; (интервал попытки обновления
        зоны)
1W    ; (интервал истечения полномочий)
1D)   ; ttl
        IN NS ns.myname.ru
        IN NS ns.server.ru; (вторичный
        DNS-сервер, который должен
        находиться в другой сети класса C)
        IN MX 10 smtp.myname.ru; (почтовый
        сервер; цифра 10 указывает его
        приоритет)
        IN MX 20 smtp1.myname.ru;
        (вспомогательный почтовый сервер)
myname.ru. IN A 85.85.85.121;
        (IP-адрес хоста с именем myname.ru)
www     IN A 85.85.85.129; (IP-адрес
        web-сервера)
ftp     IN CNAME www
```

Следующий шаг: редактируем файл zone/85.85.85. Это так называемый файл обратной зоны (она служит для преобразования имен в IP-адреса).

```
@ IN SOA ns.myname.ru. hostmaster.ns.
        myname.ru. (
2004011501 ; (номер версии)
8H    ; (интервал обновления зоны)
2H    ; (интервал попытки обновления)
1W    ; (интервал истечения полномочий)
1D)   ; Minimum TTL
      IN NS     ns.myname.ru.
121 IN PTR    myname.ru.
129 IN PTR    www
```

PTR — это доменное имя для соответствующего IP-адреса (в данном случае пишется только последний октет). Для поиска доменных имен по IP-адресам используется in-addr.arpa. Его поддоменами являются домены с простыми именами от 0 до 255, соответствующими старшему октету IP-адреса. Их поддоменами являются домены с простыми именами от 0 до 255, соответствующие второму октету IP-адреса, и так далее, до четвертого октета. Таким образом, IP-адрес оказывается записанным в доменном имени в обратном порядке. Например, адресу 195.161.72.28 соответствует доменное имя 28.72.161.195.in-addr.arpa. (и значение PTR — deol.deol.ru). Обратная запись необходима для более легкого делегирования зон в соответствии с выделением IP-адресов.

Зоны верхнего уровня в домене in-addr.arpa. делегированы IANA региональным регистраторам (RIR — Regional Internet Registrator) вместе с блоками IP-адресов. Отображение адресов в имена может быть обязательным для работы некоторых сервисов в Интернете: нет отображения — нет обслуживания. В нашем случае мы создаем домен для всей сети класса C (85.85.85.0/24). На этом настройка первичного DNS-сервера закончена. Запускаем named и смотрим логи.

Настройка вторичного DNS-сервера

Настройка DNS-сервера данного типа сводится к редактированию файла named.conf, где описывается зона, для которой мы настраиваем данный сервер как вторичный:

```
zone "myname2.ru" {
        type slave;
        file "zone/myname2.dns";
        masters { ip адрес primary
        dns сервера }; };
```

После этого перезапускаем named. Он автоматически создает файл myname2.dns.

Защита DNS-сервера

По умолчанию DNS-сервер выполняет запросы рекурсивно. С точки зрения безопасности это может вызывать некоторые про- »

» блемы. При получении запроса на разрешение имени от клиента или другого DNS-сервера (например, если требуется найти IP-адрес сервера www.exampleco.com) DNS-сервер проверяет свой локальный кеш имен. В случае неудачи он пытается получить необходимую информацию от других серверов. Если сервер получит ложную или недостоверную информацию, она все равно будет передана запрашивающему ее клиенту. Точно так же, если DNS-сервер поддерживает пространство имен домена для доступа в Интернет и отвечает на запросы, приходящие оттуда, любой компьютер с выходом в сеть может использовать ресурсы этого сервера для опроса других доменов.

В секцию options файла named.conf рекомендуется добавить специальное условие, в котором будут указаны пользователи, имеющие право на выполнение рекурсивных запросов.

```
allow-recursion {
        85.85.85.0/24;
    };
};
```

В этом примере компьютеры внутренней сети из диапазона адресов 85.85.85.0/24 могут использовать DNS-сервер для разрешения имен Интернета. Также следует отключить ряд рекурсивных функций DNS-сервера:

```
options {
A recursion no;
B fetch-glue no;
    };
```

Метка «A» предваряет условие, запрещающее локальному DNS-серверу использовать для разрешения имен другие серверы. Метка «B» предваряет условие, по которому локальному DNS-серверу запрещается выступать в роли транслятора чужих запросов на разрешение имен в записях Name Server (NS). Вместе с тем отключение рекурсивных функций у интернет-сервера BIND DNS приводит к тому, что внутренние DNS-серверы и пользователи не смогут выполнять разрешение имен Интернета при помощи этого сервера.

Для обеспечения безопасности многие организации стремятся скрыть свою внутреннюю инфраструктуру сети. Особенно это касается имен компьютеров и их IP-адресов из локальной сети, которые находятся в базе DNS-серверов. В то же время служба DNS должна выполнять разрешение имен внешних интернет-серверов (например, почтовых и веб-серверов). Метод, который позволяет поделить службу DNS на внешнюю и внутреннюю, называется разделением (split) DNS.

Способ разделения DNS основан на новой возможности девятой версии BIND, позволяющей разместить внутренний и внешний DNS на одном компьютере. Предположим, некая компания поддерживает внутренний домен myname.local (содержит имена внутренних компьютеров и их IP-адресов) и внешний — myname.ru (имеет отдельный файл зоны, содержащий имена внешних компьютеров и их IP-адреса). Обычный DNS-сервер смог бы использовать лишь один файл зоны для домена myname.ru. BIND 9 позволяет единственному DNS работать с несколькими файлами зон для одного и того же доменного имени. Такой сервер умеет отличать внутренних клиентов от внешних по их IP-адресам и использовать для ответов соответствующие файлы зон.

```
view internal-exampleco {
A match {
192.168.1.0/24; }; (адреса внутренней сети)
zone "myname.local"
{ type master; file
      "zone/myname.local.dns"; };
};
view external-exampleco {
B match any;
zone "myname.ru"
{ type master; file "zone/myname.dns"; };
 };
```

Пример конфигурации секции named.conf определяет файлы внутренней и внешней зоны домена myname для сервера BIND 9. DNS-сервер выполняет запросы клиентов в том порядке, в котором строки конфигурации размещены в секции View. Таким образом, добавляя в нее новую строку, следует обращать внимание на то, какое место она занимает по отношению к другим ее строкам-утверждениям. Первое утверждение относится к внутреннему домену, второе — к внешнему. »

Справочная информация

Загадочные аббревиатуры

SOA — эта запись может быть только одна. Описание зоны должно начинаться с записи данного типа, определяющей для указанного домена:

- первичный уполномоченный сервер (primary master);
- адрес электронной почты ответственного за зону (@ в почтовом адресе заменяется на точку, в конце добавляется точка);
- номер версии (32 бита; должен увеличиваться при каждом изменении; используется вторичным уполномоченным сервером для проверки необходимости обновления зоны; его принято записывать в виде даты и номера изменения в этот день в формате ГГГГММДДNN — например, 2004011501);
- интервал обновления зоны (в секундах) для вторичных уполномоченных серверов;
- интервал попытки обновления зоны (в секундах) при неудаче обновления;
- интервал истечения полномочий для вторичных уполномоченных серверов при неудаче обновления (в секундах).
- TTL — до RFC 2308 минимальное TTL для ресурсов зоны (оно же значение по умолчанию), после — время жизни отрицательного кеширования (не более трех часов).

NS — доменное имя уполномоченного сервера указанного домена; должно быть несколько серверов (в том числе и указанный в SOA). Имя не обязано лежать в том же домене.

A — представляет собой IP-адрес для указанного доменного имени.

CNAME — каноническое доменное имя для определяемого синонима; доменное имя-синоним не должно иметь других записей ресурсов; синонимы не должны использоваться в данных любых других ресурсов.

» Файл зоны внутреннего домена называется myname.local.dns, а файл зоны внешнего домена — myname.dns. Как только DNS-сервер получает запрос, он сначала пытается определить, попадает ли IP-адрес клиента в диапазон, отмеченный в области view идентификатором A. Если это так, то DNS-сервер считает, что запрос пришел из внутренней сети, и возвращает ответ на основании данных внутреннего домена. Если же нет, то DNS-сервер определяет, попадает ли IP-адрес клиента в диапазон, отмеченный идентификатором B. В нашем примере идентификатор B содержит условие с ключевым словом «any» (любой), поэтому сервер возвращает ответ с информацией о внешнем домене любому клиенту, чей IP не попадает в список внутренних адресов.

Чтобы использовать данную возможность, требуется расположить сервер BIND 9 на компьютере, доступном для клиентов как внутренней, так и внешней сети. Такой компьютер может служить сетевым экраном между Интернетом и интранетом или сервером в зоне DMZ. Хорошая зона DMZ имеет как внутренний сетевой экран, так и внешний. Внутренний экран открывает исходящий порт UDP 53 для DNS-запросов из внутренней сети, а внешний экран открывает для запросов входящий порт UDP 53.

Для обеспечения безопасности DNS-серверов можно внести в настройки BIND еще несколько дополнений и разрешить внутреннему серверу BIND отвечать только на те запросы, которые приходят с указанных IP-адресов:

```
options {
      allow-query {
          192.168.1.0/24;
      }; };
```

В приведенном примере сервер обрабатывает запросы только от клиентов из сетей 192.168.1.0/24.

Что касается внешнего DNS-сервера, то ограничение на IP-адреса использовать очень трудно, так как нельзя заранее сказать, какие клиенты обратятся к серверу. Зато в настройках named.conf очень легко запретить доступ тем клиентам и серверам, которые явно представляют опасность (имеется в виду, что их IP-адреса известны). Например, если известно, что кто-то пытается атаковать сервер DNS из сети 172.36.0.0/16, можно ввести в конфигурационном файле blackhole условие и запретить запросы из этой сети:

```
options {
      blackhole {
          172.36.0.0/16
      };
};
```

В секции options файла named.conf только конкретным серверам DNS (например, вторичным DNS-серверам с адресами 85.85.85.100 и 198.168.1.100) разрешается копировать информацию о зоне указанного DNS-сервера (например, первичного).

```
options {
      allow-transfer {
          192.168.10.10; 192.168.11.10;
      }; };
```

Далее мы рассмотрим вспомогательные программы и утилиты для проверки работоспособности и настройки DNS-сервера. Вместе с сервером поставляются необходимые для исследования доменного пространства утилиты dig и nslookup .

Утилита dig позволяет создавать файлы root.hints указанного типа для указанного доменного имени в формате файла зоны: # dig@(имя dns сервера) > root.hints. По умолчанию используется сервер, описанный при настройке клиентской библиотеки. Доменные имена считаются абсолютными. Список поиска опции dig:

- b — исходящий IP-адрес запроса;
- c — класс записи;
- f — имя файла (список запросов читается из файла, один запрос на строку);
- k — имя файла (файл ключей для подписи запроса и ответа TSIG);
- p — порт сервера;
- t — тип записи (по умолчанию: A, если не указан ключ — x);
- x — addr (запрос имени по IP-адресу);
- y — имя_ключа:ключ (явное задание ключа для подписи запроса и ответа TSIG).

Утилита nslookup объявлена устаревшей и навязчиво напоминает об этом при каждом запуске (к ней даже не поставляется документация, отсутствует команда «Help» и некоторые другие). Формат вызова:

```
nslookup [-ключи] доменное_имя
      [опрашиваемый_сервер]
```

Если доменное имя и имя сервера опущены или используется символ «минус» вместо доменного имени, то утилита переходит в интерактивный режим работы. Выход из интерактивного режима происходит по команде «Exit» или по нажатию «^D» (конец ввода). По команде «^C» (прерывание программы) утилита прерывает выполнение текущей операции и возвращается в интерактивный режим. Предусмотрены следующие команды (имена параметров можно сокращать):

- доменное_имя — произвести поиск записи установленного класса и типа;
- set all — показать текущие значения всех параметров;
- set [no]параметр — установить значение переключателя;
- set параметр=значение — установить значение параметра;
- type=тип-записи (по умолчанию: A для имени и PTR для адреса; можно использовать также псевдотипы AXFR и ANY);
- class=имя-класса (по умолчанию: IN);
- timeout=секунд;
- retry=число-попыток (по умолчанию: 2);
- domain=имя-локального-домена (по умолчанию берется из /etc/resolver.conf);
- server опрашиваемый_сервер — использовать указанный сервер при последующих запросах, адрес сервера определить с помощью текущего сервера;
- ls [-d] имя-зоны — передать зону целиком; -d — это синоним для -t ANY.

Значение параметра или переключателя можно установить ключом командной строки (символ «минус» перед именем параметра) или в файле настройки ~/.nslookuprc (может содержать команды «set» по одной в строке).

Напоследок необходимо заметить, что надо стараться использовать самую последнюю версию BIND, постоянно следить за выходом отчетов об обнаруженных ошибках, ставить патчи и, соответственно, своевременно обновлять версии до самых свежих.

■ ■ ■ Максим Сухомлин

ВСТРЕЧАЙТЕ НОВЫЙ ЖУРНАЛ

В ПРОДАЖЕ С 15 СЕНТЯБРЯ

VOGEL BURDA COMMUNICATIONS

...# Автоматический администратор/_.

Динамическое конфигурирование

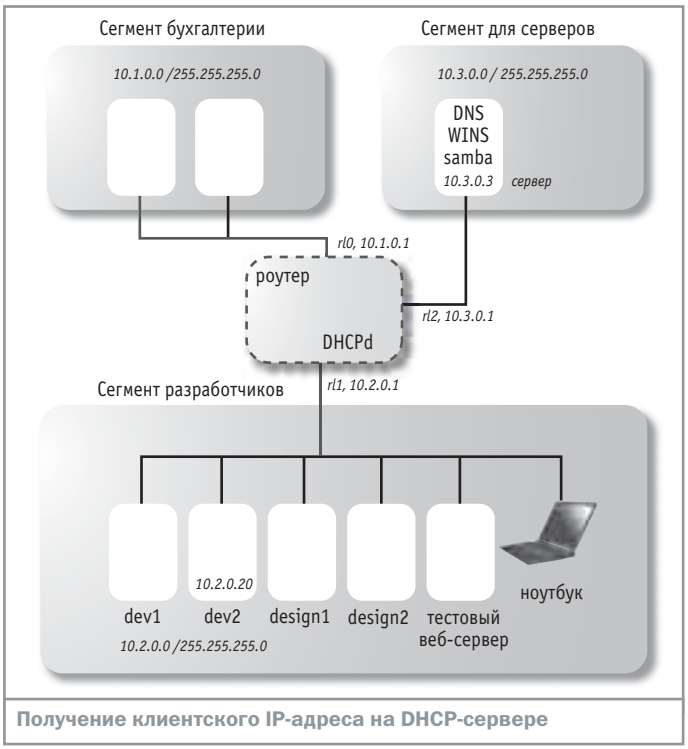
Любая компания может столкнуться с ситуацией, когда при переходе от одного провайдера к другому меняется используемый диапазон IP-адресов. Процедура его настройки однообразна и неинтересна, поэтому неудивительно, что ее попытались автоматизировать.

В настоящее время существует несколько протоколов для получения настроек сетевых сервисов. Об одном из них, самом популярном и гибком — DHCP, — и пойдет речь. Использование этой технологии уменьшает объем работ, выполняемых администратором при обслуживании большой сети, поскольку избавляет от необходимости вручную прописывать многочисленные параметры конфигурации при переустановке на клиентском компьютере операционной системы или, например, после изменения схемы адресации. При этом использование DHCP не ограничивается только выдачей настроек IP-адреса. С его помощью можно также передавать адрес прокси-сервера, настройки NetBIOS и т. д.

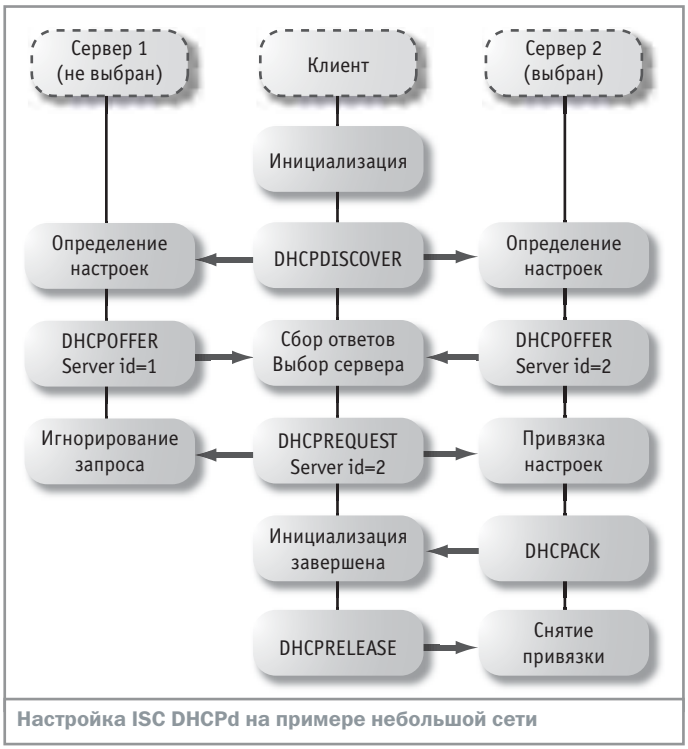
Принцип действия

Рассмотрим действие протокола на примере получения клиентом IP-адреса. Как известно, клиенты в процессе работы идентифицируются по MAC-адресу (Media Access Control address — аппаратный адрес сетевого адаптера). Для Ethernet-сетей этот адрес имеет размер 6 байт и является уникальным для каждого адаптера. Протокол использует UDP, порты 67 и 68. Получение адреса происходит следующим образом:

- Во время запуска клиент отправляет широковещательное сообщение DHCPDISCOVER.
- Доступные в сети серверы отвечают клиенту, отправляя сообщение DHCPOFFER с содержащимся внутри предлагаемым IP-адресом. »



Получение клиентского IP-адреса на DHCP-сервере



Настройка ISC DHCPd на примере небольшой сети

» ► Клиент на основе полученных предложений выбирает сервер, с которого будет производиться дальнейшее получение настроек, и шлет широковещательный пакет DHCPREQUEST с содержащимся внутри идентификатором DHCP-сервера.

► Сервер, получив пакет, узнает, что клиент отклонил его предложение (если ID сервера в пакете не совпадает с ID данного сервера), либо сохраняет на диске привязку ID клиента (как правило, это MAC-адрес) и отвечает сообщением DHCPACK, в котором содержатся все необходимые для данного клиента настройки.

► При выключении клиент посылает серверу сообщение DHCPRELEASE; после этого сервер снимает привязку выданного IP-адреса к данному клиенту.

Важно учитывать, что IP-адреса выдаются DHCP-сервером только на определенное время (lease time). При этом поддерживается два таймера, T1 и T2. Когда заканчивается первый (обычно равный примерно половине отведенного времени), клиент пытается связаться с DHCP-сервером, с которого была получена информация о настройке, и отправляет ему сообщение DHCPREQUEST. Процедура, описанная выше, повторяется, и клиент заново инициализирует T1 и T2.

Если до окончания T2 (по умолчанию это 0,875 от общего времени) от сервера не пришло сообщение с настройками, клиент вновь посылает DHCPREQUEST и после получения DHCPACK от любого другого сервера продолжает работу с новым IP-адресом (который может совпадать со старым). При отсутствии ответа от какого-либо сервера клиент регулярно, но не чаще, чем раз в минуту, продолжает посылать запрос. Когда заканчивается время, отведенное по второму таймеру, клиент обязан остановить всю свою сетевую активность и начать процедуру получения адреса заново, начиная с первого пункта.

Сервис ограничен своей подсетью, и для использования одного DHCP-сервера в нескольких сегментах необходимо включить на маршрутизаторе функцию маршрутизации DHCP/BOOTP-запросов (BOOTP relay agent). В этом случае широковещательные DHCP-запросы будут транслироваться на указанный DHCP-сервер. В маршрутизаторах от Cisco Systems это можно сделать с помощью команд ip forward-protocol udp (в глобальной конфигурации) и ip helper-address (в конфигурации интерфейса). Многие аппараты Cisco могут сами работать как DHCP-сервер (для этого используется команда ip dhcp).

При использовании маршрутизатора, построенного на Unix-подобной ОС, можно на нем самом (или на любой другой машине) в нужном сегменте сети запустить команду dhcrelay пакета ISC DHCPd (http://www.isc.org/products/DHCP). Преимущество запуска агента состоит в том, что его достаточно настроить только один раз, так как он уже находится во всех или нескольких сегментах.

От теории к практике

В заключение рассмотрим настройку ISC DHCPd на примере небольшой сети компании, занимающейся разработкой веб-сайтов.

Допустим, что есть три сегмента, соединенных между собой маршрутизатором. В одном из них находятся машины бухгалтерии, в другом — рабочие станции веб-дизайнеров и программистов, в третьем — сервер организации, на котором работает Samba и DNS. В качестве операционной системы маршрутизатора используется FreeBSD. У бухгалтеров подсеть 10.1.0.0/255.255.255.0 с адресом маршрутизатора 10.1.0.1, сконфигурированным на интерфейсе rl0; у дизайнеров и программистов — 10.2.0.0/255.255.255.0, 10.2.0.1, rl1; у сервера — 10.3.0.0/255.255.255.0, »

» 10.3.0.1, rl2; IP-адрес сервера — 10.3.0.3. Сервер один, он имеет статический IP-адрес, поэтому его подсеть DHCP-сервером обслуживаться не будет. В бухгалтерской подсети находятся только две рабочие станции. В третьей подсети находятся несколько рабочих станций программистов и дизайнеров и веб-сервер для тестирования (иногда сюда же подключается ноутбук начальника).

DHCP-сервер лучше всего запускать прямо на маршрутизаторе, поскольку он находится сразу во всех сегментах сети и это избавит от необходимости настройки dhcrelay. Для работы выберем пакет ISC DHCPd версии 3.0. В FreeBSD чаще всего используется именно этот сервер от ISC (Internet Software Consortium). Однако в стандартный дистрибутив он не входит, поэтому сначала его надо установить из коллекции портов. Путь к порту пакета —/usr /ports/net/isc-dhcp3-server. Основной файл конфигурации сервера /usr/local /etc/dhcpd.conf. При установке сервера в эту же директорию копируется образец файла конфигурации dhcpd.conf.sample. Вы можете изменить название и редактировать его или создать новый файл. Итак, начнем. По умолчанию lease time будет равным 12 часам, что позволит администратору успеть приехать в офис и все исправить, если с DHCP-сервером вдруг что-то случится.

```
default-lease-time 43200;
```

Ограничим максимальное время резервирования клиентом IP-адреса пятью днями.

```
max-lease-time 432000;
```

Все машины в сети используют WINS, который предоставляет работающая на сервере Samba; для всех используется один DNS-сервер и они находятся в домене myorg.ru:

```
option netbios-name-servers 10.3.0.3;
option domain-name-servers 10.3.0.3;
option domain-name "myorg.ru";
```

Объявим подсеть бухгалтерии:

```
subnet 10.1.0.0 netmask 255.255.255.0 {
        range 10.1.0.2 10.1.0.10;
        option routers 10.1.0.1;
}
# Объявим подсеть программистов и дизайнеров:
subnet 10.2.0.0 netmask 255.255.255.0 {
        option routers 10.2.0.1;
```

Для «гостевых» компьютеров выделим поддиапазон и уменьшим lease time:

```
pool {
          range 10.2.0.200 10.2.0.254;
          default-lease-time 300;
          max-lease-time 600;
          allow unknown clients;
        }
```

Рабочим станциям желательно иметь постоянный IP-адрес, поскольку это, например, облегчает анализ логов веб-сервера. Клиентам, не упомянутым ниже в описаниях host {}, не будут выдаваться адреса из диапазона 10.2.0.10 — 10.2.0.199.

```
        pool {
          range 10.2.0.10 10.2.0.199;
          default-lease-time 43200;
          deny unknown clients;
        }
}
group developers {
```

Клиенты будут получать свои имена хостов от DHCP-сервера; имя берется из host hostname { .. }

```
        use-host-decl-name on;
        host design1 {
```

Клиенты идентифицируются по MAC-адресу сетевого адаптера:

```
          hardware ethernet
        00:01:02:03:04:05;
        }

        host design2 {
          hardware ethernet
        00:00:00:00:00:02;
        }

        host dev1 {
          hardware ethernet
        00:00:00:00:00:03;
        }
        host dev2 {
          hardware ethernet
        00:00:00:00:00:04;
```

На хосте dev2 установлен сервис, с которым работают другие приложения, поэтому этому компьютеру нужно всегда выдавать один и тот же IP-адрес.

```
        fixed-address 10.2.0.20;
        }
}
```

■ ■ ■ Константин Стародубцев

Стандарты и организации

История происхождения

Протокол динамической настройки DHCP (Dynamic Host Configuration Protocol) был создан одной из рабочих групп IETF (Internet Engineering Task Force) — некоммерческой организации, которая определяет и разрабатывает протоколы для Интернета. За основу был взят описанный в RFC 951 (http://www.ietf.org /rfc/rfc2131.txt) протокол загрузки BOOTP (Bootstrap Protocol). С его помощью клиент во время загрузки может получить назначенный ему IP-адрес, узнать IP-адрес сервера и имя файла, который нужно загрузить и выполнить для загрузки системы. DHCP, который был описан в RFC 2131 (http://www.ietf.org /rfc/rfc951.txt), отличается от него тем, что позволяет динамически генерировать необходимую для настройки информацию, а также перераспределять уже существующие адреса прямо в процессе работы клиента и получать различные параметры с одного или нескольких DHCP-серверов.

Подписка!

Кто знает, читает CHIP SPECIAL

Цена за:
6 спецвыпусков 552 руб.
12 спецвыпусков 1104 руб.

(цена с компакт-диском)

Для оформления подписки заполните платежный документ и оплатите свой заказ через отделение Сбербанка.

При заполнении бланка разборчиво укажите фамилию, имя, отчество полностью, почтовый индекс и адрес получателя. В графе «Назначение платежа» напишите количество номеров издания. В графе «Сумма» проставьте сумму за выбранное вами количество номеров.

Адрес для писем: 125284
Москва, а/я 125

Телефоны для справок:
ЗАО «Бурда Директ» (095) 916-5706
E-mail: abo@burdadirect.ru
ЗАО «АПР» (095) 101-2537

Подписка через Интернет:
www.burdadirect.ru, www.pressa.apr.ru
Распространение и подписка в **Белоруссии:**
УП «РЭМ-Инфо», Минск, тел. (017) 291-9891/98

Извещение	ИНН 7705056238 ЗАО “Издательский дом “Бурда” р/сч № 40702810900020106298 в Сбербанке России г. Москва к/сч № 30101810400000000225 в ОПЕРУ Моск. ГТУ Банка России БИК 044525225 Плательщик Адрес Назначение платежа: ***CHIP Special _____ номеров*** — Сумма
Кассир	Подпись плательщика
Квитанция	ИНН 7705056238 ЗАО “Издательский дом “Бурда” р/сч № 40702810900020106298 в Сбербанке России г. Москва к/сч № 30101810400000000225 в ОПЕРУ Моск. ГТУ Банка России БИК 044525225 Плательщик Адрес Назначение платежа: ***CHIP Special _____ номеров*** — Сумма
Кассир	Подпись плательщика

Калифорнийский стрелочник/_...

Общий выход в Интернет

Доступ в Интернет сегодня стремительно дешевеет. Все чаще на различных форумах спрашивают: «Как сделать так, чтобы несколько компьютеров у меня дома могли работать в Интернете через одно подключение?» Если же говорить о нормальном сервере крупной локальной сети, этот вопрос становится едва ли не первоочередным.

Для решения поставленной задачи используется NAT — Network Address Translation. Этой функцией должен обладать маршрутизатор, включаемый между вашей внутренней сетью и Интернетом. Маршрутизатор — это специализированный компьютер, основным назначением которого является дальнейшая передача приходящих на него IP-пакетов в нужном направлении. Направление выбирается исходя из набора правил, который называется «таблицей маршрутизации». Различают «статическую» и «динамическую» маршрутизацию: в первом случае записи в таблице меняются только администратором системы, а во втором изменения вносятся автоматически, согласно информации, пришедшей от других маршрутизаторов.

Любая работа в сети — будь то просмотр веб-страниц, разговор по ICQ или чтение почты — представляет собой обмен IP-пакетами между компьютером клиента и сервером в Интернете. Каждый IP-пакет состоит из заголовка и поля данных. В заголовке есть два ключевых поля: адрес отправителя и адрес получателя, SRC IP и DST IP. NAT исправляет в проходящем через маршрутизатор пакете поле SRC IP так, чтобы вместо адреса клиентского компьютера в этом поле значился IP-адрес самого маршрутизатора, запоминая при этом все параметры исходного пакета. Таким образом, когда от сервера в Интернете придет ответный пакет, маршрутизатор сможет произвести обратную замену и доставить этот пакет клиентскому компьютеру. Подробнее о NAT можно прочитать в руководстве или в man-страницах, набрав команду man natd.

FreeBSD представляет собой, пожалуй, идеальный программный маршрутизатор. Автор долгое время использовал Linux, но после серии экспериментов обнаружил, что FreeBSD в »

» данной ситуации устанавливается и настраивается быстрее и проще, занимает меньше места, а работает практически так же. При работе в качестве маршрутизатора сети основной недостаток этой системы — менее гибкий NAT, чем в Linux. Но в большинстве случаев необходимости тонкого конфигурирования трансляции сетевых адресов не возникает.

Далее мы рассмотрим процесс настройки ОС для выполнения функций исключительно маршрутизатора. Если вы хотите, чтобы эта же машина отвечала за почту и другие функции, достаточно просто не вносить соответствующих упоминаемых изменений в конфигурационные файлы.

В глубины системы

Поскольку маршрутизация и NAT являются стандартными компонентами системы, все их настройки вынесены в системные конфигурационные файлы. Все корректировки производятся в файле /etc/rc.conf (в нем переопределяются настройки системы по умолчанию, описанные в /etc/defaults /rc.conf). Вот примерное содержание этого файла после установки системы:

```
# -- sysinstall generated deltas -- # Tue
      May 27 20:27:58 2004
# Created: Tue May 27 20:27:58 2004
# Enable network daemons for user con-
      venience.
# Please make all changes to this file, not
      to /etc/defaults/rc.conf.
# This file now contains just the overrides
      from /etc/defaults/rc.conf.
font8x14="cp866-8x14"
font8x16="cp866b-8x16"
font8x8="cp866-8x8"
kern_securelevel_enable="NO"
keymap="ru.koi8-r"
keyrate="fast"
mousechar_start="3"
nfs_reserved_port_only="YES"
saver="daemon"
scrnmap="koi8-r2cp866"
sendmail_enable="YES"
sshd_enable="YES"
```

Прежде всего отключите автоматический запуск почтового сервера sendmail — его необходимость на маршрутизаторе крайне сомнительна. Для этого в параметре sendmail_enable нужно изменить значение «YES» на «NO». После перезагрузки sendmail все-таки будет запущен, но только на внутреннем сетевом интерфейсе lo0 с адресом 127.0.0.1 — это нужно для того, чтобы операционная система могла присылать администратору отчеты о своей работе, которые автоматически создаются с заданной периодичностью.

Теперь нужно вписать IP-адреса для обеих сетевых плат, установленных в компьютере. В операционной системе FreeBSD, в отличие от Linux, название сетевого интерфейса зависит от изготовителя сетевой платы. В нашем случае это широко распространенные карты Realtek. Ядро операционной системы определяет такие карточки как устройства Rl0, Rl1 и так далее.

Устанавливаем на одной карте IP-адрес и маску подсети, выданную провайдером, а также указываем шлюз по умолчанию (12.34.45.1). На другой карте устанавливаем приватный адрес (IP-адрес из диапазона, специально выделенного для использования в локальных сетях) 192.168.232.1 с маской подсети 255.255.255.0. Делается это посредством добавления с помощью текстового редактора в /etc/rc.conf следующих строк:

```
ifconfig_rl0="inet 12.34.45.56 netmask
      255.255.255.0"
ifconfig_rl1="inet 192.168.232.1 netmask
      255.255.255.0"
defaultrouter="12.34.45.1"
```

Затем указываем, на каком интерфейсе нам нужно выполнять трансляцию адресов (обычно это внешний интерфейс), и включаем автоматический запуск NAT с нужными нам опциями:

```
natd_enable="YES"
natd_interface="rl0"
natd_flags="-u -s -m"
```

Здесь следует сделать небольшое отступление и рассказать о том, как вообще во FreeBSD организуется NAT. В отличие от Linux, где трансляция сетевых адресов выполняется непосредственно ядром, во FreeBSD она выполняется в пользовательском программном пространстве с помощью специального демона natd. Строчка natd_enable как раз и включает запуск natd при загрузке системы и передает ему параметрами имя интерфейса из natd_interface и ключи из natd_flags.

Для взаимодействия между ядром операционной системы и программой natd существует специальный тип соединений, называемых Divert Sockets. Принцип работы прост: брандмауэр, который по умолчанию установлен в системе, любой пришедший на маршрутизатор пакет пересылает в Divert Socket, а natd из него читает пакеты, обрабатывает и отправляет назад.

К сожалению, по неизвестной причине в ядре FreeBSD, которое устанавливается по умолчанию, этот тип соединений отключен. Для его активации необходима пересборка ядра операционной системы (о том, как это делается, вы можете прочитать в другой статье этого номера). Коротко говоря, это можно сделать следующим образом (все команды выполняются пользователем root):

```
cd /usr/src/sys/i386/conf
cp GENERIC MYKERN
echo “options IPDIVERT” >> MYKERN
cd ../../compile/MYKERN
make depend all install
reboot
```

Теперь у вас в операционной системе есть поддержка всего необходимого и можно заняться настройкой брандмауэра. »

Плюсы и минусы FreeBSD

Надежность и простота

+ Низкие требования к аппаратному обеспечению — для нормальной работы достаточно центрального процессора уровня Pentium 166, 64 Мбайт ОЗУ и 1 Гбайт свободного пространства на жестком диске.
+ Простота настройки.
+ Надежность работы, особенно в условиях большой нагрузки.
– Неполная поддержка самого нового аппаратного обеспечения или ее отсутствие.

```
router# netstat -rn
Routing tables
Internet:
Destination        Gateway            Flags     Refs      Use      Netif Expire
default            12.34.45.1         UGSc      4       7221547    rl0
12.34.45.0         ff:ff:ff:ff:ff:ff  UHLWb     1         95903    rl0     =>
12.34.45.0/24      link#1             UC        2             0    rl0
127.0.0.1          127.0.0.1          UH        1         36013    lo0
192.168.232        link#2             UC        1             0    rl1
router#
```

Таблица 1. Результат выполнения команды netstat

» Для случая простого маршрутизатора никаких особенных настроек не требуется — достаточно сделать его полностью открытым, добавив в /etc/rc.conf следующие две строки:

```
firewall_enable="YES"
firewall_type="open"
```

Осталось лишь вписать в конфигурационный файл имя сервера и включить маршрутизацию:

```
hostname="router"
gateway_enable="YES"
```

Теперь можно, записав файл, перезагрузить машину. После перезагрузки маршрутизатор полностью готов к работе, что можно проверить с помощью программы ping:

```
router# ping 12.34.45.1
PING 12.34.45.1 (12.34.45.1): 56 data
        bytes
64 bytes from 12.34.45.1: icmp_seq=0
        ttl=58 time=42.992 ms
64 bytes from 12.34.45.1: icmp_seq=1
        ttl=58 time=43.173 ms
64 bytes from 12.34.45.1: icmp_seq=2
        ttl=58 time=44.291 ms
--- 12.34.45.1 ping statistics ---
3 packets transmitted, 3 packets received,
        0% packet loss
round-trip min/avg/max/stddev =
        42.992/43.485/44.291/0.574 ms
router#
```

Как видите, пакеты ходят. Но если проблемы все-таки возникают, их диагностику надо начинать с просмотра состояния сетевых интерфейсов. Делается это с помощью команды ifconfig. Вот примерный результат ее работы:

```
router# ifconfig -a
rl0:
        flags=8843<UP,BROADCAST,RUN-
        NING,SIMPLEX,MULTICAST> mtu 1500
  inet 12.34.45.56 netmask 0xffffff00
        broadcast 12.34.45.255
  ether 00:40:f4:60:9e:c2
  media: Ethernet autoselect (100baseTX
        <full-duplex>)
  status: active
rl0:
        flags=8843<UP,BROADCAST,RUN-
        NING,SIMPLEX,MULTICAST> mtu 1500
  inet 192.168.232.1 netmask 0xffffff00
        broadcast 192.168.232.255
  ether 00:40:f4:60:9e:c3
  media: Ethernet autoselect (100baseTX
        <full-duplex>)
  status: active
lo0:
        flags=8049<UP,LOOPBACK,RUNNING,M
        ULTICAST> mtu 16384
  inet 127.0.0.1 netmask 0xff000000
router#
```

Из приведенного примера видно, что у нашего маршрутизатора есть три сетевых интерфейса (rl0, rl1 и служебный lo0). Кроме того, мы видим, какие IP-адреса назначены каждому из интерфейсов, и состояние соединения на каждой из сетевых карточек. В случае проблем с кабелем в графе status будет написано no link. Полезно будет также посмотреть таблицу маршрутизации с помощью команды netstat, результаты работы которой приведены в таблице 1.

Теперь можно настраивать клиентские компьютеры (подразумевается, что они будут работать под одной из ОС семейства Microsoft Windows). Для этого достаточно прописать в настройках сети адреса из подсети 192.168.232.2-192.168.232.254 с маской 255.255.255.0 и указать в качестве шлюза по умолчанию 192.168.232.1. Впишите тот DNS-сервер, который указан провайдером. После выполнения всех необходимых операций на клиентских компьютерах доступ в Интернет должен работать. Проверить это можно с помощью команды ping:

```
C:\>ping 12.34.45.1
Обмен пакетами с 12.34.45.1 по 32 байт:
Ответ от 12.34.45.1: число байт=32 вре-
        мя=16мс TTL=55
Ответ от 12.34.45.1: число байт=32 вре-
        мя<10мс TTL=55
Ответ от 12.34.45.1: число байт=32 вре-
        мя<10мс TTL=55
Ответ от 12.34.45.1: число байт=32 вре-
        мя<10мс TTL=55
Статистика Ping для 12.34.45.1:
Пакетов: отправлено = 4, получено = 4,
        потеряно = 0 (0% потерь),
Приблизительное время передачи и при-
        ема: наименьшее = 0 мс, наиболь-
        шее = 16 мс, среднее = 4 мс
```

Выход открыт

Как вы, наверное, убедились, настроить функции маршрутизации во FreeBSD гораздо проще, чем, например, в Windows Server 2003. Конечно, как мы уже говорили, по сравнению с Linux данная служба несовершенна. Однако, учитывая другие плюсы FreeBSD, можно сказать, что у вас вряд ли возникнут какие-нибудь проблемы (разве что вы поставите перед собой задачу создания какой-нибудь сверхсложной сети, в которой будут присутствовать машины с совершенно разными операционными системами). Ну, а если трудности все-таки возникнут, на помощь вам всегда придут файлы помощи. Помните, что обращаться к ним не зазорно даже опытному администратору. ■ ■ ■ Антон Ногинов

...# Серверная нирвана/_...

Настройка основных сервисов

Различные серверы, которые обслуживают веб-сайты, обеспечивают передачу файлов по протоколу FTP, помогают людям обмениваться текстовыми сообщениями и т. д., как правило, работают под управлением Unix-систем. И когда требуется обеспечить комплексные требования для серверной ОС, выбор зачастую падает именно на FreeBSD.

Не всегда возможно организовать множество серверов, выделяя под каждую конкретную задачу по отдельной машине. FreeBSD позволяет реализовать несколько серверов на одном ПК. Главное в этом случае — помнить, что такая машина должна иметь соответствующий уровень производительности, несмотря на низкие требования этой ОС к аппаратному обеспечению. В данной статье мы рассмотрим организацию отдельных серверов, но поговорим и о том, как организовать работу всех этих служб на одной машине. Начнем с того, как настраивается WWW-сервер.

Платформа для сайтов

Существующие для Unix-платформ веб-серверы заметно различаются по функциональности. Одни из них представляют собой небольшие, но быстродействующие демоны, другие — целые программные пакеты, масштабируемые при помощи модулей от сторонних разработчиков. Когда речь заходит о выборе конкретной программы, даже новички в области сайтостроения обычно сразу вспоминают Apache Web Server (http://httpd.apache.org). История развития этого веб-сервера начинается с 1995 года. Его разработчики постоянно обменивались обновлениями и «заплатками» (patches), поэтому сервер назывался A Patchy Server. Позднее это название трансформировалось в современное «воинственное» Apache. На сегодня это один из самых распространенных, гибких и надежных веб-серверов. Он доступен как на Unix-подобных, так и на Windows-платформах, поддерживает всевозможные PHP- и CGI-скрипты. Все это позволяет создавать кроссплатформенные активные веб-приложения. Существуют две ветви данного сервера: серии 1.3 (последняя стабильная версия 1.3.31) и серии 2.0 (последняя версия 2.0.48).

В данной статье мы будем рассматривать Apache 1.3. Хотя синтаксис и процесс уста- »

» новки для версий 2.0.x немного отличаются, в основном они схожи с версиями 1.3.x.

Минимальная конфигурация компьютера для работы сервера с настройками по умолчанию и подключенными модулями: процессор Pentium с частотой 166 МГц, 64 Мбайт оперативной памяти и 50 Мбайт свободного места на жестком диске.

Существует два типа установки Apache. Первый — установка с DSO (Dynamic Shared Objects). В этом случае подключаемые модули по мере необходимости загружаются в память, выделенную под запускаемый сервер. С одной стороны это позволяет избежать ситуации, когда в нужный момент не окажется необходимого модуля, с другой стороны — отнимает много машинных ресурсов, что при большой нагрузке на сервер может даже привести к зависанию. Второй, более производительный способ установки, представляет собой сборку Apache со статическими модулями, то есть с их необходимым набором, подходящим для поставленных задач. Такой способ сложнее для установки и конфигурации, но именно он, вкупе с chroot- и jail-конфигурациями, чаще всего используется хостинговыми компаниями. При установке из коллекции портов Apache компилируется с DSO следующим образом:

```
# cd /usr/ports/www/apache13
# make all install clean
```

Введя в директории /usr/ports/www команду ls, можно увидеть большое количество дополнительных модулей для Apache. Например, если вы хотите защитить контент сайта с помощью алгоритма шифрования SSL, вам понадобится модуль mod_ssl:

```
# cd /usr/ports/www/mod_ssl
# make all install
```

Аналогично, для поддержки PHP3 служит модуль mod_php3 и т. д. Конфигуратор автоматически внесет необходимые изменения в конфигурационный файл Apache httpd.conf, расположенный по умолчанию в директории /usr/local/etc/apache. Если вам необходимо самостоятельно добавить в конфигурацию нужные модули или удалить их, в httpd.conf надо добавить строки следующего вида:

```
AddModule ../modules/имя_вашего_модуля
```

При самостоятельной сборке из исходников Apache по умолчанию собирается без DSO. Для активации модулей необходимо задать параметры компиляции:

```
--activate_module=путь_где_лежит_модуль
```

Но до этого нужно предварительно скомпилировать и установить необходимые модули. Например, для внедрения SSL в исходники Apache вам потребуется уже упомянутый пакет mod_ssl (ftp://ftp.cronyx.ru/pub/mirror/modssl/source), а также пакет openssl-engine-0.9.7b который можно скачать с сайта http://www.openssl.org. Распаковываем и устанавливаем последний:

```
# tar -zxvf openssl-engine-0.9.7b
# sh config no-idea no-threads -fPIC
# make
# make test
```

Теперь установим mod_ssl:

```
#./configure --with-apache=../
      apache_1.3.31 --with-ssl=../
      openssl-engine-0.9.7b --
      prefix=/usr/local/apache
# make
# make install
```

Для динамических сайтов также часто требуется PHP. Сконфигурируем и установим модуль mod_php:

```
# ./configure --with-apache=../
      apache_1.3.31 --prefix=/
      usr/local/Apache
# make
# make install
```

Не забудьте установить и сам модуль PHP:

```
# tar -zxvf php-4.3.0
# ./configure --prefix=/usr --with-
      apache=../apache_1.3.31
# make
# make install
```

Наконец, устанавливаем сам Apache с поддержкой модулей. Конфигурационная строка будет выглядеть так:

```
# ./configure --prefix=/usr/local/apache --
      bindir=/usr/bin --sbindir=/usr/sbin --
      sysconfdir=/usr/local/apache/conf --
      logfiledir=/var/log/apache --datadir=/
      usr/local/apache/data --activate-
      module=src/modules/ssl/libssl.a --
      enable-module=ssl --activate-
      module=src/modules/php4/
      libphp4.a --enable-module=php4
```

Все необходимые изменения будут автоматически внесены в httpd.conf. Для поддержки DSO при установке сервера надо указать параметр --enable-shared=all для установки всех модулей, или --enable-shared=max для установки максимального количества основных расширений. Если необходимо подключить или отключить конкретный модуль, используйте параметр --disable(enable)-module=название_модуля.

Вернемся к файлу конфигурации и проведем предварительную настройку сервера. В частности, для работы программ, написанных на PHP4 в httpd.conf, надо добавить строки:

```
AddType application/x-httpd-php.php
AddType application/x-httpd-php-
      source.phps
```

Отредактируем строку DirectoryIndex, чтобы дать понять серверу, что файлы ти- »

Веб-сервер Xitami

Достойный выбор

Еще один популярный масштабируемый веб-сервер — Xitami (http://www.xitami.com). В отличие от Apache, он обслуживает все соединения (кроме CGI) в одном процессе, что уменьшает количество требуемой памяти и нагрузку на процессор. Xitami позволяет работать по протоколам FTP, CGI/1.1 и SSI, поддерживает определяемые пользователем MIME-типы, конфигурирование сервера «на лету» и многое другое. Фактически сразу после установки сервер готов к работе.

» па index.php надо обрабатывать как файлы по умолчанию:

```
DirectoryIndex index.html index.php
```

Для удобства администрирования вместе с Apache устанавливается утилита apachectl, имеющая следующие параметры:

► apachectl start | stop | restart — старт сервера , остановка и перезапуск;

► apachectl configtest — проверка файла конфигурации на ошибки;

► apachectl startssl — запуск Apache с SSL (при условии что Apache собран с поддержкой OpenSSL).

С Apache также поставляются еще несколько полезных утилит: httpd-l показывает список загруженных модулей. http_load — тест сервера на производительность. Ниже приведен пример использования последней команды:

```
http_load -rate 10 -seconds 300 urls
```

Значение rate может меняться от 1 до 10. В нашем примере тест проводится с максимально возможным количеством запросов. Время теста — 300 секунд (что вполне достаточно). Файл urls содержит адрес сервера по протоколам HTTP и HTTPS.

Дальнейшее конфигурирование сервера достаточно хорошо освещено в печатных и электронных источниках. Заметим только, что конфигурационный файл Apache снабжен многочисленными комментариями, поэтому настроить сервер, обслуживающий простую веб-страничку, труда не составит.

FTP

Встроенные средства

Стандартный, входящий в базовый дистрибутив FreeBSD FTP-сервер — ftpd.

Поскольку данный сервис инсталлируется вместе с системой, опустим процедуру его установки. Для автоматического запуска сервера достаточно раскомментировать строку в файле /etc/inetd.conf:

```
ftpd nowait 400 stream tcp /usr/sbin/
    tcpd in.ftpd
```

Также можно запускать сервер из командной строки в режиме Standalone. Для этого строка ftpd в файле inetd.conf должна быть закомментирована. Итак, запускаем сервер:

```
# /usr/libexec/ftpd -D (параметр D
    сообщает серверу, что он
    запускается в режиме демона)
```

Далее проверяем статус сервера:

```
# netstat -na | grep LISTEN

mydomain.ru 21 LISTEN 'результат
    выполнения команды
```

Сервер запущен и готов к работе. Файлы по умолчанию располагаются в директории /var/ftp/pub. Для желающих сконфигурировать службу более тонко напомним, что все конфигурационные файлы расположены в директории /etc/default/ftpd/.

Выбор профессионала

Несмотря на наличие встроенного средства в ОС, практически безальтернативным выбором для FTP-сервера на платформе FreeBSD выступает программа Proftpd. О ее надежности и производительности говорит уже то, что она используется на сайте sourceforge.net — хостинге огромного числа opensource-проектов. Этот сервер отличается от других большей безопасностью (в том числе по сравнению с встроенным по умолчанию ftpd, который мало подходит для создания мощного FTP-сервера), а также более гибкой настройкой, достойной стабильностью, кроссплатформенностью и поддержкой большого числа различных расширений — например, для работы с MySQL.

Как и в случае с любой другой службой или программой, установку данного клиента будет лучше произвести из портов. Делается это следующим образом:

```
# cd /usr/ports/ftp/proftpd/
# make all install
# make clean
```

При отсутствии подключения к глобальной сети можно установить Proftpd из пакетов. При этом вы можете сразу собрать необходимые модули, добавив опцию:

```
# make --with-modules=${MODULES}
```

Этот FTP-сервер может запускаться как из /etc/inetd.conf, так и с помощью специального скрипта. В случае с упомянутым файлом /etc/inetd.conf надо сделать следующее:

```
# ftp stream tcp nowait root
    /usr/local/libexec/proftpd proftpd
# ftp stream tcp nowait root
    /usr/libexec/ftpd ftpd -l
# ftp stream tcp6 nowait root /usr/libex-
    ec/ftpd ftpd -l
```

»

FTP-сервер vsftpd

Маленький, но мощный

Отдельного упоминания заслуживает очень хороший FTP-сервер под названием vsftpd (http://vsftpd.beasts.org), который распространяется по лицензии GPL и, соответственно, существует для всех Unix-подобных операционных систем, а также для Linux. Кроме стандартных функций, он наделен обширными современными возможностями, которыми могут похвастать далеко не все аналогичные серверы. Вот некоторые из них:

► поддержка виртуальных пользователей и IP-конфигураций;

► поддержка IPv6;

► возможность шифрации посредством алгоритма SSL;

► тонкая настройка и конфигурация пропускной способности, запретов и разрешений в привязке к конкретному IP и т. п.

На сегодняшний день под управлением этого сервера работает огромное число ресурсов в Интернете, в том числе и очень мощных и известных. Если для вас важны надежность, безопасность и возможность тонкого конфигурирования, vsftpd станет хорошим выбором.

» Таким образом, демон будет загружаться сразу при старте системы.

Что касается второго варианта со скриптом, то в /usr/local/etc/rc.d/proftpd.sh.sample существует файл-пример, который необходимо переименовать в proftpd.sh и присвоить права на запуск chmod 755 proftpd.sh.

В соответствии с целями

Теперь пришла пора настроить FTP-сервер под наши конкретные нужды. Для этого необходимо создать (если по каким-то причинам это не было сделано ранее) файл конфигурации proftpd, а затем отредактировать его следующим образом в текстовом редакторе:

```
# cp /usr/local/etc/proftpd.conf.default
/usr/local/etc/proftpd.conf
# vi /usr/local/etc/proftpd.conf
```

Итак, давайте по порядку рассмотрим различные опции и возможности Proftpd, а также их настройку:

```
# Имя сервера
ServerName "Corporate FTP Server"

#Запуск демона (мы рассмотрим случай
    с использованием inetd)
ServerType inetd
DefaultServer on
ServerIdent off
```

Если сервер находится в локальной сети, зачастую требуется предоставить root-пользователю возможность соединяться по протоколу FTP в целях администрирования.

```
RootLogin on

# Далее идет стандартный FTP-порт
Port 21

# Маска для ограничения создания
    директорий и файлов
Umask 022

# Пользователь и группа,
    под которой работает демон
User admin
Group nogroup
```

Введем необходимые ограничения и соответствующие предупреждения:

```
MaxClients 50 "Лимит количества
    соединений с сервером достигнут"
MaxClientsPerHost 5 "%m клиента
    уже подключены с вашего хоста,
    больше не разрешено"
MaxLoginAttempts 10
    "Слишком много попыток войти"
```

Ограничение трафика также будет весьма полезно. В данном примере мы ограничиваем полосу пропускания в обе стороны на уровне 256 Кбит/с для всех пользователей:

```
TransferRate RETR,STOR,APPE 256
```

При входе на FTP-сервер можно осуществить вывод всевозможных сообщений. Вот небольшой пример:

```
DisplayConnect /etc/ftp_connect.msg
DisplayLogin /etc/ftp_login.msg
AccessDenyMsg "Доступ к серверу
    в данный момент невозможен"
AccessGrantMsg "Теперь вы можете
    скачивать/закачивать файлы"
DisplayGoAway "В доступе отказано"
```

Также можно задать IP-адреса, вход с которых разрешен или запрещен:

```
#UseHostsAllowFile /etc/proftpd.allow
#UseHostsDenyFile /etc/proftpd.deny
```

Различные временные ограничения на проведение ряда операций устанавливаются путем добавления следующих строк (время в данном случае указывается в секундах):

```
TimeoutIdle 60
TimeoutLogin 60
TimeoutNoTransfer 360
TimeoutStalled 720
```

Можно задать пользователям с различными привилегиями возможность доступа к отдельным папкам. Например, для root:

```
DefaultRoot / wheel
```

По умолчанию сервер не позволяет подключаться анонимному пользователю. Чтобы снять это ограничение, надо добавить следующую строку:

```
DefaultRoot путь_до_папки users
```

Задание «домашней» пользовательской директории осуществляется следующей несложной строкой:

```
DefaultRoot пользовательская_
    директория пользователь
```

Не стоит забывать и о таком важном обстоятельстве, как ведение необходимых лог-файлов. Впрочем, для начала мы создадим необходимые файлы и директории следующими командами:

```
# cat > /var/log/proftpd-error.log
# cat > /var/log/proftpd-tranfer.log
# mkdir /var/run/proftpd/ && cat
    /var/run/proftpd/proftpd.scoreboard
```

Теперь вернемся к файлу конфигурации и добавим туда следующие строки:

```
SyslogLevel notice
UseReverseDNS off
TransferLog /var/log/proftpd-tranfer.log
SystemLog /var/log/proftpd-error.log
```

Рассмотрим часто возникающую задачу — обеспечение доступа анонимных пользователей на FTP-сервер. Для ее решения необходимо добавить в конфигурационный файл следующий раздел:

```
<Anonymous /путь_к_директории_
    для_анонимных_пользователей>
User anonftp
Group nogroup
UserAlias anonymous anonftp
MaxClients 50 "Лимит в %m пользователей
    достигнут, попробуйте зайти позже"
DisplayFirstChdir .message
    <Limit WRITE>
    DenyAll
    </Limit>
</Anonymous>
```

»

» Следите за ошибками

Вы можете и дальше настраивать свой сервер под специфические задачи, которые на него возлагаются. Мы привели только самые основные сведения и настройки, которые пригодятся в любом случае. Не забывайте после ввода в эксплуатацию периодически проверять состояние вашего FTP-сервера, анализируя лог-файлы:

```
# tail -f /var/log/proftpd-error.log
```

Местный почтамт

Под FreeBSD существует большое число почтовых серверов, но начнем мы с встроенного в ОС Sendmail. Для большинства типовых конфигураций возможностей этого сервера будет вполне достаточно.

На весь домен

Вы можете отправлять почту любому внешнему адресату, если у вас запущен свой DNS-сервер или правильно составлен файл /etc/resolv.conf. Теперь нужно настроить Sendmail так, чтобы пользователи вашего хоста получали свою почту. Очевидно, что для этого ваш сервер должен иметь статический IP-адрес, а установленный брандмауэр пропускать SMTP-пакеты. Не лишним будет свериться и со службой DNS: запись MX должна соответствовать IP-адресу вашего хоста. Предположим, что почтовый сервер будет постоянно подключен к Интернету. Тогда его MX-запись может выглядеть так:

```
myhost.ru IN MX 10 mail.myhost.ru
```

Теперь пришло время установить Sendmail. Свежую версию можно загрузить с ftp://ftp.sendmail.org/pub/sendmail/. Установка Sendmail ничем не отличается от других программ, на ней мы останавливаться не будем. Перейдем к конфигурированию почтового сервера. Ниже приведена информация об основных конфигурационных файлах Sendmail:

► /etc/mail/access — файл, определяющий, какие хосты имеют доступ к локальному почтовому серверу, и тип предоставляемого доступа.

```
213.33 RELAY
```

Например, при такой записи будет разрешена пересылка почты от соответствующего диапазона хостов.

► /etc/mail/aliases — база данных с синонимами почтовых ящиков. В примере ниже почта на адрес general рассылается на три локальных почтовых ящика.

```
general: ivan, petr, masha
```

► /etc/mail/local-host-names — список локальных доменов и хостов, для которых Sendmail принимает почту.

► /etc/mail/mailer.conf — список вспомогательных почтовых программ, вызывающих Sendmail через оболочку mailwrapper.

► /etc/mail/mailertable — таблица почтовых доменов, на которые надо пересылать корреспонденцию. Полезна, например, в том случае, когда у компании есть несколько филиалов — каждый со своим почтовым доменом.

► /etc/mail/sendmail.cf — основной файл настройки почтового сервера, управляющий общим поведением Sendmail. После внесения в этот файл изменений почтовый сервер необходимо перезапускать.

► /etc/mail/virtusertable — таблицы сопоставлений виртуальных пользователей и доменов с реальными почтовыми ящиками.

Sendmail — очень гибкая в конфигурировании программа, содержащая бесчисленное количество опций. Подробное описание всех файлов и примеры их содержимого можно найти в объемной справке /usr/src/contrib/sendmail/cf/README (привести их здесь не позволяет объем данной статьи).

Больше почтовых отделений

Иногда возникает ситуация, когда надо поддерживать несколько виртуальных почтовых серверов: например, если у вас несколько доменов, но надо, чтобы почта приходила только на один хост. Предположим, у вас есть домен office1.ru, а ваш хост называется mail.myhost.ru. В этом случае DNS надо настраивать следующим образом:

```
office1.ru    MX    10    mail.myhost.ru
```

При этом, если возникнет необходимость каким-либо образом обратиться к хосту office1.ru, у вас вряд ли что-либо получится без A-записи для него.

Теперь осталось только сообщить программе Sendmail, для каких доменов или хостов она должна принимать почту. Есть несколько способов сделать это:

► добавить названия этих хостов в файл /etc/sendmail.cw, если вы используете Feature(use_cw_file). Для Sendmail версии 8.10 или выше удобнее отредактировать файл /etc/mail/local-host-names.

► добавить строку Cwyour.host.com в файл /etc/sendmail.cf или /etc/mail/sendmail.cf (в том случае, если версия выше 8.10).

Впрочем, далеко не все системные администраторы любят пользоваться Sendmail, потому что она считается весьма уязвимой для удаленных атак. Отключить Sendmail очень легко: для этого достаточно всего лишь внести следующие коррективы в файл /etc/rc.conf:

```
sendmail_enable="NONE"
```

»

Настройка виртуального хостинга

Каждому – по псевдониму

Часто один сервер обслуживает несколько сайтов. В таком случае для одного физического интерфейса назначается несколько сетевых адресов. Этим сетевым адресам присваиваются псевдонимы (Alias). Удобнее всего добавлять описания псевдонимов в файл /etc/rc.conf. В общем случае alias выглядит так:

```
ifconfig_fxp0_alias0="inet xxx.xxx.xxx.
xxx netmask xxx.xxx.xxx.xxx"
```

Можно создавать любое количество псевдонимов, но первый из них должен быть указан как alias0, и далее определения должны следовать по порядку.

» Давайте теперь познакомимся с альтернативными популярными почтовыми серверами.

Модульный Qmail

Альтернативой популярному Sendmail является Qmail. Он несколько проще в настройке, безопаснее и также обладает неплохой производительностью. На компьютере с процессором класса Pentium Qmail легко обрабатывает свыше 200 тыс. отдельных почтовых сообщений в день. Скачать и самостоятельно установить сервер из исходных кодов можно, например, с сайта http://cr.yp.to/software. Файл последней версии программы — qmail-1.03.tar.gz. Но самый простой способ установки Qmail — установка из коллекции портов.

```
# cd /usr/ports/mail/qmail-tls
# make all install -DWITH_BIG_TODO_PATCH
```

При этом загрузится и откомпилируется версия почтового сервера Qmail с поддержкой SSL- или TLS-шифрования.

Если вы намереваетесь в дальнейшем защитить почтовую систему от вирусов и спама с помощью GPL-программ Clam Antivirus и SpamAssassin, то Qmail нужно компилировать с опцией with_qmailqueue_patch.

```
# make certificate
# cp work/servercert.pem /var/qmail/
      control/servercert.pem
# chmod 640
      /var/qmail/control/servercert.pem
# chown qmaild:qmail
      /var/qmail/control/servercert.pem
# make clean
# cp /etc/rc.conf /etc/rc.conf.bak
# grep -v sendmail_enable /etc/rc.conf >
      /etc/rc.conf2
# echo 'sendmail_enable="NONE"' >>
      /etc/rc.conf2
# mv /etc/rc.conf2 /etc/rc.conf
# rm /usr/sbin/sendmail
# cp /var/qmail/bin/sendmail
      /usr/sbin/Sendmail
```

В приведенной выше последовательности команд мы получили сертификат для Qmail и произвели некоторые настройки системного окружения для удаления Sendmail. Установим теперь пакет ucpi-tcp, позволяющий использовать Qmail без суперсервера inetd.

```
# cd /usr/ports/sysutils/ucspi-tcp
# make extract
# cd work/
# fetch http://www.qmail.org/ucspi-rss.diff
# patch <ucspi-rss.diff
# rm ucspi-rss.diff
# cd ../
# make install clean
```

Хорошим решением будет установить vpopmail — инструмент для администрирования виртуальных доменов и аккаунтов Qmail, а также ezmlm-idx — менеджер мейл-листов.

```
# cd /usr/ports/mail/vpopmail
# make all install clean
# cd /usr/ports/mail/ezmlm-idx
# make all install clean
# cp /usr/local/etc/ezmlm/ezmlmrc.sam-
      ple /usr/local/etc/ezmlm/ezmlmrc
```

Для администрирования аккаунтов почтового сервера установим Qmailadmin. С его помощью можно создавать почтовые ящики, мейл-листы и осуществлять многие другие операции через веб-интерфейс.

```
# cd /usr/ports/mail/qmailadmin
# make extract
# cd work/qmailadmin-1.2.0
# ./configure --enable-cgibindir=/
      path/to/your/cgi-bin --enable-
      htmldir=/path/to/your/html/directory
      --enable-autoresponder-path=/usr/
      local/bin/qmail-autoresponder
# make && make install-strip
```

Естественно, значения опций указывающих на директории cgi-bin и html, должны соответствовать настройками вашего веб-сервера. Qmailadmin установлен. Теперь, чтобы попасть в администраторский интерфейс, достаточно набрать в браузере http://www.yourhost.com/cgi-bin/qmailadmin.

Помимо упомянутых выше программ, полезно будет установить также daemontools (http://cr.yp.to/daemontools.html) — отличный пакет утилит для обслуживания сервисов. CHIP

Безопасность

Почта без спама и вирусов

Самый известный в России производитель антивирусных программ для рабочих станций, «Лаборатория Касперского» уделяет внимание и информационной безопасности серверных систем. «Антивирус Касперского» для FreeBSD/OpenBSD Mail Servers способен интегрироваться в почтовые системы Sendmail, Qmail, Postfix и Exim. В режиме реального времени программа проверят не только прикрепленные к письмам файлы, но и тело письма, и внедренные в него OLE-объекты. Вся обнаруженная подозрительная или инфицированная корреспонденция может быть отправлена в «карантин». При этом администратору отправляется сообщение с описанием вредоносного кода, адресами отправителя и получателя и названием вируса. Антивирус способен интегрироваться с системой удаленного администрирования Webmin.

Помимо вирусов, постоянное беспокойство доставляет также нежелательная корреспонденция рекламного характера — спам. Для борьбы с ним «Лаборатория Касперского» разработала систему Kaspersky Anti-Spam. Она работает как отдельный сервер фильтрации или совместно с почтовыми серверами Sendmail, Qmail, Communigate Pro, Postfix и Exim. Надежную фильтрацию сообщений обеспечивают выпускаемые раз в два часа обновления. Поддерживаются операционные системы Linux и FreeBSD 4.X. Продукты «Лаборатории Касперского» просты в установке и настройке, имеют хорошую техническую поддержку, но недешевы. В качестве альтернативы можно рекомендовать распространяемые под лицензией GPL бесплатные Clam AntiVirus (http://www.clamav.net) и SpamAssassin (http://spamassassin.apache.org).

Непреодолимый барьер/_...

Настройка безопасности сервера

Современный Интернет можно уподобить большому мегаполису — настолько широки его возможности и разнообразна представленная в нем информация. Но если вы всерьез решили прописаться в этом «городе», необходимо правильно организовать защиту, которая поможет избавиться не только от хакерских атак, но и от случайных ошибок.

Что такое firewall?

Под английским термином firewall (буквально «огненная стена» — брандмауэр, огнеупорная стенка, разделяющая смежные здания или части одного строения в противопожарных целях) сегодня подразумевается система или группа систем, аппаратных или программных, которые реализуют правила управления доступом между сетями. При этом, как правило, одна часть этих систем работает на блокирование информации, а другая — на пропуск.

Самое главное при выборе firewall и его использовании — знать и понимать, что именно и каким образом вы хотите блокировать или пропускать. Если вы используете свою BSD-систему в качестве домашней рабочей станции, это довольно просто. Гораздо сложнее реализовать на практике корпоративную политику безопасности: ко всему прочему, она должна включать еще и длинный список внесетевых мероприятий, без которых настоящую информационную защиту построить невозможно. Кроме того, при построении firewall надо помнить, что ваш сетевой экран — это ваше «лицо» в сети, способное создать репутацию или подорвать ее, а следовательно, повысить или снизить ваши возможные доходы.

Большинство системных администраторов не упустят случая в шутку упрекнуть коллегу в излишней мнительности и «системной паранойе». Но если говорить серьезно, то каждый, кто выстраивает систему безопасности, должен помнить: идеально защищенным »

» можно назвать лишь тот компьютер, который в данный момент не работает. Все службы, которые вы собираетесь сделать доступными для себя или своих пользователей, могут оказаться «лазейкой» для злоумышленника, стремящегося взломать вашу систему.

Подобный подход нашел отражение в политике безопасности «все запретить, разрешать только нужное». Но он не является единственно верным. Как показывает практика, иногда гораздо удобнее использовать принцип «все разрешить, запретить только ненужное». При этом применение «запретительной» политики означает более высокую стоимость начальной настройки, а применение «разрешительной» влечет за собой затраты на периодическую поддержку.

Есть еще один важный аспект безопасности, который следует иметь в виду. Firewall не может защитить вашу сеть от атак, которые проходят мимо него. Многие руководители компаний, подключаясь к глобальной сети, более всего опасаются потенциальной утечки конфиденциальной информации. Но, к большому несчастью, «увести» информацию с помощью обыкновенной дискеты ничуть не сложнее, чем путем взлома защищенной корпоративной сети. Кроме того, чем больше компания, тем выше вероятность того, что в ней найдется кто-то, способный по ошибке стереть или испортить наиболее важные данные. К сожалению, от опасностей, находящихся «по эту сторону» брандмауэра, сетевой экран вас тоже не защитит.

Для того чтобы правильно настроить сетевой экран, надо разобраться, какие принципы лежат в основе его работы. Каждый пакет, который попадает в firewall, проверяется на совпадение его информационного содержания с заданными правилами. Проверяются, как правило, следующие параметры:

- тип пакета (TCP, UDP, ICMP);
- адрес, с которого пришел пакет (источник);
- порт источника;
- адрес, на который отправляется пакет (фактический получатель);
- порт получателя;
- физический интерфейс движения пакета.

Содержимое пакета в данном случае не имеет никакого значения. Такой подход имеет свои недостатки (например, нет возможности анализировать трафик, передаваемый протоколами прикладного уровня). Однако подобную процедуру обработки можно включать непосредственно в ядро операционной системы. »

Настройка IPFW

Листинг 1. Файл rc.firewall

```
fwcmd ="/usr/bin/ipfw –q"
# Описываем сеть и интерфейсы.
# Внешний интерфейс.
oint="vx0"

# Внешний IP-адрес.
oip="200.200.200.1"

# Внутренний интерфейс.
iint = "vx1"

# Внутренний IP-адрес.
iip="192.168.1.1"
mask="255.255.255.240"

# Запрещаем прохождение
      фрагментированных пакетов.
${fwcmd} add deny icmp from any to any frag

# Запрещаем NetBios-трафик
      вне локальной сети.
${fwcmd} add deny udp from any
      137-139 to any via ${oint}
${fwcmd} add deny udp from any to
      any 137-139 via ${oint}

# Разрешаем трафик по локальному
      интерфейсу.
${fwcmd} add pass all from any
      to any via lo0

# Разрешаем трафик только
      в пределах локальной сети.
${fwcmd} add pass all from any
      to any via ${iint}

# Разрешаем прохождение
      ICMP-пакетов.
${fwcmd} add pass ICMP from any to any

# Разрешаем работу с SMTP-протоколом.
${fwcmd} add pass tcp from any
      to any 25 out
${fwcmd} add pass tcp from any 25
      to any out

# Разрешаем работу
      с HTTPS-протоколом.
${fwcmd} add pass tcp from any
      to any 443 out
${fwcmd} add pass tcp from any 443
      to any out

# Разрешаем работу с HTTP-протоколом.
${fwcmd} add pass tcp from any to any 80
${fwcmd} add pass tcp from any 80 to any

# Разрешаем работу с DNS-серверами.
${fwcmd} add pass udp from any to any 53
${fwcmd} add pass udp from any 53 to any

# Разрешаем работу с NEWS-серверами.
${fwcmd} add pass udp from any
      to any 119 out via vx1
${fwcmd} add pass udp from any 119
      to any out via vx1

# Разрешаем забор почты
      по POP3-протоколу.
${fwcmd} add pass udp from any to any 110
${fwcmd} add pass udp from any 110 to any

# Разрешаем работу с FTP-серверами
${fwcmd} add pass tcp form any 21 to any
${fwcmd} add pass tcp from any to any 21
${fwcmd} add pass tcp from any 20 to any
${fwcmd} add pass tcp from any to any 20

# Разрешаем доступ с домашней
      машины администратора, имеющей
      IP 200.200.200.15.
${fwcmd} add pass tcp from
      200.200.200.15 to any
${fwcmd} add pass tcp from any
      to 200.200.200.15
```

Настройка IPFW

Листинг 2. Файл firewall.sh

```
#!/bin/sh

# Запускаем natd
        с соответствующими параметрами.
natd -use_sockets -same_ports -unregis-
        tered_only -dynamic -interface rl0

# Выполняем отдельным файлом
        загрузку правил firewall.
/etc/firewall/rules.sh
```

Начинаем и зажигаем

В отличие от Windows, в операционной системе FreeBSD (IPFW) firewall встроен в ее состав. Правда, разработчики в свое время посчитали, что такую «стену» не стоит возводить на каждом BSD-компьютере. Поэтому чтобы снабдить свою BSD-машину полноценным межсетевым экраном, необходимо предпринять некоторые действия, которые на первый взгляд кажутся сложными.

Встроенный во FreeBSD firewall не только блокирует нежелательный трафик, но при соответствующей настройке становится мощным инструментом сетевого управления. Если вы не хотите изменять ядро вашей системы, придется ограничиться простейшими функциями брандмауэра типа «запретить-разрешить» (deny-allow). В самом простом случае необходимо лишь запустить его в файле /etc/rc.conf командой firewall_enable=”yes”. При этом, конечно, весьма желательно настроить файл с правилами для брандмауэра (по умолчанию — /etc/ rc.firewall) для нужд конкретного пользователя.

Тем же, кто решил воспользоваться «полным набором» услуг от IPFW, придется заняться сборкой нового ядра. В него надо включить следующие опции:

► IPFIREWALL — именно эта директива «включает» все возможности IPFW.

► IPFIREWALL_VERBOSE — ее включение позволяет firewall вести лог-файл, в котором хранятся записи обо всех событиях. Эта опция полезна для анализа работы системы и слежения за попытками взлома, однако вместе с ней целесообразно использовать ограничение числа попадающих в лог-файл сообщений.

► IPFIREWALL_VERBOSE_LIMIT — при отсутствии лимита злоумышленник или просто сбойный компьютер в сети способны практически моментально сгенерировать огромное число пакетов. Это может привести к тому, что лог-файл firewall займет все имеющееся дисковое пространство.

► IPFIREWALL_FORWARD и IPDIVERT разрешают пользователю использовать два самых мощных инструмента IPFW: методы перенаправления другому адресату и передача пакета на обработку внешней программы. Метод FORWARD (fwd) дает возможность перенаправлять пакеты другому адресату. Используя его, можно создать, например, HTTP-прокси сервер, работающий для всех клиентов в принудительном режиме (transparent proxy), или перенести некоторые сервисы (WWW, Mail) внутрь закрытой сети так, чтобы они, тем не менее, оставались доступными из внешнего мира. Метод DIVERT передает пакет на обработку внешней программе, что позволяет организовать доступ к «большой» сети клиентам, находящимся под защитой брандмауэра. Кроме того, пакеты можно «завернуть» в специальную программу для подсчета трафика.

► IPFIREWALL_DEFAULT_TO_ACCEPT — по умолчанию IPFW включается в режиме «все запретить». Данная опция обеспечивает начальный старт firewall в режиме «все разрешено». Если вы не совсем уверены в том, что вам это нужно, то лучше всего оставить «параноидальный» режим по умолчанию. Если же вы используете FreeBSD на бездисковой рабочей станции, отсутствие этой опции в ядре не позволит операционной системе использовать сетевые диски и загрузить с них необходимый рабочий набор правил.

► DUMMYNET — одна из самых полезных администраторских возможностей. Данная команда включает в ядро систему ограничения пропускной способности каналов, основанную на задержке прохождения пакетов через роутер. Для многих системных администраторов именно эта возможность стала определяющей при выборе операционной системы: переадресацию пакетов и NAT можно организовать на практически любой сетевой ОС, но с ограничением пропускной способности мало кто справится так же хорошо, как FreeBSD.

Из истории вопроса

Становление «огненных стен»

Первое поколение firewall представляло собой всего лишь маршрутизаторы с пакетной фильтрацией. Они впервые появились примерно в 1985 году и до сих пор остаются самым популярным типом сетевых экранов.

Второе поколение «огненных стен» — firewall цепного уровня — использовали механизм цепной передачи информации от источника к получателю. При этом firewall проверял как целостность всей цепи пакетов, так и соответствие источника, получателя и некоторых других параметров заданным правилам. Целый пакет собирался на firewall и лишь после этого передавался получателю.

Третье поколение — firewall программного уровня (Application Layer firewall) — могут проверять и сами данные, передаваемые в пакетах. Это позволяет отслеживать передачу паролей. Вместе с ними используется прокси-сервис, который кеширует информацию для ускорения ее обработки.

Четвертое поколение firewall добавляет к функциям своих предшественников полезную возможность динамического изменения правил фильтрации (Dynamic Packet Filter firewall).

Пятое поколение firewall, которое появилось в 1996 году, базируется на архитектуре Kernel Proxy. Ее идея состоит в том, чтобы встроить механизм программного уровня непосредственно в ядро операционной системы, что значительно ускоряет процесс обработки.

» ► HZ, TCP_DROP_SYNFIN, ICMP_BANDLIM — эти опции используются для увеличения производительности firewall. HZ определяет частоту просмотра системой DUMMYNET-очереди пакетов, поставленных на отправку. ICMP_BANDLIM и TCP_DROP_SYNFIN увеличивают защищенность системы от хакерских атак, позволяя ограничивать количество ошибочных ICMP-сообщений, генерируемых системой, и отбрасывать TCP-пакеты, в которых одновременно установлены флаги начала и завершения соединений.
► BRIDGE — данная опция позволяет превратить компьютер в особый тип сетевого соединения — мост. В некоторых случаях, особенно в сочетании с опцией DUMMYNET, это поможет создать простую, но хорошо управляемую систему.

Без музыкального слуха не обойтись

По завершении успешной сборки и компиляции нового ядра сетевой экран автоматически стартует после перезагрузки системы. При этом с помощью команды dmesg можно увидеть сообщения об успешном его старте:

```
«BRIDGE 020214 loaded
DUMMYNET initialized (011031)
IP packet filtering initialized, divert
    enabled, rule-based forwarding
    enabled, deaault to accept, logging
    limited to 100 packets/entry
    by default».
```

Однако без соответствующей настройки ваш брандмауэр не будет работать так, как вам этого хочется. Более того: если при старте системы показываются именно те сообщения, которые приведены выше, firewall, скорее всего, не будет работать вообще. Дело в том, что в настройки сетевого экрана автоматически добавляется правило под номером 65535. Его нельзя ни удалить, ни исправить. При наличии в ядре опции IP-FIREWALL_DEFAULT_TO_ACCEPT оно разрешает весь трафик (65535 allow all from any to any); а при ее отсутствии — запрещает весь трафик (65535 deny all from any to any).

Искусство написания правил для IPFW заключается в том, чтобы должным образом распределить правила блокировки и пропуска трафика. Подробную процедуру добавления правил можно, как обычно, выяснить с помощью команды man ipfw. Сам по себе процесс выстраивания «защитной стены» довольно тривиален. Подробный пример типичного firewall для обычного роутера можно увидеть на листинге 1.

Локально выходим в глобальную сеть

Гораздо интереснее вместе с защитными функциями реализовать с помощью механизмов IPFW некоторые сервисные возможности. Наиболее типичной для сети малого офиса можно считать задачу организации доступа в сеть из локальной сети с использованием единственного внешнего IP-адреса. Средствами IPFW эта задача решается следующим образом.

Прежде всего, необходимо разрешить демону natd стартовать при загрузке. Сделать это можно либо с помощью дополнительного скрипта, либо с помощью команд в файле /rct/rc.conf:

```
gateway_enable="YES"
natd_enable="YES"
natd_interface="rl0"
```

Последняя команда в качестве значения должна использовать имя внешнего интерфейса, то есть именно с этой «сетевой стороны» должен располагаться ваш провайдер. После этого необходимо «развернуть» все входящие пакеты на порт демона natd. Делается это с помощью команды divert: ipfw add divert natd all from any to any via rl0

Для удобства можно всю процедуру запускать автоматически. Можно, например, создать для этого специальную директорию firewall в /etc/, и затем создать в ней пару файлов (см. Листинги 2 и 3). Вам останется лишь добавить в /etc/rc.conf пару строк:

```
firewall_enable="YES"
```

(эта строка у вас уже должна быть) и

```
firewall_script="/ect/firewall/firewall.sh".
```

Популярным вариантом решения этой задачи можно назвать организацию «прозрачного» проксирования. Ее идея заключается в том, что пользовательские запросы принудительно направляются на прокси-сервер (чаще всего squid). За счет этого достигается некоторая экономия потребления трафика. По некоторым оценкам, при правильной настройке она может составить до 30%.

Организовать подобную схему довольно просто — достаточно лишь перенаправить запросы, приходящие от пользователя, на порт прокси-сервера. Делается это единственной командой:

```
ipfw add fwd 127.0.0.1,3128 tcp from
    <адрес внутренней сети> to any 80.
```

При этом необходимо помнить, что порт 3128 в предыдущей команде должен обозначать именно тот порт, на котором работает прокси-сервер. Кроме того, в файл настройки squid необходимо добавить несколько строк:

```
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

»

Настройка IPFW

Листинг 3. Файл rules.sh

```
#!/bin/sh

# Определяем внутренний интерфейс,
    IP-адрес и …
iip="192.168.0.1"

# имя
iint="rl1"

# Определяем внешний интерфейс,
    IP-адрес и …
oip="212.XXX.XXX.3"

# имя
oint="rl0"
```

Сложный локальный случай

Необходимо упомянуть еще одно приложение встроенного firewall FreeBSD, которое самым лучшим способом может проиллюстрировать его «расширенные» возможности. Речь пойдет о так называемом Policy-Based Routing (PBR). В самом простом случае это понятие обозначает способ перенаправления трафика в соответствии с заранее заданными условиями (например, адресом источника). Проще говоря, при наличии нескольких каналов для подключения к сети можно оптимальным образом организовать работу своих клиентов. Допустим, у нас есть два «внешних» сетевых интерфейса: fxp0 — 1.1.1.1 /24 (gateway 1.1.1.111 ISP1) и fxp1 — 2.2.2.2 /24 (gateway 2.2.2.222 ISP2), и два «внутренних»: rl0 — 3.3.3.3/24 (сеть NET1) и rl1 — 4.4.4.4 /24 (сеть NET2). Задача состоит в том, чтобы направить трафик сети NET1 через провайдера ISP1, а трафик сети NET2 — через провайдера ISP2. Очевидно, что для каждого «внутреннего» сегмента необходимо поднять свой собственный демон natd. При этом следует помнить, что параметры запуска natd принципиально могут совпадать, но порты, которые будут «прослушивать» эти демоны, должны быть разными (natd -a 1.1.1.1 -p 8668 и natd -a 2.2.2.2 -p 8778). После этого надо создать набор правил для ipfw, которые, собственно, и реализуют поставленную задачу (см. Листинг 4).

Представленная задача, несмотря на ее кажущуюся сложность, решается довольно просто, но только в данном конкретном случае. Довольно часто встречается неправильное применение PBR. Как правило, происходит это тогда, когда на firewall, выполняющем еще и роль роутера, исполняются сервисы (например, WWW), привязанные к конкретному адресу конкретного интерфейса. В этом случае необходима дополнительная настройка.

Управление скоростью

Вторая по важности задача, которую обычно решают системные администраторы, — разделение ширины полосы пропускания между пользователями. В любом случае полезно уметь настраивать тот самый «кран», сквозь который трафик поступает к вашим клиентам. В простейшем случае, когда необходимо просто ограничить ширину канала для одного из них, можно применять конструкцию следующего вида (см. Листинг 5).

Давайте правильно конфигурировать

Как только вы всерьез займетесь конфигурированием вашего брандмауэра, обязательно встанет вопрос о правильном порядке написания правил для него. Необходимо помнить, что все правила IPFW выполняются последовательно, и как только какое-то из них выполняется для данного пакета, дальнейший анализ не ведется. Поэтому порядок написания крайне важен. Есть практически испытанная и наиболее эффективная схема написания правил для firewall, которая должна работать не только во FreeBSD, но и во всех остальных системах, поскольку логика ее достаточно универсальна.

В первую очередь необходимо запретить вредный трафик. Инструкции deny и reject должны идти в самом начале. Далее для сбора внешней статистики, если это необходимо, нужно использовать инструкции count на внешних интерфейсах. После чего наступает черед инструкции forward. После того как данные посчитаны на внешнем интерфейсе, они могут быть перенаправлены внутрь локальной сети или на внешние объекты. После этого необходимо настроить внутренние обработчики трафика — директивы divert. Далее, если это нужно, стоит посчитать трафик на внутренней сети — count на внутренний интерфейс. И только потом можно открывать командой allow внутреннюю сеть и соединения, разрешенные непосредственно для существующего роутера. В заключение следует включить log и запрет для всего остального, что каким-то образом сумеет проскочить через все предыдущие правила.

Конечно, если вы настроили свой сетевой экран, предварительно не ознакомившись с приведенными рекомендациями, и вполне довольны его работой, — перенастраивать пока ничего не надо. По крайней мере, до тех пор, пока вы не убедитесь, что все можно сделать еще лучше.

Настройка IPFW

Листинг 4. Файл policy_base.sh

```
#!/bin/sh

# Направляем исходящий трафик сети NET1
      на первый демон natd (порт 8668)
ipfw add 10 divert 8668 ip from
      3.3.3.0/24 to any

# Направляем исходящий трафик сети NET2
      на второй демон natd (порт 8778)
ipfw add 20 divert 8778 ip
      from 4.4.4.0/24 to any

# Направляем трафик с внешнего интер-
      фейса fxp0 на роутер провайдера ISP1
ipfw 30 add fwd 1.1.1.111 ip from
      1.1.1.1 to any

# Направляем трафик с внешнего
      интерфейса fxp1 на роутер
      провайдера ISP2
ipfw 40 add fwd 2.2.2.222 ip from
      2.2.2.2 to any

# Направляем входящий трафик провайде-
      ра ISP1 на первый демон natd
      (порт 8668)
ipfw 50 add divert 8668 ip from any to 1.1.1.1

# Направляем входящий трафик провайде-
      ра ISP2 на первый демон natd
      (порт 8778)
ipfw 60 add divert 8778 ip from any
      to 2.2.2.2
```

IPFilter – компактность и производительность

Пакет IPFilter, созданный независимой командой разработчиков, в настоящее время поставляется не только в составе FreeBSD, но и еще как минимум в паре десятков Unix-систем. Преимуществ у него довольного много: во-первых, он компактнее »

Настройка IPFW

Листинг 5. Ограничение полосы пропускания

```
# Конфигурируем «трубу» (pipe) 1:
    ограничиваем ширину канала
    в 1 Мбит/с
ipfw pipe 1 config bw 1Mbit/s

# Конфигурируем «трубу» (pipe) 2:
    ограничиваем ширину канала
    в 128 Кбит/с
ipfw pipe 2 config bw 128Kbit/s

# Определяем, что для клиента
    192.168.0.1 на вход ширина канал
    не будет превышать 1 Мбит/с
ipfw add pipe 1 ip from any
    to 192.168.0.1 in

# То же самое, но на исходящий трафик
    выделяется не более 128 Кбит/с
ipfw add pipe 2 ip from 192.168.0.1 to any out
```

» встроенного IPFW; во-вторых, его интерфейс более понятен даже непосвященному пользователю; и в-третьих, приемы и решения типичных сетевых задач в этом пакете выглядят проще, нежели у IPFW.

Чтобы включить IPFilter в систему, необходимо добавить в ядро опции IPFILTER и IPFILTER_LOG. После этого ядро, естественно, придется собрать заново, зато после перезагрузки вы получите готовую для дальнейших экспериментов систему.

Основная задача любого сетевого экрана — блокирование и (или) пропуск данных. В IPFilter все решается довольно просто: трафик блокируется командой block и разрешается командой pass. Кроме собственно пакетного фильтра, в состав IPFilter входят утилиты мониторинга ipmon, утилита сбора и отображения статистики ipfstat, несколько тестовых утилит и программа ipnat, которая пользуется наибольшей популярностью, так как позволяет быстро и просто организовать доступ из локальной сети в Интернет.

Если говорить конкретно, то в самом простом случае после установки IPFilter вам надо лишь создать файл /etc/ipnat.rules и поместить в него строку: map int1 192.168.0.0/24 → 123.123.123.123/32. Здесь int1 — интерфейс с реальным IP-адресом (на котором будет производиться трансляция), 123.123.123.123 — реальный IP, выданный провайдером, 192.168.0.0/24 — блок адресов локальной сети. После этого вам останется лишь добавить в /etc/rc.local строку запуска ipnat при загрузке системы: ipnat -f /etc/ipnat.rules.

Для безусловного перенаправления (redirect) пакетов, пришедших извне (например, для «прозрачного» доступа к WWW/SMTP-серверам, расположенным во внутренней сети) используется команда rdr fxp0 200.200.200.1/32 port 8080 → 192.168.1.17 port 80 tcp. В данном случае любой пакет, пришедший на порт 8080 внешнего интерфейса 200.200.200.1, будет перенаправлен на 192.168.1.18 порт 80.

Еще одно неоспоримое удобство ipnat состоит в умении этой программы не только транслировать внутренние адреса во внешнюю сеть, но и некоторым образом анализировать передаваемые данные. Лучше всего это можно продемонстрировать на примере протокола FTP. Приведенных выше простеньких команд будет недостаточно для корректной работы этого протокола, поскольку это сетевое взаимодействие требует указания реально существующего IP-адреса клиента. Проблему можно решить добавлением следующей команды: map fxp0 192.168.1.149/32 → 200.200.200.1/32 proxy port ftp ftp/tcp.

Следуя этой директиве, ipnat будет подставлять в данные, передаваемые во время FTP-сеанса, нужные адреса. Следует лишь помнить, что прокси-правило должно обязательно стоять перед другими правилами (за исключением redirect).

Быть иль не быть? Не в том вопрос

В самом деле, для того, кто хоть пару недель провел в глобальной сети, вопроса о необходимости сетевого защитного экрана не существует. Обязательность его использования определяется даже не потенциальной опасностью «большого и злобного» Интернета, а удобством этого мощного сетевого инструмента. Кроме того, следует отметить, что во многих случаях использование правильно настроенного брандмауэра позволяет защитить внутреннюю корпоративную сеть от некоторых видов сетевых вирусов, которые способны атаковать компьютеры по заранее известным портам. Конечно, со временем могут появиться модификации вредоносных программ, которые будут искать лазейку через другие порты. Но даже если вас пугает некоторая относительная сложность настройки межсетевого экрана, обязательно найдите время и силы, чтобы внимательно и не спеша в этом разобраться. Потому что, грамотно сделав это один раз, вы почти наверняка сможете забыть о каком-либо сбое в локальной сети на достаточно долгое время.

■ ■ ■ Сергей Кондрашев

Вопросы терминологии

Firewall, брандмауэр или сетевой экран?

Без правильного употребления специальной терминологии невозможно достичь взаимопонимания специалистов и пользователей. В приложении к средствам сетевой защиты все три упомянутых термина употребляются в нашей стране как синонимы, хотя это не совсем верно. Термин firewall больше подходит для описания маршрутизатора с фильтрацией пакетов, работающего на сетевом уровне модели OSI. Позже непосредственно в фильтрах началась реализация политики безопасности, что привело к созданию шлюза уровня приложений (брандмауэра). Термин «межсетевой экран» был принят для обозначения совокупности компонентов, которые находятся между внешней и защищаемой сетью и образуют «барьер».

Парад посредников/_

Обзор файловых протоколов

Многообразие сетевых протоколов позволяет решить любые задачи, связанные с совместной работой в сети: обеспечить высокую защищенность, быстродействие, а также обмен данными между разнородными сетями.

Условия и требования передачи файлов могут заметно различаться в зависимости от решаемых задач. Поэтому особенно важно правильно настроить файл-сервер, выбрав протокол передачи данных, сетевую файловую систему и одну из реализаций сервера. Ниже будут описаны основные используемые протоколы передачи файлов, их достоинства и недостатки. Однако окончательный выбор, как всегда, остается за вами.

Ветеран Интернета

FTP — один из самых «старых» протоколов среди применяющихся сегодня. Однако он до сих пор не потерял актуальности при использовании в локальных сетях. Это связано и с регулярным обнаружением новых уязвимостей реализации SMB-протокола в Windows, и с появлением простых в настройке и «легких» FTP-серверов.

Клиенты и серверы для FTP существуют для всех систем, даже для Palm OS. Никакого шифрования данных или паролей протоколом не предусмотрено, хотя существует схема, при которой общение между клиентом и сервером осуществляется через SSL-соединение. Однако данный вариант потребует от вас либо специальных клиентов, либо предварительной ручной установки SSL-туннеля (например, с помощью пакета openssl, о котором вы можете прочитать в другой статье), поэтому он пока не получил широкого распространения.

Важной особенностью FTP является использование во время сессии одновременно нескольких соединений. Команды сер- »

» веру от клиента передаются по двадцать первому TCP-порту. Для этого каждый раз устанавливается новое TCP-соединение, инициатором которого может служить как сервер, так и клиент.

В так называемом активном режиме (Active Mode) клиент сообщает серверу адрес и порт, на котором он ожидает данные, после чего сервер устанавливает соединение по указанному адресу. Далее происходит непосредственно передача данных: списка доступных на сервере файлов (каталогов) или собственно файла. Эта особенность активного режима требует специальных мер при использовании NAT, так как клиент в данном случае сообщает свой адрес из «внутренней» сети, который недоступен для сервера. Во FreeBSD для решения этой проблемы нужно включить ftp proxy модуль в /etc/ipnat.rules перед остальными правилами трансляции:

```
map fxp0 0/0 -> 0/32 proxy port ftp ftp/tcp
```

В пассивном режиме (Passive Mode) для передачи данных сервер открывает сокет на своей стороне, сообщает об этом клиенту и ждет установки соединения для начала передачи данных.

Получить наиболее широкие возможности и отличную производительность позволяет использование пакета proftpd (http://www.proftpd.org). Синтаксис его конфигурационного файла похож на применяемый в Apache, что облегчает понимание и облегчает настройку. Он поддерживает различные форматы (backends) для хранения авторизационных данных пользователей: PAM, LDAP, SQL и passwd. Схожей функциональностью обладает pureftpd (http://www.pureftpd.org). Любителям минимализма можно рекомендовать vsftpd (http://vsftpd.beasts.org), который разрабатывался исходя из принципов максимальной безопасности.

Универсальность ценой защищенности

Протокол SMB (Server Message Block) представляет собой нечто большее, чем просто протокол для обмена файлами. Например, он позволяет совместно использовать принтеры, последовательные порты и даже такие абстрактные ресурсы, как named pipes и слоты для обмена сообщениями (mailslots). SMB использует протокол более низкого уровня — NetBIOS, разработанный IBM в 1985 году, который, в свою очередь, может работать на любом другом протоколе третьего уровня: TCP/IP, SPX/IPX, DECnet или специально разработанном для транспорта NetBIOS-пакетов в небольших сетях NetBEUI. В NetBIOS/SMB входят не только методы для работы с файлами, но также средства обнаружения («browsing») SMB-серверов, что отличает этот протокол от всех остальных. Серверы с помощью широковещательных пакетов анонсируют свое присутствие в сети, а также отвечают на широковещательные запросы от клиентов, что позволяет последним иметь актуальные списки своего сетевого окружения.

В SMB используются две модели защиты: уровня ресурса (share level) и уровня пользователя (user level). В первом случае устанавливается пароль на ресурс (share) в целом, и при успешной аутентификации клиент получает доступ ко всем файлам, находящимся внутри. Во втором случае при установке сессии происходит аутентификация пользователя, и ему выдается UID, который затем применяется для определения прав доступа на файловом уровне.

При работе используются фиксированные «хорошо известные» (well-known) TCP/UDP-порты (135, 137-139, 445) что позволяет легко настроить firewall для защиты от посторонних вторжений. Это особенно актуально, если вспомнить о регулярно обнаруживаемых уязвимостях в сетевых сервисах от Microsoft. Защиты передаваемых данных не предусмотрено, за исключением шифрования пароля при аутентификации.

Для данного протокола характерны небольшие задержки при работе с файловой системой и, при должной настройке сервера, высокая скорость передачи данных. Значительным преимуществом является поддержка различных кодовых страниц, что позволяет клиентам, использующим FreeBSD, «понимать» русские имена файлов на Windows-сервере, и наоборот.

В пакет Samba входит smbclient, предоставляющий, помимо пользовательского интерфейса командной строки, схожего с FTP, еще и возможность печати на удаленных принтерах. Samba-сервер позволяет компьютеру с FreeBSD подключаться к NT-домену как клиенту или контроллеру домена (разумеется, поддерживается и работа без него), предоставлять Windows- и Unix-клиентам доступ к локальному принтеру, а также поддерживать список серверов для своей рабочей группы.

Протокол не стоит использовать при наличии заметных задержек в сети (например, когда пакетам нужно пройти через несколько маршрутизаторов). В целом SMB делает ограничение доступа к файлам на уровне ресурсов несколько проще, а множество настроек Samba-сервера и SWAT (веб-интерфейс для его конфигурирования, также входящий в пакет Samba) позволяют оптимально настроить сервер даже начинающему администратору.

Сеть из мира Unix

В 1985 году компания Sun Microsystems выпустила первую версию сетевой файловой системы NFS (Network File System). Изначально предназначенная для экспорти- »

Другие файловые системы

Удобный доступ

Проект Linux UserLand Filesystem (http://lufs.sourceforge.net) расширяет возможности FreeBSD, создавая виртуальную файловую систему, прозрачную для пользовательских приложений.
Она состоит из модуля для ядра, обеспечивающего общение со специально написанными демонами, которые работают как обычные приложения. Реализация их не в виде модулей ядра, с одной стороны, значительно увеличивает нагрузку на систему, но с другой стороны, позволяет реализовать поддержку протоколов любой сложности. Это обеспечивает возможность монтирования ресурсов FTP, SSH и Gnutella к локальной файловой системе.

» рования частей файловой системы с одного сервера на другой, она активно использовала RPC для общения компонентов между собой, поддерживала совместный доступ к файлам, а также все их атрибуты, использующиеся в Unix-системах, и являлась совершенно прозрачной для пользователя. Использование UDP-протокола вместо TCP уменьшило влияние сетевых задержек и требовательность слабых по нынешним меркам компьютеров того времени к ресурсам. В некоторых реализациях NFS приобрела различные полезные дополнения, в частности, поддержку кеширования файлов NFS-сервера на диске клиента (обратите внимание — это не является уникальным свойством AFS!).

Время шло, требования росли. Первое значительное изменение NFS-протокола в основном было связано с необходимостью поддержки файлов объемом свыше 2 Гбайт. В состав этого изменения вошли:

- увеличение максимального размера блока данных при операциях чтения/записи до 32 Кбайт (large block file transfers);
- поддержка отложенной записи (прежние стандарты требовали от сервера сбросить данные на диск или в NVRAM, прежде чем отвечать на клиентский запрос на запись);
- readdirplus — возврат атрибутов файлов вместе с листингом каталога за одну операцию (в старых версиях для получения атрибутов всех файлов в каталоге с N файлами потребовалась бы N+1 операция);
- поддержка TCP-протокола, что положительно сказалось на загрузке маршрутизаторов и firewall.

Все это привело к созданию сетевой файловой системы, которая стала настоящим чемпионом по производительности практически для любых операций внутри локальной сети. Основным недостатком NFS осталась слабая защищенность. Система изначально создавалась для экспортирования файловой системы с Unix-хоста на Unix-хост внутри корпоративной сети. Сами пользователи работают с ней как с частью файловой системы, никаких клиентских программ, как для FTP, не требуется, поэтому и защита была реализована достаточно примитивно. Она работает только при условии, что и сеть, и клиент, и сервер защищены от хакерских действий. NFS-запросы должны приходить с привилегированных портов 1-1024, которые не могут быть использованы пользовательскими приложениями в Unix. Права доступа к файлу определяются UID пользователя, переданного клиентом серверу.

Со временем серверы обзавелись дополнительными возможностями:

- root_squash и all_squash указывают серверу, что операции, заявленные клиентом как проводящиеся с uid=0 (root_squash) или с любым UID вообще (all_squash), должны исполняться с привилегиями пользователя nobody. Это позволяет легко организовать анонимный readonly-доступ к несекретной информации (например, к видеофайлам, музыке, документации, дистрибутивам, /usr/share и так далее);
- uid mapping позволяет организовать трансляцию клиентских UID в соответствующие им UID на стороне сервера, что полезно, например, когда пользователи с оди- »

Сравнительные характеристики протоколов

	FTP	SMB	NFS
Скорость линейного чтения	высокая	высокая	очень высокая
Скорость случайного доступа и операций с файловой системой	низкая	высокая	очень высокая
Возможность монтирования в Linux	через LUFS	есть, smbfs	есть
Возможность монтирования в других Unix-ОС	нет	FreeBSD	все Unix-системы
Бесплатные клиенты для Windows	встроен в Explorer, FTP, FAR	встроен в Explorer	есть SFU v3.5
Защищенность	низкая	средняя; только шифрование паролей	низкая
Сложность настройки клиентов в Windows и Linux	очень просто	Windows — очень просто Linux — просто	Windows — средняя Linux — просто
Сложность настройки серверной части	просто	просто	средняя
Рекомендуемая область применения	хранение данных, не требующих случайного доступа, анонимные файл-серверы с дистрибутивами, мультимедийной информацией и т. п.	хранение любых разнородных данных, кроме особо секретных	разделение файловой системы между серверами либо организация файл-сервера с анонимным доступом только на чтение
	применим в сетях любых масштабов	применима в локальных сетях и сетях масштаба предприятия с достаточно низкими задержками	применима в локальных сетях
			при открытии доступа на запись применима только в сетях, защищенных от несанкционированного доступа

» наковыми именами имеют различные UID на сервере и клиенте;

► insecure mounting и возможность привязки RPC-сервисов к определенным портам отменяют обязательное условие использования клиентом портов 1-1024. Это избавило многих администраторов, настраивающих firewall, от подлинных кошмаров.

Для привязки сервисов NFS к определенным портам, чтобы облегчить себе жизнь при настройке брандмауэра, стоит просмотреть следующие страницы документации: rpc.statd (ключ "-o"), rpc.mountd (ключ "-p"), rpc.rquotad (ключ "-p"), rpc.nfsd (ключ "-p").

Есть несколько реализаций NFS-серверов, поддерживающих шифрование трафика. Это, например, sNFS (http://www.crufty.net/ftp/pub/sjg/help/sNFS.html). Кроме того, при работе с NFS через TCP, как и практически для любой другой сетевой файловой системы, трафик можно перенаправить в предварительно установленный между хостами SSL-туннель (созданный, например, с помощью openssl, SSH или stunnel (http://www.stunnel.org). В NFSv4 (http://nfsv4.org) поддерживается Kerberos-аутентификация и шифрование трафика.

Существует несколько NFS-клиентов для Windows, большинство из которых являются коммерческими (http://hummingbird.com/products/nc/nfs/index.html?cks=y). Однако Microsoft не так давно сделала бесплатным свой Services For Unix v3.5 (http://www.microsoft.com/windows/sfu), в состав которого входят и NFS-клиент, и NFS-сервер.

На сегодня область применения NFS не изменилась, и эта файловая система по-прежнему является самым универсальным методом для обмена частями файловой системы между логически связанными друг с другом серверами. С ее помощью можно раздавать дистрибутивы и обновления для серверов (используя параметр all_squash в /etc/exports). При возможности создания изолированного сегмента сети для серверов можно получить общий /home. Можно создать также бездисковые рабочие станции (http://www.linuxcenter.ru/lib/networking/nfs_root_minihowto.phtml), (http://www.remoteboot.ru/ru/remoteboot/dskless.html).

	SFTP/SCP	AFS	HTTP/WebDAV
	низкая	высокая	высокая без использования SSL
	низкая, случайный доступ не поддерживается	высокая	низкая
	через LUFS	есть, AFS	нет
	нет	все Unix-системы	нет
	есть SSH с www.ssh.com, PuTTY	есть, OpenAFS	встроен в Explorer
	очень надежное шифрование и трафика, и паролей	надежное шифрование данных, авторизация через Kerberos	шифрование паролей, или трафика, или и того, и другого
	Windows — средняя Linux — средняя	Windows — сложно Linux — средняя	Windows — очень просто Linux — просто
	очень просто	сложно	средняя, в IBM HTTP Server — просто
	периодический доступ пользователей сервера к своим данным	децентрализованное хранение пользовательских или любых других данных в организациях и учебных заведениях	работа с данными через SSL
	требуется shell-доступ к серверу	применима в любых сетях, в том числе и при работе через Интернет, благодаря эффективному алгоритму кеширования данных	раздача файлов через Интернет
	высокая защищенность позволяет безопасно передавать информацию через любые сети		

Немалый интерес представляет возможность смонтировать часть файловой системы сервера в chroot-окружение какого-либо демона, работающего на том же самом сервере, в readonly-режиме: в случае взлома это гарантирует вам отсутствие «троянов», например, в /usr. «Легкость» NFS в этом случае позволяет свести к минимуму издержки монтирования удаленной файловой системы.

Для корпоративной сети

Файловую систему AFS (Andrew File System) разработали в известном Университете Карнеги-Меллоуна, после чего дальнейшая ее поддержка и разработка долгое время осуществлялась на коммерческой основе корпорацией Transarc. Хотя клиенты для нее были бесплатны, серверную часть приходилось покупать. Ситуация изменилась, когда Transarc была приобретена IBM: в конце 2000 года код IBM AFS 3.6 был опубликован под свободной лицензией как OpenAFS 1.0 .

AFS — распределенная файловая система, способная эффективно работать и в локальных сетях, и в Интернете. Пользователю она представляется одним большим диском, хотя на самом деле часть данных может лежать на ближайшем к нему сервере, а другая часть — на сервере в другом городе. Для оптимизации производительности данные, забранные с сервера, кешируются на диске клиента, после чего работа с ними по скорости ничем не отличается от работы с локальными файлами. При внесении в файл изменений они откладываются до его закрытия или сохранения, благодаря чему объем трафика между сервером и клиентом значительно уменьшается.

Авторизация пользователей происходит с помощью Kerberos, что обеспечивает защищенную аутентификацию. Нужную информацию предоставляет сам пользователь в начале сессии, что не требует хранения паролей всех пользователей на клиенте для автоматического монтирования и обеспечивает надежную защиту пользовательских данных. Для поддержки прозрачной авторизации на клиенте рекомендуется установить имя пользователя, модифицированное для поддержки аутентификации в AFS (например, эту задачу значительно облегчает PAM). »

» В AFS предусмотрена гибкая система списков прав доступа (ACL), позволяющая предоставлять необходимые для совместной работы права группам пользователей. Очень полезной особенностью является поддержка репликации частей файловой системы на другие AFS-серверы. Это заметно повышает надежность хранения данных, а иногда и скорость доступа к ним, если копия необходимой для клиента информации находится «ближе» оригинала. Поддерживаются так называемые backup volumes, которые позволяют интуитивно понятным для пользователя образом организовать резервное копирование и хранение пользовательских данных на ленте. В отличие от классической организации этого процесса, в данном случае пользователь может самостоятельно восстановить данные с backup volume, работая с ним как с обычным каталогом. Клиенты и серверы существуют для всех основных платформ, включая Mac OS X.

Из недостатков AFS можно отметить сложность в администрировании и тяжеловесность серверной части. Однако ее надежность, защищенность и хорошая масштабируемость делает AFS неплохим выбором для использования внутри большой корпоративной сети или организации grid-вычислений. В качестве альтернативы AFS можно рассматривать файловую систему Intermezzo (http://www.inter-mezzo.org), которая обладает схожими функциональными возможностями. Основной акцент при ее разработке делается на высокой доступности хранимых данных. Поддержка работы в автономном режиме, когда связь с сервером по каким-либо причинам разрывается, не только полезна для grid-вычислений, но и очень удобна для использования на ноутбуках, так как эта возможность обеспечивает пользователю прозрачную синхронизацию данных с рабочей станцией.

Вездесущий HTTP

HTTP был рожден в начале девяностых годов прошлого века по заказу CERN. Несмотря на свою простоту (реализации HTTP-серверов существуют на всех языках, даже на Shell и Postscript — http://www.pugo.org), протокол получился достаточно гибким, и с его помощью можно передавать не только сам гипертекст, но и любую другую информацию.

Главное отличие HTTP от всех описанных ранее протоколов — это отсутствие в нем понятий «сессия» и «файловая система». Следствием этого стали отсутствие «текущего каталога», необходимость подтверждения авторизации при каждом запросе к серверу, невозможность таких операций, как перемещение файлов из каталога в каталог, переименование и удаление объектов и т. п.

Протокол не делает различия между файлами и любой другой, в том числе и динамически генерируемой, информацией, поэтому нет никакой возможности узнать, что скрывается за ссылкой: файл с фотографией любимой бабушки, HTML-документ с последними новостями или листинг каталога (который, как правило, является обычным HTML-документом, создаваемым сервером «на лету»). Следовательно, такая же простая реализация клиента, как с FTP, становится невозможной. В браузерах скачивать файлы приходится по одному, об их размерах, если администратор не предусмотрел показ нужной информации, остается только догадываться. Существующие клиенты, поддерживающие рекурсивную скачку, зачастую рискуют, пойдя не по той ссылке, прихватить заодно с парой любимых песен половину всего сайта. С закачкой дело обстоит еще хуже. В лучшем случае файл можно «прицепить» к форме, после чего CGI-скрипт должен ее обработать, и сохранить полученные данные в нужном месте на сайте. Если учесть, что максимальный размер POST-запроса обычно ограничен на уровне сервера и не превышает пары мегабайт, то в целом построение сервера для работы с файлами оборачивается значительными трудозатратами для администраторов и большими неудобствами для пользователей. Для устранения этих недостатков было разработано расширение HTTP-протокола — WebDAV.

Если быть точным, то WebDAV вовсе не ограничивается работой с файлами (но это уже — тема для отдельной статьи). Он использует HTTP как транспорт и добавляет все необходимые для реализации файловой системы методы, например:

- блокировки (locking);
- атрибуты (properties), причем не только стандартные, присущие файлам, — возможно также добавление любой метаинформации, например, списка авторов или набора ключевых слов;
- операции над пространством имен, в частности, создание, копирование или перемещение объектов.

Используя для доступа к файлам WebDAV, клиенты получают все преимущества HTTP-протокола перед FTP: шифрование атрибутов доступа при аутентификации, шифрование данных при передаче (с применением HTTPS), поддержку HTTP-прокси и кеширование документов. Кроме того, при передаче группы файлов используется всего одно TCP-соединение.

Для веб-сервера apache (http://httpd.apache.org) поддержка WebDAV реализуется через mod_dav (http://www.webdav.org/mod_dav). Протокол также включен в IBM HTTP-server (http://www.ibm.com/software/webservers/httpservers), основанный на коде apache, который к тому же имеет веб-интерфейс, позволяющий легко настроить любой параметр httpd.conf.

■ ■ ■ Константин Стародубцев

Программистам на заметку

Быстрая передача данных

Системный вызов sendfile (man 2 sendfile) осуществляет передачу данных на уровне ядра между двумя открытыми файловыми дескрипторами (включая TCP-сокеты). Поскольку при этом не приходится копировать данные в промежуточный буфер в сегменте данных программы и обратно, то загрузка процессора и памяти значительно снижается, повышается быстродействие. Вызов sendfile не входит в стандарт POSIX, но поддерживается многими операционными системами, однако его реализация может различаться. Это требует особого подхода при написании программ, использующих данную возможность.

Как сделать домашнюю страницу

Как установить Windows

Как составить фотоальбом

к грамотно оформить документ в Word

Как организовать рабочее врем

Как получать почту

составить таблицу в Excel

Как оформить Рабочий стол

Как найти информацию в Интернет

...# Игра в защите/_...

Защита от несанкционированного доступа

Информационная безопасность — это комплекс мер по защите данных от несанкционированного доступа. Основная цель этих мер — максимально затруднить доступ злоумышленника к серверу и его службам, сделав стоимость такого вторжения выше стоимости хранимой на сервере информации.

Информационная безопасность подразумевает ограничение как физического доступа к самому серверу, так и доступа через Интернет и локальные сети к службам, запущенным на сервере. Доступ извне к самому серверу практически невозможен, так как он обычно находится на значительном удалении от злоумышленника и охраняется. Поэтому способы защиты от физического доступа в этой статье мы затрагивать не будем. Рассмотрим подробно другую сторону информационной безопасности — доступ к запущенным на сервере службам. Уровень безопасности зависит от ценности хранимой информации, поэтому построение защиты следует выбирать исходя из этого критерия. Нет смысла обеспечивать серьезную защиту на сервере, содержащем домашнюю страничку пользователя, но совершенно непростительно ограничиваться простейшими мерами для защиты корпоративного сервера.

Что выбрать и как настроить?

Для начала необходимо определить, какие данные будут храниться на сервере, и какие люди будут иметь доступ к этим данным. Если это веб-сервер компании, к которому имеет доступ множество людей по всему миру, необходимо открыть доступ для всех, но только на порты этого веб-сервера. Если это корпоративный сервер, где хранится жизненно важная для компании информация, его следует установить внутри корпоративной сети или в демилитаризованной зоне и обеспечить ограниченный доступ — например, по IP-адресу или с идентификацией пользователей. Устанавливая и настраивая программы, следует иметь в виду, что в любой программе присутствуют ошибки; и хотя чем сложнее программа, тем труднее их распознать, рано или поздно эти ошибки бу- »

» дут обнаружены. Поэтому, выбирая программное обеспечение, старайтесь устанавливать наиболее свежие версии, чтобы избежать использования программы, безопасность которой под вопросом. Перед установкой внимательно просмотрите в Makefile все возможные ключи при сборке. Некоторые порты выводят эти параметры после команды make; в таком случае при необходимости нажмите Ctrl-C и запустите сборку заново с необходимыми ключами. Если необходимо собрать порт со специфическими параметрами команды configure, эти параметры надо внести в переменную CONFIGURE_ARGS в Makefile порта. Настраивая службу, внимательно прочитайте документацию, особенно файл, в котором описаны изменения, внесенные в программу. Обращайте также внимание на конфигурацию доступа администратора: в любой доступный администратору интерфейс доступ по IP-адресу желательно ограничить средствами firewall или самой службы. Если вы хотите получать свежую информацию об обнаружении уязвимых мест в установленных на сервере службах, подпишитесь на соответствующие рассылки, посвященные информационной безопасности (например, на BugTraq, freebsd-security или русскоязычный вариант http://security.nnov.ru). При обнаружении «дыр» вносите исправления в конфигурацию службы или устанавливайте более свежую версию.

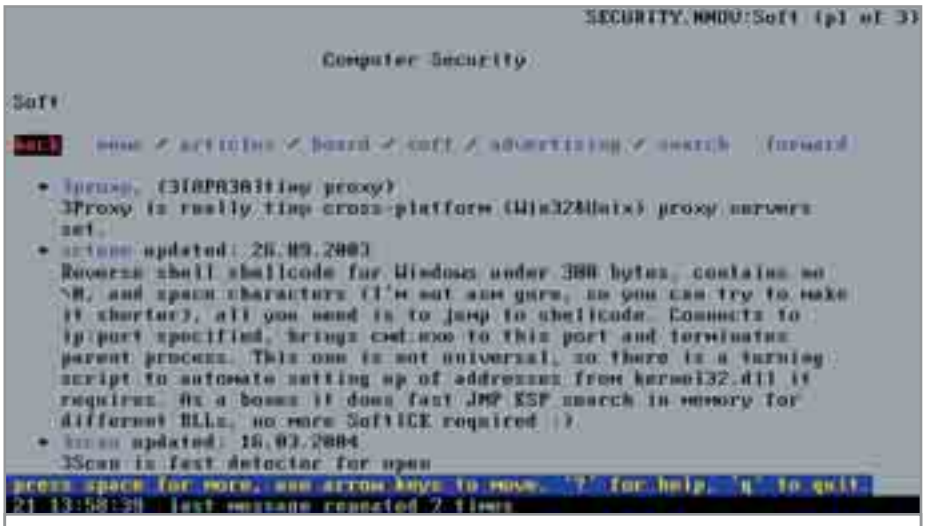
Выбранный пароль также влияет на безопасность соединения: слишком короткий или легко угадываемый, он позволит злоумышленнику гораздо быстрее получить доступ к ресурсам, предоставляемым для этой учетной записи. Достаточно безопасными сегодня считаются пароли длиной не менее восьми символов, в которых присутствуют цифры, заглавные и строчные буквы. Естественно, в качестве пароля не следует употреблять даты рождения, имена и прочие данные, которые могут быть явно связаны с вами.

Почтовые серверы

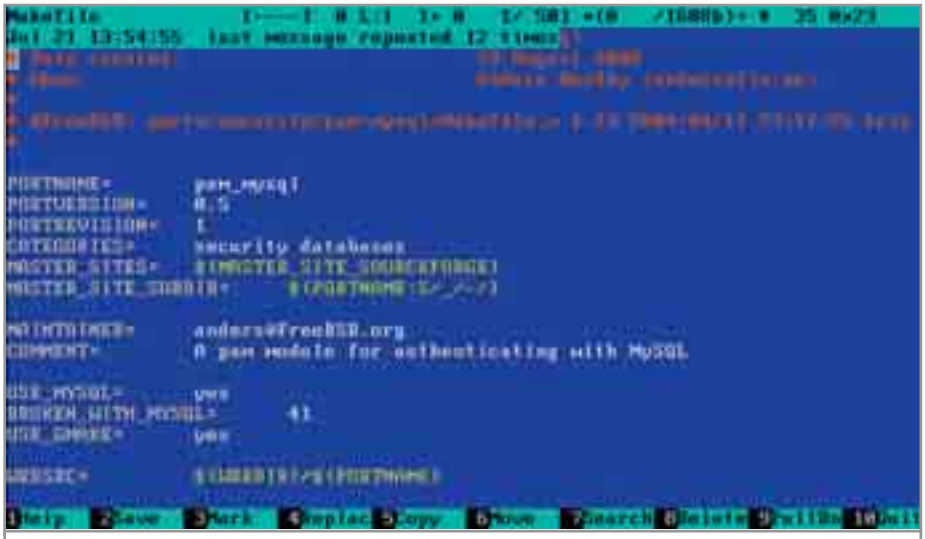
Протоколы обмена почтой очень уязвимы, так как вся информация, в том числе и учетные данные пользователя, передаются в открытом виде и могут быть перехвачены и модифицированы. Для предотвращения перехвата паролей и сообщений достаточно зашифровать весь трафик с помощью SSL. Если шифрование по каким-либо причинам невозможно, предотвратить перехват пароля можно, используя альтернативные методы авторизации, при которых передается не сам пароль, а его хеш (hash — уникальный идентификатор, однозначно идентифицирующий парольную фразу). К сожалению, подобные методы авторизации реализованы только в протоколах SMTP и IMAP4.

Наиболее простым способом зашифровать соединение остается использование программы stunnel (/usr/ports/security/stunnel). Она позволяет организовать прозрачный шифрованный туннель для программ, которые не поддерживают SSL. Программа самостоятельно устанавливает шифрованное соединение с клиентом и передает на сервер уже расшифрованные данные. Для ее работы необходим ключ, созданный при помощи утилиты openssl или полученный от центра сертификации, и сертификат, выписанный для этого ключа. На основе полученных ключа и сертификата будет производиться шифрование данных. Обратите внимание: поле CN сертификата должно совпадать с именем сервера, к которому вы будете обращаться. Если сервер называется mail.wormhole.ru, а сертификат в поле CN содержит wormhole.ru, то почтовый клиент будет выдавать ошибку несоответствия имени сервера и сертификата при каждой проверке почты. Для работы нужен расшифрованный ключ; если ключ был зашифрован паролем, его надо расшифровать при помощи программы openssl (openssl rsa –in зашифрованный_ключ –out расшифрованный_ключ). Полученный ключ и сертификат для удобства использования можно объединить в один файл, установив для этого файла владельца root и права доступа 440.

При компиляции программы настроек не требуется. После установки в конфигурационном файле /usr/local/etc/stunnel.conf необходимо прописать путь к созданному файлу ключа и сертификата в параметр cert, а также параметры для необходимых сервисов. В файле stunnel.conf уже присутствуют описания для зашифрованных версий протоколов SMTP, POP3 и IMAP4 (они называются соответственно ssmtp, pop3s и imaps) — необходимо только снять комментирование с соответствующих строк. Если сервер защищен firewall, организуйте соответствующие доступы для работы с шифрованными версиями протоколов (465, 995 и 993). Далее запустите stunnel при помощи /usr/local/etc/rc.d/stunnel.sh start. Если stunnel по какой-либо причине не запустился, посмотрите сообщение об ошибке в /var/log/messages: оно содержит в себе причину сбоя службы. При настройке почтового клиента обратите внимание, что »



Существует множество сайтов, предлагающих самую свежую информацию о найденных уязвимостях и последних обновлениях программ



При компиляции программ стоит обратить внимание на ключи в файле Makefile — это поможет избежать утомительной пересборки

» SMTP уже готов к работе с SSL, и команду starttls для него подавать не надо; кроме того, некоторые почтовые клиенты (например, TheBat или Sylpheed) требуют явно указать тип SSL-соединения.

Данный способ очень прост в реализации, но имеет серьезный недостаток: все входящие соединения в логах почтового сервера будут исходить от адреса 127.0.0.1, и определить реальный IP-адрес, с которого пришел клиент, можно будет только с помощью анализа обоих логов файла (лога почтового сервера и лога программы stunnel). От этого недостатка свободен способ реализации SSL- шифрования на самом почтовом сервере.

Для реализации SSL-шифрования в SMTP-сервере sendmail понадобится установить пакет cyrus-sasl2 (/usr/ports/security/cyrus-sasl2), а далее либо пересобрать sendmail из исходных кодов системы (если они установлены в /usr/src/), либо установить sendmail-sasl из портов (/usr/ports/mail/sendmail-sasl). Для первого способа в файл /etc/make.conf необходимо добавить следующие строки

```
SENDMAIL_CFLAGS=-I/usr/local/include/
    sasl –DSASL
SENDMAIL_LDFLAGS=-L/usr/local/lib
SENDMAIL_LDADD=-lsasl2
```

Перейдя в каталог /usr/src/usr.sbin/ sendmail, нужно выполнить команды:

```
make
make install
```

Для второго способа достаточно перейти в каталог порта и выполнить make install. После сборки и установки sendmail надо перейти в каталог /etc/mail, где хранятся конфигурационные файлы сервера, и добавить следующие строки в файл имя_сервера.mc (если же такого файла нет, надо выполнить команду make, которая создаст этот файл из прототипа freebsd.mc)

```
define(`confSERVER_CERT', `путь
до файла с ключом и сертификатом')
define(`confSERVER_KEY', `путь до файла с
    ключом и сертификатом')
```

После добавления строк выполните make и протестируйте полученный конфигурационный файл на наличие ошибок при помощи

```
sendmail –bt –C имя_сервера.cf.
```

Если все было сделано правильно, ошибок быть не должно. Далее выполните make install и make restart, чтобы установить конфигурационный файл и заставить почтовый сервер прочитать его. Для проверки правильности установки необходимо с помощью telnet зайти на порт сервера (telnet 127.0.0.1 25) и выполнить команду «ehlo test». Если сервер собран с поддержкой SASL и прочитал ключ и сертификат, он даст ответ STARTTLS; если такой строки нет, придется внимательно изучить лог-файл почтового сервера (/var/log/maillog) и определить, на каком этапе настройки допущена ошибка.

Для реализации шифрации на сервере POP3/IMAP необходима ее поддержка: это может быть qpopper (/usr/ports/ mail/qpopper, только pop3 и pop3s) или cyrus-imapd (/usr/ports/mail/cyrus-imapd2 и /usr/ports/mail/cyrus-imapd22). Оба сервера поддерживают SSL без дополнительных настроек при компиляции; для cyrus-imapd так же понадобится cyrus-sasl. После установки и сборки qpopper надо внести следующие изменения в файл конфигурации (/usr/ local/etc/qpopper.config): переменную tls-support установить в STLS или в alternate-port — первый параметр разрешает работу команды STLS, чтобы шифрованные и обычные соединения шли на один порт, а второй устанавливает для шифрованных соединений отдельный порт (995). Однако команду STLS надо использовать только в крайних случаях, поскольку ее поддерживают не все современные клиенты. Обычно достаточно значения alternate-port, а в переменной tls-identity-file нужно указать путь к файлу с ключом и сертификатом. После запуска qpoppper проверить правильность настройки можно с помощью команды sockstat, в которой должен быть открытый порт 995. Настройка cyrus-imapd практически не отличается от настройки qpopper. В его конфигурационном файле (/usr/local/etc/imapd.conf) надо указать в параметрах tls_cert_file и tls_key_file путь до файла с ключом и сертификатом, а также в /usr/local/etc/cyrus.conf снять комментирование службы pop3s и imaps. После запуска сервера правильность настройки проверяется командой telnet (telnet 127.0.0.1 995 и telnet 127.0.0.1 993). При правильной настройке telnet установит соединение, а в случае ошибки вернет сообщение о невозможности инициализировать SSL и разорвет соединение.

FTP-серверы

FTP-серверы так же плохо защищены от перехвата передаваемых данных, как и почтовые серверы. Логин и пароль пользо- »

» вателя передаются по протоколу в открытом виде и могут быть перехвачены. Зачастую пароль проверяется по системной базе данных, поэтому перехваченные учетные данные могут быть использованы для получения доступа к другим, более устойчивым к взлому службам, также использующим системную базу данных (например, для доступа по SSH). Поскольку стандарт FTP-протокола и существующие почтовые клиенты не поддерживают никаких методов шифрования данных и пароля, использование FTP-служб следует свести к минимуму и применять их только там, где это совершенно необходимо. Также следует активно применять ограничения доступа к FTP-серверу по IP-адресу и использовать механизм chroot, чтобы запретить доступ к другим каталогам, кроме рабочего. Для встроенного в систему FTP-сервера достаточно внести в файл /etc/hosts.allow строки вида «ftpd : 195.230.89.77 : allow». Если вместо allow написать deny, то для указанного IP-адреса или диапазона адресов доступ может быть закрыт. Также для ограничения доступа по IP-адресам может использоваться firewall. Если в качестве FTP-сервера используется proftpd (/usr/ ports/ftp/proftpd), то для ограничения доступа по IP-адресам следует применять имя сервиса «ftp» вместо «ftpd». Более того, сервер можно настроить на использование отдельной базы для авторизации пользователей, исключая возможность перехвата пароля для последующего доступа через защищенные службы. Для отдельной базы надо скомпилировать proftpd из портов с использованием ключа WITH_MYSQL и внести в конфигурационный файл (/usr/local/etc/proftpd.conf) следующие изменения:

```
SQLAuthTypes Plaintext
SQLAuthenticate users*
SQLConnectInfo BASE@127.0.0.1 LOGIN PASS
SQLDefaultGID 65534
SQLDefaultUID 65534
SQLMinUserGID 100
SQLMinUserUID 500
SQLUserInfo users userid passwd uid
      gid homedir shell
```

где BASE — это имя базы на MySQL-сервере, LOGIN и PASS — соответственно, логин и пароль для доступа к этой базе данных. Также надо создать на MySQL базу данных, а в ней таблицу users, и предоставить доступ к этой базе пользователю USER:

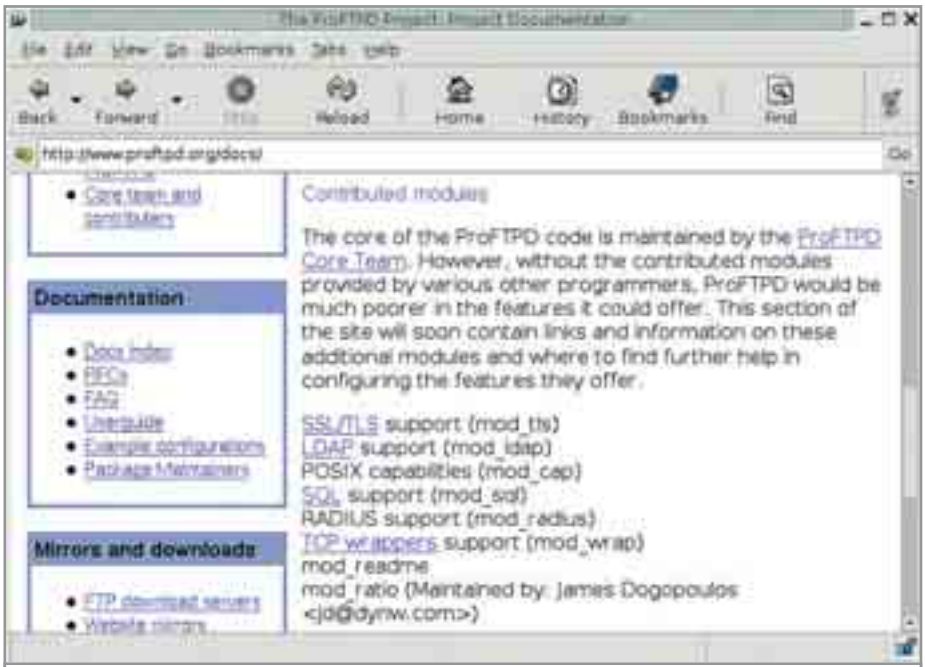
```
CREATE database BASE; // сделали базу
CREATE TABLE users (
 userid varchar(30) NOT NULL default '',
 passwd varchar(80) NOT NULL default '',
 uid int(11) default NULL,
 gid int(11) default NULL,
 homedir varchar(255) default NULL,
 shell varchar(255) default NULL,
 UNIQUE KEY userid (userid)
) TYPE=MyISAM; // создали таблицу
GRANT all on BASE.* to USER@127.0.0.1
      identified by PASS; // добавили
      пользователя
```

Ограничить разрешенный пользователю доступ рабочим каталогом (chroot) можно директивой «DefaultRoot ~»: в таком случае в proftpd корневым каталогом пользователя будет его рабочий каталог, и он не сможет увидеть каталоги других пользователей или системные каталоги. Можно также указать исключения из группы DefaultRoot, поставив после тильды «~» имя группы с восклицательным знаком «DefaultRoot ~ !stuff», — тогда пользователи, включенные в группу stuff, будут видеть все каталоги сервера, а всем остальным корневой каталог будет подменяться на рабочий, за пределы которого они не смогут выйти. Во встроенном FTP-сервере список пользователей, которым необходимо сменить корневой каталог на их рабочий, находится в файле /etc/ftpchroot (этого файла в системе изначально нет, и его надо создать).

Этим механизмы защиты FTP-сервера исчерпываются, поэтому использовать «открытый всему Интернету» FTP-сервер стоит только в соответствующих случаях — например при предоставлении услуг хостинга или публичного FTP-каталога. Во всех остальных случаях лучше ограничить круг IP-адресов, с которых будет осуществляться доступ, или заменить FTP на SFTP (это одна из программ, входящих в состав SSH, которая реализует функциональность FTP-сервера и FTP-клиента на базе шифрованного соединения).

HTTP-серверы

Самой востребованной службой в Интернете и сетях интранет является HTTP-сервер. Его применение чрезвычайно широко — от тривиального доступа к файлам до исполь- »



proftpd способен работать по защищенному SSL-соединению и поддерживать несколько расширений стандартной аутентификации

» зования в больших системах управления данных и реализации структуры «тонкого клиента». Поэтому большинство атак приходится именно на HTTP-серверы. Атаки на HTTP-сервер можно разделить на две части: атаки на программное обеспечение сервера, при которых используются уязвимости самого сервера, и атаки на содержимое, при которых уязвимости ищутся уже в коде выполняемых на этих серверах программ (например, CGI- и PHP-скрипты). Авторизация по методу basic, которая встречается на 99% сайтов, также недостаточно защищена от перехвата пароля, поскольку логин и пароль передаются открытым текстом: передаваемые от сервера к серверу данные не шифруются и могут быть перехвачены злоумышленником. Выбор того или иного метода защиты зависит от характера хранимых данных и их важности.

Защитить передаваемые данные от перехвата можно, используя вместо обычного HTTP-сервера его SSL-версию (/usr/ports/www/apache13-ssl; в apache2 нет SSL-реализации в системе портов). Настройка SSL-версии сервера ничем не отличается от обычной настройки — необходимо лишь указать в конфигурации сервера «SSLEngine on», а в переменных «SSLCertificateFile» и «SSLCertificateKeyFile» указать путь до сертификата и ключа. Запускать такой сервер надо командой apachectl startssl (если просто ввести start, то модуль для SSL не будет активирован). Поле CN-сертификата должно совпадать с именем сервера, иначе браузер будет сообщать о несоответствии полей при каждом обращении к странице. Поскольку создание и поддержка SSL-соединения требует некоторой части ресурсов процессора, SSL желательно использовать в случае острой необходимости: например, можно разделить сайт на две части, в одной из которых будет находиться общедоступная информация (без шифрования), а в другой — закрытая информация, зашифрованная при помощи SSL и недоступная для перехвата.

Для защиты самого сервера от атак есть только одно действенное решение — быстрое реагирование на сообщения о нарушении безопасности в используемых программах (apache, php, perl). Также необходимо своевременное обновление программного обеспечения серверов.

Одним из самых уязвимых мест в безопасности являются программы, запущенные пользователем на веб-сервере (PHP- и CGI-скрипты). Непрофессионально написанный скрипт становится источником проблем для сайта, а в худшем случае — для многих сайтов или сервера в целом. Например, если программа пользователя не проверяет переданный в параметрах путь к файлу, злоумышленник может открыть системные файлы или файлы из защищенной области этого сервера. Для защиты от подобного вида атак существует метод suexec и набор методов safe-mode. Первый метод (suexec) работает для CGI-скриптов: он ограничивает местоположение скрипта в файловой системе и привилегии, с которыми работает скрипт. Второй метод (safe-mode) является частью php и ограничивает функциональность PHP-скриптов.

Для включения suexec достаточно в конфигурации сайта указать директивами User и Group пользователя или группу, от имени которых будет запускаться CGI-скрипт.

Условия работы CGI-скрипта при включенном suexec следующие:

- ID пользователя должно быть не меньше 1000, что автоматически запрещает запуск программ от имени различных служб;
- владельцем и группой скрипта должны быть те владелец и группа, которые указываются директивами User и Group в конфигурации сервера;
- каталог, в котором расположен скрипт, должен принадлежать только указанному владельцу и группе;
- сам скрипт и каталог могут быть доступны для записи только владельцу и группе;
- скрипт должен находиться в каталоге указанном при настройке веб-сервера (обычно это /home или /home/htdocs).

К сожалению, сообщения об ошибке работы скрипта по вине самого скрипта »

Свой центр сертификации

Полезное самоуправство

Для создания защищенного SSL-соединения требуется ключ и сертификат на основе этого ключа, который подтверждает подлинность как самого ключа, так и отправителя. Получить сертификат на основе созданного SSL-ключа можно, официально купив сертификат у одного из общеизвестных доверенных центров сертификации (Verisign, Trawte и т. д.). Можно также самому создать центр сертификации, при помощи которого вы сможете самостоятельно выписать необходимое количество ключей и сертификатов. Полученный собственный центр сертификации ничем не будет отличаться от общеизвестных, за исключением того, что никто, кроме вас, не будет о нем знать.

Для организации доверительных отношений необходимо добавить на всех компьютерах корневой сертификат центра в список доверенных сертификатов. Такой собственный центр сертификации удобен в небольших компаниях, когда сертификаты используются в пределах фирмы, например, для шифрования почтовых сообщений и цифровой подписи.

Для создания центра сертификации воспользуемся входящим в состав openssl скриптом центра сертификации (этот скрипт находится в /usr/share/openssl/misc/CA.pl). Порядок создания центра сертификации таков:

- Создаем каталог /home/ca, в котором будут находиться файлы центра сертификации.
- Копируем в него CA.pl и выставляем ему права на исполнение.
- Вносим в /etc/ssl/openssl.cnf значения по умолчанию.
- Создаем корневой сертификат при помощи ./CA.pl –newca. Пароль, указанный при создании корневого сертификата, необходимо сохранить, поскольку при его потере выписать новый сертификат будет невозможно.

» или по вине suexec идентичны. Чтобы разобраться, кто из них создает ошибку, рекомендуется отключить suexec и попробовать выполнить скрипт. Если скрипт работает, то ошибку следует искать в настройках suexec; если нет — возможно, причина кроется в самом скрипте.

Следует отметить, что safe-mode распространяется лишь на запущенный скрипт. Сам скрипт может запускать любые приложения и получать доступ к любым данным на его уровне доступа (это имя, от которого работает apache, обычно WWW или nobody, или имя и группа, указанные для suexec). В отличие от suexec, safe-mode позволяет более гибко управлять ограничениями: например, можно заблокировать только запуск внешних программ или разрешить их запуск из определенного каталога. Можно ограничить область видимости диска каталогом сайта. Учитывая, что модуль php выполняет все PHP-скрипты от имени сервера apache, необходимость включения safe-mode очевидна: любой PHP-скрипт в обычном незащищенном режиме способен прочитать каталоги других сайтов, которые находятся на этом же компьютере, а при наличии разрешений на запись (каталог загрузки картинок, форум, гостевая книга) может также внести несанкционированные изменения в эти данные или удалить их.

Вернемся к критическим, с точки зрения безопасности, ошибкам в скриптах. Например, если ваш скрипт получает в качестве параметра имя файла, который надо показать, то при отсутствии проверки злоумышленник сможет получить любой файл, указав его полный путь по абсолютному имени (например, /etc/passwd) или по относительному (../../../../../../../etc/passwd). Поэтому скрипт, выполняющий подобные действия, должен уметь проверять каталог, из которого берутся файлы,и преобразовывать относительный путь в абсолютный, чтобы проверить, разрешено ли открыть запрашиваемый файл. Если программа принимает данные из формы, следует позаботиться о том, чтобы злоумышленник не смог вставить лишние HTML-теги. Например, при отсутствии проверки на теги, злоумышленник может вставить javascript, который позволит получить данные с компьютера пользователя или администратора. Особенно это актуально, если используются сессии: получив идентификатор сессии администратора, любой злоумышленник легко сможет стать таковым, даже не имея соответствующего логина и пароля. Поэтому, работая с сессиями, следует проверять IP-адрес, с которого изначально пришел пользователь, и отсекать попытки получить доступ к этой сессии с других IP-адресов.

Создавая на сайте интерфейс администратора, по возможности ограничьте доступ к нему по IP-адресам, чтобы уменьшить вероятность подбора пароля или входа с несанкционированного адреса с помощью перехваченного пароля. Защита по IP-адресам организуется при помощи инструкций allow и deny. HTTP-сервер apache реализует две политики поведения: «запрещено все, что не разрешено» и «разрешено все, что не запрещено», которые определяются ключевым словом «order». Политика «запрещено все, что не разрешено» (Order deny, allow) применяется при организации ресурсов, доступ к которым можно получить с определенных IP-адресов. Политика «разрешено все, что не запрещено» (Order allow, deny) применяется для предотвращения DOS-атак, приводящих к блокировке работоспособности. Списки хостов, которым предоставляется или закрывается доступ, указываются в ключевых словах Allow и Deny. В параметрах можно использовать как одиночные IP-адреса, так и сети, указывая маску сети после символа «/».

DNS-серверы

Серверы доменных имен, составляя основу функционирования сети Интернет, в большинстве случаев сами по себе не являются источниками проникновения в систему, но »

Теперь, когда центр сертификации создан, файл cacert.pem следует выложить в доступное место, откуда его смогут забрать пользователи, которые будут использовать выданные им сертификаты. Для удобства работы в Windows файлу надо присвоить расширение .crt; в этом случае сертификат будет автоматически опознаваться.

Для создания нового ключа и сертификата надо выполнить следующие действия:

► Создать ключ, запустив специальный скрипт ./CA.pl –newreq. После получения ответов на вопросы будет создан файл newreq.pem, содержащий закрытый паролем ключ и запрос на создание сертификата. Следует обратить внимание, что если сертификат будет использоваться для работы сервера, то поле Common Name (CN) должно содержать имя сервера или службы, для которой будет использоваться этот сертификат (например, mail.domain.ru).

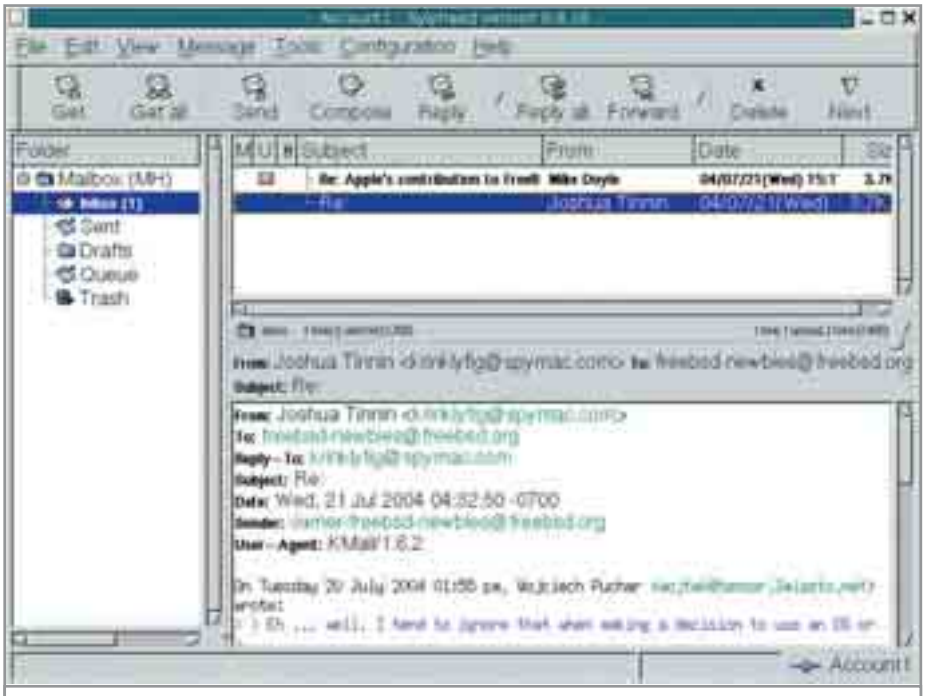
► После выполнения запроса надо создать сертификат. Для этого опять запустите скрипт ./CA.pl с параметром –sign и, введя пароль от корневого сертификата, создайте сертификат. Созданный сертификат будет находиться в файле /etc/ssl/newcert.pem.

Внутри созданного сертификата содержатся сигнатуры в формате Netscape (компании, разработавшей этот открытый протокол шифрования), которые следует удалить, так как большинство систем не поддерживают эти заголовки и не могут загружать сертификат. Ключ, полученный на первом шаге, находится в зашифрованном виде; некоторые серверы требуют, чтобы ключ был расшифрован. Расшифровать ключ можно при помощи следующей команды «openssl rsa –in enc.key –out dec.key». Ключ будет расшифрован и готов для использования разными службами.

» могут использоваться злоумышленниками для получения информации о конфигурации сети и серверов.

По умолчанию DNS-сервер запускается от имени пользователя root и имеет доступ ко всей файловой системе. В случае взлома сервера злоумышленник получает полный контроль над сервером и может использовать компьютер в своих целях. Чтобы предотвратить это, следует запускать DNS-сервер в chroot–окружении и не от имени пользователя root. Для выполнения этих условий необходимо в /etc/rc.conf добавить параметр named_flags в котором надо указать следующие параметры «-u bind –t /etc/namedb –w /». Если на сервере находится вторичный сервер имен (Secondary DNS), то каталог, в котором сервер хранит загруженные с первичного сервера файлы зон должен быть доступен на запись пользователю bind.

Для предотвращения возможности получения информации о конфигурации сети и серверов надо удалить из файлов конфигурации зоны все записи кроме SOA, A, PTR, NS и CNAME, которые используются для работы DNS-сервера. Также необходимо запретить передачу файла зоны всем, кроме вторичных серверов, так как получив файл зоны, злоумышленник узнает имена и IP-адреса всех серверов, перечисленных в этой зоне. Для этого в named.conf необходимо добавить параметр «allow-transfer { ip1; ip2; ip3; };» где ip1, ip2 и ip3 это IP-адреса вторичных серверов. На вторичных серверах следует запретить передачу зоны указав «allow-transfer { none; };». Если сервер используется только для поддержки DNS–зон в Интернете, то на нем для экономии трафика следует отключить рекурсивные запросы, добавив параметр «recursion no;» в named.conf. Если сервер обслуживает локальную сеть, рекурсивные запросы надо разрешить только для этой сети; сделать это можно при помощи параметра «allow-recursion { net/mask; };» (например, allow-recursion { 192.168.123.0/255.255.255.0; };). Если запрашивать записи из зоны должны только из определенной сети или сетей (например сервер локальной сети), то при помощи параметра «allow-query { net/mask; };» следует задать сеть, для которой разрешены запросы (по умолчанию запросы разрешены отовсюду). Если DNS-сервер обслуживает и локальную сеть и Интернет, то в зоне, смотрящей в Интернет, нежелательно будет указывать имена и адреса компьютеров, находящихся в локальной сети. Лучше разнести DNS-серверы локальной сети и Интернета на разные компьютеры или сделать отдельную зону для внутренней сети (например, для домена domain.ru зона может выглядеть как office.domain.ru) и разрешить запросы только из локальной сети (при помощи allow-query). Подобная настройка DNS–сервера позволит усилить защищенность серверов компании в том случае, если злоумышленник попытается получить информацию о сети и серверах.



Почтовый клиент Sylpheed требует явно указывать тип SSL-соединения

Защищенное окружение программ

Зачастую для усиления эффекта защиты применяется окружение chroot или jail. Использование этих методов усиливает защиту данных, которые предоставляются сервером, а также позволяет противостоять попыткам вывести из строя весь сервер и получить доступ к другим службам, работающим на этом сервере. Принцип действия обоих методов идентичен: они реализуют защищенное окружение вокруг работающего процесса или процессов.

Метод chroot меняет корневой каталог (/) на указанный перед запуском программы, так что злоумышленник, взломав программу, работающую в таком окружении, сможет получить доступ только к тем службам и данным, которые были запущены в этом окружении. Но этот метод имеет и свои минусы: большинство программ скомпилировано с использованием динамических библиотек (/usr/lib* и /usr/local/lib*), а chroot подменяет корневой каталог до запуска программы; следовательно, программа, не обнаружив необходимых библиотек, просто не станет работать. Чтобы избежать этого, необходимо либо собирать программу без использования динамических библиотек, либо скопировать необходимые библиотеки в рабочий каталог программы, создав там аналогичную структуру каталогов с библиотеками. Получить список используемых программой библиотек можно при помощи команды «ldd имя_программы». Вам могут понадобиться и некоторые другие данные и программы для нормальной работы службы chroot, но все они подбираются лишь опытным путем в процессе »

» тестирования окружения. Однако chroot защищает только файловую систему, оставляя незащищенными память, сетевые соединения и системные вызовы. Например, злоумышленник может, получив доступ в систему, модифицировать правила firewall или загрузить программу, которая будет сканировать сеть или перехватывать пароли и данные.

От вышеуказанных недостатков свободен метод jail, который организует защищенное окружение, подобно chroot, но при этом защищает память, сетевые интерфейсы и системные вызовы. Правда, для каждого jail-окружения требуется отдельный IP-адрес, который используется этим окружением. Поэтому jail чаще используется для создания виртуального компьютера FreeBSD, в котором работают службы. Например, можно, используя мощный компьютер, запустить на нем несколько одновременно работающих различных серверов (почтовый сервер, HTTP/FTP-сервер, сервер доступа), которые смогут взаимодействовать друг с другом не иначе как через сеть. Дисковое и адресное пространство такого компьютера будет разделено, но при этом возможное проникновение в одно из работающих jail-окружений никак не повлияет на работоспособность всех остальных: системные вызовы, способные изменить состояние всей системы, будут блокированы, а значит, даже пользователь с правами root внутри такого окружения не сможет внести изменения в правила firewall, «прослушать» сеть или отформатировать диски. Если jail используется как расширенный вариант chroot, его настройка производится аналогично chroot-окружению. Формат запуска jail следующий: «jail “полный путь до рабочего каталога” “имя сервера” “ip-адрес jail” “команда или программа”». Если jail используется для организации виртуального компьютера, для его настройки понадобится исходный код системы (находится в /usr/src). Для настройки jail надо выполнить следующие шаги:

► Вручную создать каталог в котором будет находиться jail (например, /home/jail)
► Перейти в директорию /usr/src
► Выполнить «make world DESTDIR=/home/jail»
► Выполнить «cd etc ; make distribution DESTDIR=/home/jail»
► Скопировать /stand/sysinstall в /home/jail/sbin/sysinstall
► Выполнить «jail /home/jail jail_HOSTNAME jail_IP /bin/sh» (на этом шаге будет запущен jail в режиме конфигурирования)
► Создать пустой файл /etc/fstab
► Перейти в /etc/mail и выполнить там команду «make»
► В /etc/rc.conf добавить строчку network_interfaces=""
► В /etc/resolv.conf занести настройки DNS
► Запустить /sbin/sysinstall и произвести настройки временной зоны.
► Установить пароль пользователя root и завести необходимое количество пользователей.

Все, Jail настроен. Для его запуска надо выполнить команду «jail /home/jail jail_HOSTNAME jail_IP /bin/sh /etc/rc», после чего виртуальный компьютер будет запущен и на него можно будет зайти по защищенному SSH-туннелю, чтобы выполнить дальнейшую конфигурацию и установку необходимых программ.

■ ■ ■ Николай Толкачев

...# Воссоздание мира/_...

Компилирование ОС из исходных кодов

Сегодняшний мир живет по «интернет-времени»: защита системы может быть взломана хакерами через несколько часов после публикации сведений о ее потенциальной уязвимости. Поэтому любая операционная система непрерывно развивается, и пользователи должны постоянно поддерживать ее в актуальном состоянии.

Никто из производителей операционных систем не отрицает этой необходимости. Microsoft выпускает периодические обновления в виде Service Pack для пользователей Windows. Обновления для индивидуальных приложений Windows и Macintosh появляются в течение нескольких дней с момента обнаружения каких-либо проблем. Подобный механизм обновлений реализован и в Linux. FreeBSD также позволяет осуществлять поддержку похожим способом. Но поскольку ее пользователи имеют доступ к исходному коду ОС, поддержка возможна и с помощью более удобных методов.

Основные версии FreeBSD (4.0, 5.0) выходят с интервалом в один-два года; промежуточные (3.4, 4.1.1) — как правило, через три-шесть месяцев. Но как бы часто ни появлялись обновления, с точки зрения администратора, отвечающего за защиту системы, это все равно случается слишком редко. Чтобы обновлять систему «в реальном времени», вы можете воспользоваться способами, которые обеспечивает FreeBSD: CVSup и make world.

Отслеживание исходного кода FreeBSD

Простейший и наиболее распространенный способ обновления — ожидание новой официальной версии системы. Однако это слишком медленное решение. Чтобы идти в ногу со временем, необходимо следить за исходными кодами системы и приложений.

Дерево исходных файлов ОС FreeBSD содержится в хранилище CVS (CVS repository), представленном на нескольких зеркальных серверах. С помощью определенного инструмента это хранилище можно синхронизировать с локальным. »

Для cvsup существует также графический интерфейс, доступный в коллекции портов

Две ветви исходного кода

Исходные коды ОС FreeBSD находятся в состоянии постоянного изменения. Многочисленные разработчики исправляют найденные ошибки, ликвидируют уязвимости, обновляют утилиты и добавляют новые возможности. Параллельно поддерживаются несколько ветвей дерева исходных кодов. И только две из них постоянно находятся в активной разработке — ветви Current (текущая) и Stable (стабильная).

Ветвь Current является наиболее современной версией исходных кодов системы. Она предназначена главным образом для разработчиков и бета-тестеров, а также для тех, кому немедленно, причем без всяких альтернатив, нужны новые свойства системы: поддержка каких-либо устройств или программные обновления.

В определенный момент разработки (обычно после первого релиза операционной системы) версия Current начинает считаться стабильной. После этого этапа ветви дерева исходных кодов сдвигаются вверх: теперь Current обозначает самую новую, верхнюю ветвь, а Stable — бывшую ветвь Current.

Прежде чем собрать систему

Необходимо отметить, что сборка системы из исходных кодов, особенно на активно работающем сервере, может быть чревата уничтожением существующей системы, утерей каких-либо файлов или даже крахом файловой системы. Поэтому в первую очередь перед обновлением системы необходимо произвести ее резервное копирование.

Одним из давно проверенных способов использования CVS для синхронизации является установка портированного приложения cvs-without-gui:

```
# cd /usr/ports/net/cvsup-without-gui
# make install clean
```

Затем необходимо подготовиться к настройке синхронизации исходного кода.

Решение проблем

Налаживаем работу CVSup

► CVSup не может соединиться с выбранным сервером. Выдается сообщение Connection refused — «В соединении отказано». Для решения этой проблемы выберите другой сервер CVS в конфигурационном файле /etc/cvsupfile. Чем выше его номер (например, cvsup5.ru.freebsd.org), тем меньше он должен быть загружен.

► CVSup соединяется, но ничего не происходит. Возможно, ваше соединение с сетью защищено firewall. Убедитесь, что он пропускает соединение по порту 5999, который использует CVSup.

► После синхронизации полностью удалился каталог /usr/src!. Это может произойти, если вы указали неверный дескриптор в файле /etc/cvsupfile. Дело в том, что при соединении сервер просто возвращает содержимое дерева CVS для выбранной ветви, и если последняя задана на несуществующим значением, то и результата вы не получите. Исправьте дескриптор и запустите cvsup заново.

Настройка эта производится один раз и впоследствии выполняется автоматически. Для обновления исходных кодов системы с помощью CVSup необходимо создать файл /etc/cvsupfile и в текстовом редакторе внести в него примерно следующее:

```
*default
  host=cvsup1.ru.freebsd.org
  (это русское зеркало основного дерева)
*default          base=/usr
*default          prefix=/usr
*default          release=cvs
*default          tag=RELENG_4
*default          delete use-rel-suffix
src-all
*default          tag=.
ports-all
doc-all
```

Строку с дескриптором версии нужно изменить в зависимости от вашей системы и от желания получить исходные коды определенной ветви. Например, получить исходные коды для версии 4.10-RELEASE позволяет следующая строка:

```
*default    tag=RELENG_4_10_0_RELEASE
```

Теперь исходные коды системы выбранной ветви можно получить простой командой:

```
# cvsup -g -L 2 /etc/cvsupfile
```

после отработки которой вы получите самую свежую версию исходных кодов.

Опции для сборки

Файл /etc/make.conf является глобальным конфигурационным файлом, который управляет всеми действиями Make World. Свежеустановленная система FreeBSD не имеет такого файла, однако в ней существует файл примера /etc/defaults/make.conf, в который включены все возможные настройки по умолчанию и который используется при отсутствии файла /etc/make.conf. Обычно в нем можно оставить все как есть, но в некоторых случаях придется изменить некоторые параметры — это позволит значительно ускорить работу или тонко настроить процесс сборки системы. »

» Для начала создадим файл /etc/make.conf (если он еще не существует):

```
# touch /etc/make.conf
```

Теперь внесем в него некоторые параметры, рекомендуемые FreeBSD handbook:

```
CFLAGS= -O –pipe
NOPROFILE=true
```

Можно также добавить некоторые опции, которые исключат часть не используемых вами приложений:

```
NO_I4B=true # если не работаете с ISDN
NO_LPR=true # если нет принтера
NOGAMES=true # если не до игрушек
NOUUCP=true # убираем устаревшее uucp
NO_MODULES=true # не собирать модули
      вместе с ядром
```

В этот же файл можно добавить правила и условия cvsup для более удобной синхронизации исходных кодов (после этого обновление системы можно будет произвести путем перехода в /usr/src и запуска команды make update):

```
SUP_UPDATE=   yes
SUP=          /usr/local/bin/cvsup
SUPFLAGS=     -g -L 2
SUPHOST=      cvsup5.ru.freebsd.org
SUPFILE=      /etc/stable-supfile
PORTSSUPFILE= /etc/ports-supfile
```

Здесь мы разделили один конфигурационный файл cvsup на два — для правил по синхронизации исходных кодов систем и портированных приложений. Содержимое этих конфигурационных файлов может быть, например, таким:

```
/etc/stable-supfile
*default host=cvsup5.ru.FreeBSD.org
*default base=/usr
*default prefix=/usr
*default release=cvs tag=RELENG_4_10
*default delete use-rel-suffix
*default compress
src-all
```

```
/etc/ports-supfile
*default host=cvsup5.ru.freebsd.org
*default base=/usr
*default prefix=/usr
*default release=cvs tag=.
*default delete use-rel-suffix
*default compress
ports-all
```

Сборка системы из исходных кодов

Сборка всей системы после обновления исходных кодов производится с помощью make world. Она обеспечивает необходимые меры предосторожности, которые обычно недоступны при установке бинарных (скомпилированных) файлов с компакт-диска. Например, если возникнут какие-то проблемы с исходным кодом, это выяснится уже на этапе компиляции, а следовательно, обновленная и инсталлированная система не станет в некоторый момент неработоспособной.

Сборка системы с помощью make world состоит из четырех основных этапов: сборка и инсталляция приложений основной системы, затем сборка и инсталляция ядра.

```
make buildworld
make installworld
make buildkernel
make installkernel
```

Теперь необходимо очистить каталог /usr/obj. Этот шаг не требуется, если команда make world выполняется в первый раз. Если это не так, из каталога /usr/obj необходимо удалить все файлы. Это действие ускорит процесс, а также предотвратит возможные конфликты.

В каталоге /usr/obj хранятся объектные файлы, используемые до сборки. Их удаление не всегда можно выполнить командой rm –rf, так как при сборке системы создаются файлы с установленным флагом schg («системно неизменяемый»), которые не доступны для удаления даже пользователю root. Эта мера предосторожности направлена на устранение таких случайностей, как, например, rm –rf .*.

Чтобы корректно очистить этот каталог, воспользуйтесь следующей последовательностью команд:

```
# cd /usr/obj
# chflags –R noschg *
# rm –rf *
make buildworld
```

Теперь все готово к сборке системы. Перейдите в каталог /usr/src и начните выполнение первого этапа make:

```
# make buildworld
```

Процесс buildworld занимает несколько часов (в зависимости от оборудования), поэтому не запускайте его, если у вас мало времени. Немного ускорить его можно с помощью опции –j4, запускающей одновременно четыре процесса. На машине же с несколькими процессорами можно добиться еще более высокой производительности, используя значение опции до 10.

Процесс сборки рекурсивно проходит по каталогу /usr/src в алфавитном порядке. Как только вы увидите, что компилируется нечто вроде /usr/src/usr.sbin, — можно считать, что процесс близок к завершению.

Обновление ядра

Если ваша система работает с ядром Generic (по умолчанию), процесс сборки ядра достаточно прост:

```
# make buildkernel
# make installkernel
```

Если же наименования ядра отличаются от заданного по умолчанию, процесс перекомпиляции выглядит несколько сложнее:

```
# make buildkernel KERNCONF=core
# make installkernel KERNCONF=core
```

Естественно, слово core необходимо заменить именем используемого вами ядра.

После сборки и установки ядра необходимо перезагрузиться, чтобы сделанные изменения вступили в силу.

Инспектор Mergemaster

Остался последний шаг — включить в иерархию /etc (и некоторые другие области, например, /usr/share) новые версии конфи- »

» гурационных файлов. Сам процесс make world не вносит изменения в /etc, чтобы все ранее настроенные конфигурационные файлы не были утеряны. Процесс объединения файлов из каталога /etc с их новыми версиями происходит с помощью стандартной утилиты mergemaster.

Поэтому рекомендуется выполнить резервное копирование каталога /etc. Это можно сделать, например, командой:

```
# cp –Rp /etc /etc.old
```

Далее запускаем mergemaster с параметром –p (перед сборкой системы):

```
# mergemaster –p
```

Вначале mergemaster создает временный каталог /var/tmp/temproot и устанавливает в него все необходимые файлы из исходных кодов. Программа показывает список файлов, находящихся в /etc и отсутствующих в выбранном временном каталоге. Затем идет сравнение этих файлов. При нахождении несовпадений на экране отображается вывод команды diff; при этом используется утилита постраничного просмотра, установленная в переменной Pager, или установленная по умолчанию more. При прокрутке к нижней части вывода diff программа запрашивает, какие действия произвести с новым файлом. По умолчанию утилита mergemaster не производит никаких действий, оставляя файл в каталоге /var/tmp/temproot для последующего внесения изменений вручную.

Если выбрать команду m для объединения двух файлов, программа перейдет в режим sdiff. В нем построчно будут показаны две версии измененного файла, что позволит выбрать нужную информацию из старого или нового файлов.

Иногда строки выглядят одинаково, поскольку различие просто не поместилось на экране. В таком случае mergemaster можно запустить с опцией –w, которая устанавливает ширину экрана:

```
# mergemaster –p –w 120
```

После окончания сравнения всех файлов утилита mergemaster запрашивает подтверждение на удаление всего содержимого каталога /var/tmp/temproot. Если вы оставили несколько файлов для дальнейшего ручного вмешательства, выберите “No”.

Последним этапом производится установка новой, собранной системы:

```
# make installworld
```

После этого необходимо еще раз запустим утилиту mergemaster, но уже без параметров. После ее отработки можно будет перезагрузить систему.

Перезагрузка системы после обновления

Теперь задайте себе ряд вопросов:

- Произвели ли вы синхронизацию системы с последней версией исходных кодов?
- Выполнили ли вы buildworld?
- Скомпилировали ли ядро?
- Выполнили ли installworld?
- Работают ли такие утилиты, как ps и top?
- Правильно ли прошло объединение файлов из каталога /etc с их новыми версиями?

Если ответом на все вопросы будет «да», систему можно перезагрузить. Теперь проверьте номер версии системы с помощью команды uname:

```
# uname -a
```

которая выдаст номер вашей версии FreeBSD, номер так называемого «патчлевела», а также имя собранного ядра, например:

```
FreeBSD core.firma.ru 4.10-RELEASE-p23
    FreeBSD 4.10-RELEASE-p23 #8: Mon
    Jun 14 16:41:41 KRAST 2004
    root@core.firma.ru:/usr/obj/usr/src/
    sys/core i386
```

Обновление портированных приложений

Не секрет, что в приложениях, установленных из системы портов ОС FreeBSD, со временем тоже могут найтись уязвимости. Приложения могут получить дополнительные функции, которые вам могут понадобиться. Можно обновлять приложения путем удаления старых и установки новых, но существует более простой и удобный метод — воспользоваться portupgrade, одной из портированных утилит. Установим ее:

```
# cd /usr/ports/sysutils/portupgrade
# make install clean
```

Пользоваться этой утилитой очень просто. Допустим, нам необходимо обновить приложение Midnight Commander:

```
# portupgrade mc
```

Приложение будет обновлено в соответствии с последней версией, указанной в системе портов. Утилита portupgrade имеет еще несколько полезных опций, ознакомиться с которыми можно на справочных страницах man portupgrade. ■ ■ ■ Александр Соловков

Альтернативное обновление

Заплатку – почтой

В некоторых случаях гораздо удобнее будет не обновлять систему целиком из исходных кодов, а отслеживать появившиеся изменения, подписавшись на рассылку freebsd-announce@FreeBSD.org. При наличии изменений вы получаете почтовое сообщение с указанием обнаруженной уязвимости, рекомендациями по ее устранению, а в случае необходимости — ссылки на нужный патч («заплатку»). В последнем случае необходимо будет скачать патч и запустить следующую команду для его наложения:

```
# cd /usr/src
# patch -p0 < /path/to/patch
```

где /path/to/patch — имя патча с указанием полного пути к нему. После наложения «заплатки» необходимо пересобрать систему следуя приведенным в статье рекомендациям.

Старый друг/_

Команды администрирования FreeBSD

Каждый пользователь FreeBSD из сектора SOHO — администратор своей машины, а среди пользовательских интерфейсов Unix-систем по-прежнему царствует командная строка. Поэтому, если FreeBSD — ваш выбор, вам не избежать знакомства с консольными командами администрирования.

Операционные системы семейства Unix, крайне популярные как серверные и технологические, в последнее время устремились в область SOHO. FreeBSD не отстает от своих собратьев: в погоне за лояльностью конечных пользователей до предела упростился процесс установки; адекватно пополняется список поддерживаемого оборудования и программного обеспечения; при загрузке компьютера автоматически запускается графический интерфейс. Но хотя удобный GUI становится нормой для Unix-подобных операционных систем, без командной строки по-прежнему не обойтись. Графические приложения являются лишь скромным дополнением к консольным утилитам — ведь, как и прежде, любую задачу во FreeBSD можно решить с помощью командной строки, не прибегая к помощи оконных менеджеров. В некоторых случаях без консоли вообще нельзя, так как многие Unix-программы до сих пор так и не обзавелись графическим интерфейсом. То же самое можно сказать о работе с другими машинами по сети: несмотря на наличие утилит управления удаленным графическим рабочим столом (к примеру, VNC), работа с консолью оказывается проще, быстрее, а при использовании коммерческих каналов передачи данных — еще и дешевле для пользователя. Не стоит забывать также о том, что существенным отличием FreeBSD от операционных систем компании Microsoft по-прежнему является статус владельца машины. В мире Unix он не просто пользователь, он — администратор системы. То, что операционные системы семейства Windows пытаются делать сами, ориентируясь на потребности «идеального среднестатистического пользователя», в расчете на которого они и были созданы, во FreeBSD пользователь-администратор кон- »

» фигурирует лично для себя, с учетом своих собственных потребностей и задач, а также нужд своих пользователей (если таковые, конечно, имеются).

Итак, во FreeBSD по-прежнему не обойтись без знания консольных команд и в первую очередь — консольных утилит для администрирования системы. Рассмотрим стандартные задачи администратора и консольные средства, которые помогут ему в решении возникающих проблем.

Справка прежде всего

Даже самый опытный Unix-администратор не в состоянии запомнить синтаксис всех консольных команд. Обычному пользователю достаточно владеть лишь базовыми консольными командами, чтобы никогда не «потеряться» в командной строке, — нужно лишь уметь пользоваться справочной системой FreeBSD. Ее главная программа — утилита man. Это, пожалуй, самая популярная команда у начинающих пользователей FreeBSD, однако ею не гнушаются и «гуру» операционных систем Unix. Указывая в качестве аргумента этой программе названия утилит и программ, можно получить необходимую справочную информацию о них. Например, следующая команда выведет на экран справку о программе pwd:

```
$ man pwd
```

Часть справочных материалов хранится в формате гипертекстовой справки GNU info. В man-справке таких программ прямо дается рекомендация воспользоваться справочной системой info, указывая в качестве аргумента название программы, о которой необходимо получить справку. Например, для получения гипертекстовой справки о редакторе emacs необходимо ввести следующую команду

```
$ info emacs
```

Зачастую пользователь не знает, какая конкретно программа нужна для решения его задачи. В этом случае на помощь придет команда apropos, которая производит поиск среди описаний файлов справки по ключевому слову, передаваемому этой команде в качестве аргумента. Например, чтобы получить список страниц справки о программах, которые так или иначе связаны с процессами операционной системы FreeBSD, необходимо ввести:

```
$ apropos process
```

Программа whatis позволяет узнать, для чего нужна та или иная команда, выведя ее краткое описание. Вот таким образом, например, можно узнать, что делает каждая программа из директории /usr/bin:

```
$ cd /usr/bin; whatis *
```

Управление консолями

FreeBSD — многопользовательская многозадачная операционная система; следовательно, с ней могут работать одновременно несколько пользователей, которые будут одновременно запускать различные процессы. Наличие нескольких виртуальных консолей на одной физической машине значительно упрощает решение таких задач. По умолчанию FreeBSD загружается с восемью виртуальными консолями. Переключаться между ними можно при помощи комбинаций клавиш Alt-F1, Alt-F2 и так далее до Alt-F8. Виртуальные консоли настраиваются в файле /etc/ttys, где они определены ключевыми словами ttyvX. Отредактировать этот конфигурационный файл из режима суперпользователя можно несложной командой

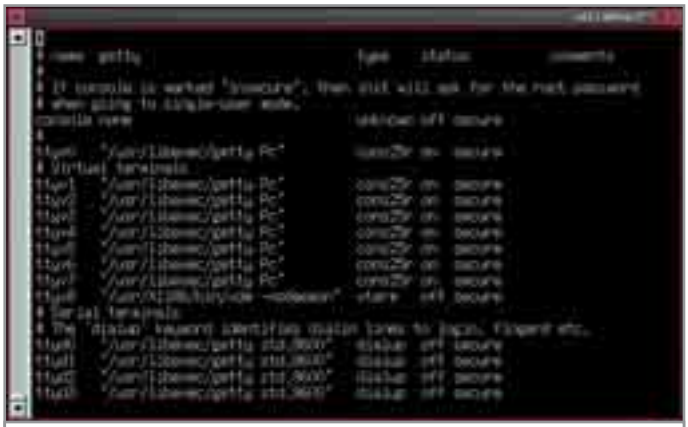
```
# ee /etc/ttys
```

С помощью утилиты kbdcontrol настраиваются различные параметры работы драйвера клавиатуры в виртуальной консоли. Можно, например, изменить скорость вывода символов при удержании одной клавиши, выключить служебный звуковой сигнал или сменить раскладку клавиатуры. Такая команда отключает надоедливую «пищалку»:

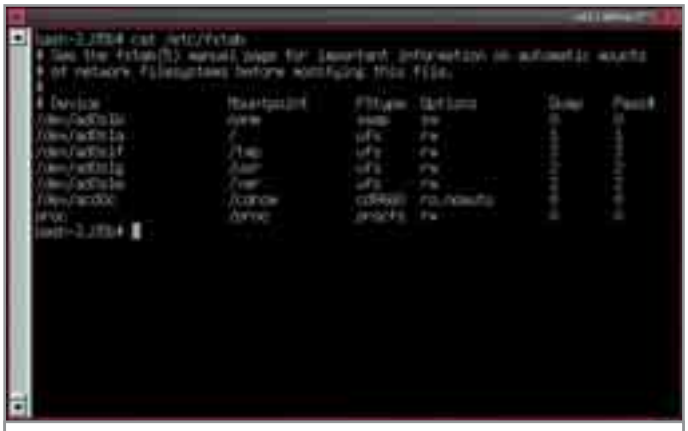
```
# kbdcontrol -b off
```

Утилита vidcontrol отвечает за конфигурацию параметров стандартного вывода консоли. С ее помощью можно, к примеру, сменить видеорежим со стандартного 80x25 символов на любой другой, изменить цвет фона и шрифта консоли, сменить сам шрифт, сделать скриншот экрана консоли и многое другое… Следующая команда устанавливает видеорежим 80x50 символов и зеленый цвет шрифта консоли: »

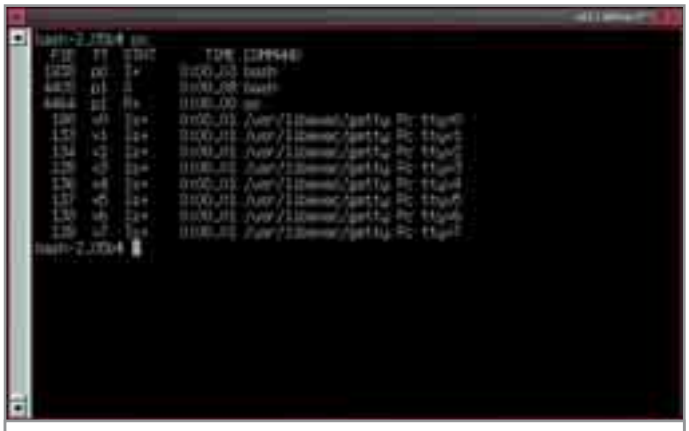
Справочник man — скорая помощь в командной строке



Стандартное определение виртуальных консолей во FreeBSD



Параметры накопителей хранятся в файле /etc/fstab



Команда ps выводит список процессов в оперативной памяти

```
# vidcontrol 80x50 green
```

А вот так можно посмотреть, что творится на третьей виртуальной консоли:

```
# vidcontrol -P </dev/ttyv3
```

Утилиты файловой системы

Во FreeBSD перед тем, как получить доступ к информации на жестких дисках, CD, DVD или любых других носителях, их необходимо сначала смонтировать в определенной точке монтирования в дереве файлов. Обычно локальные разделы монтируются операционной системой автоматически при загрузке. Для этого FreeBSD использует конфигурационный файл /etc/fstab, в котором указаны типы файловых систем на устройствах, точки и параметры их монтирования.

Для монтирования устройств в ручном режиме используется команда mount. В аргументах данной команде необходимо указать имя монтируемого устройства, тип файловой системы на нем, точку монтирования, а при необходимости также дополнительные параметры. Например, можно смонтировать носитель информации в режиме «только для чтения», чтобы избежать ошибочного изменения данных на нем. Следующая команда монтирует дискету в точке /tmp/floppy:

```
# mount -t msdos /dev/fd0 /tmp/floppy
```

Чтобы корректно и без потери данных извлечь устройство из компьютера, его необходимо первоначально размонтировать командой umount, указав в качестве параметра точку монтирования. Операционная система сбросит на устройство всю несохраненную информацию из кеш-памяти и, если есть такая возможность, остановит работу устройства.

```
# umount /tmp/floppy
```

Для администрирования разделов (слайсов) жесткого диска используется программа fdisk. С ее помощью можно получить информацию о существующих разделах на жестком диске, создать и удалить их. Программа fdisk без параметров отображает информацию о конфигурации текущего жесткого диска:

```
# fdisk
```

После создания раздела необходимо сначала присвоить ему метку командой bsdlabel. Затем утилитой newfs можно будет создать на нем новую файловую систему типа UFS. Если в качестве параметров команде bsdlabel указать только имя диска, будет выведена информация о текущей метке на разделе:

```
# bsdlabel /dev/da0s1
```

Утилита newfs имеет множество опций командой строки, позволяющих изменять параметры по умолчанию создаваемых файловых систем. Если в аргументах утилите указать только имя раздела, файловая система будет создана со стандартными параметрами:

```
# newfs /dev/da0s1
```

Часть параметров, которые можно было указать при создании файловой системы, можно будет изменить в дальнейшем с помощью команды tunefs. Запуская tunefs с ключом –p, пользователь получит текущие значения параметров, доступных для изменения:

```
# tunefs -p
```

Получить информацию обо всех параметрах файловых систем на разделах жесткого диска во FreeBSD можно с помощью команды dumpfs.

```
# dumpfs /dev/da0s1
```

Утилита growfs позволяет увеличить размер раздела файловой системы. Следующая команда увеличит партицию /dev/vinum/testvol до двух гигабайт (размер раздела задается в блоках):

```
# growfs -s 4194304 /dev/vinum/testvol
```

Для проверки файловой системы на наличие ошибок используется утилита fsck. Это основной инструмент поддержания файловых систем FreeBSD в рабочем состоянии. Без параметров fsck производит проверку разделов, указанных в файле /etc/fstab.

```
# fsck
```

Команда df выводит на экран размер каждого из смонтированных в системе раз-

» делов, объем свободного места и общий размер файлов на каждом из них.

```
$ df -h
```

Похожие функции выполняет утилита du: с ее помощью можно подсчитать, сколько места занимает та или иная директория на носителе или каждая из поддиректорий указанного каталога. Так, с ключом -sh можно получить суммарный размер всех файлов в этой директории и всех ее поддиректориях:

```
$ du -sh
```

Управление процессами

Каждая запущенная во FreeBSD программа — это один или несколько процессов в оперативной памяти компьютера. Запуская программу, пользователь порождает процесс, который, в свою очередь, может породить другие процессы. Наряду с процессами пользователей в памяти машины постоянно находится несколько системных процессов, которые обеспечивают функционирование FreeBSD. Поскольку процессы могут «зависать», потреблять слишком много ресурсов или вызывать конфликты, существует набор команд для управления ими.

Команда ps выводит список процессов, запущенных в системе на текущий момент. По умолчанию ps отображает процессы, принадлежащие запустившему ее пользователю.

```
$ ps
```

Аналогом команды ps является утилита top. Без дополнительных аргументов командной строки top выводит на экран список всех процессов в системе, рассортированный по проценту загрузки каждым процессом ЦП компьютера.

```
$ top
```

Все процессы имеют идентификационные номера: PID. Передавая номера своих процессов в качестве аргументов команде kill, пользователь может завершить их. Суперпользователь может таким образом завершать любой процесс любого пользователя, в том числе и системные процессы.

Следующая команда завершит процесс (то есть пошлет процессу сигнал 9, sigkill) под номером 137.

```
$ kill -9 137
```

С помощью команды killall можно завершить группу процессов по имени: например, следующая команда остановит все процессы веб-сервера Apache:

```
# killall -9 httpd
```

Впрочем, такое обхождение с процессами является внештатным и должно применяться лишь в крайнем случае — например, когда дальнейшее исполнение процесса угрожает стабильности всей системы в целом. Многие программы имеют стандартные способы завершения работы, и именно ими нужно пытаться пользоваться в первую очередь.

Указывая в аргументах командам kill и killall другие номера сигналов, можно различным образом влиять на работу процессов. Например, при получении сигнала 1 (sighup) процессы обычно перечитывают свои конфигурационные файлы:

```
# kill -1 2087
```

Команда jail создает для процесса виртуальную машину FreeBSD. Пользователь «запирает» процесс в рамках этой виртуальной машины, указывая в аргументах командной строки ее имя, IP-адрес, а также директорию основной файловой системы, которая будет служить корневой для указанного процесса и всех его подпроцессов. Таким образом, для пользователей на одном компьютере можно создать иллюзию работы с выделенными ма- »

Базовые команды

Основа работы в консоли

Хотя большинство базовых консольных программ имеет графические аналоги, знать синтаксис их консольных вариантов просто необходимо. Вот небольшой список основных консольных команд.

Список файлов в текущей директории:
$ ls -F

Подробный листинг текущей директории:
$ ls -l

Текущая директория:
$ pwd

Смена текущей директории:
$ cd директория

Копирование файла:
$ cp файл директория

Переименование/перемещение файла:
$ mv файл директория

Удаление файла:
$ rm файл

Смена владельца файла:
$ chown пользователь:группа файл

Смена прав доступа к файлу:
$ chmod права файл

Справка о командах и установленных программах:
$ man команда

Просмотр файла:
$ less файл

Редактирование файла (разница в используемых текстовых редакторах):
$ ee файл
$ vi файл

Поиск файла:
$ find маска

Поиск подстроки в файлах:
$ grep подстрока маска

Смена текущего пользователя:
$ su пользователь

» шинами. Чтобы процесс в рамках jail мог успешно работать, необходимо предварительно подготовить для него стандартный набор файлов дистрибутива FreeBSD, который должен быть размещен в корневой для jail-процесса директории. Подробно все инструкции по подготовке к созданию jail-процесса описаны в справочном файле по jail:

```
$ man jail
```

Утилита cron запускает программы с заданной периодичностью по расписанию. Это планировщик задач FreeBSD. Суперпользователь добавляет новые задачи в общесистемный файл расписания /etc/crontab, который периодически перечитывается демоном cron. Пользователи могут создать файлы задач в своих домашних директориях, а затем передать их на исполнение сервису cron с помощью команды:

```
$ cron имя_файла_расписания
```

Работа с пользователями

Даже если компьютер с FreeBSD используется только одним человеком, в системе по умолчанию заведены учетные записи нескольких пользователей: основная рабочая пользовательская учетная запись, учетная запись суперпользователя, системные учетные записи, необходимые для корректной работы некоторых программ и системных утилит. Посмотреть список всех пользователей системы можно, выведя на экран файл учетных записей /etc/passwd:

```
$ less /etc/passwd
```

В описании каждой системной учетной записи в этом файле указано, для каких целей она используется системой. Другой способ получить список пользователей — прибегнуть к мощной утилите pw, служащей для администрирования учетных записей и групп в системе. Следующая команда аналогична предыдущей, но использует для вывода списка пользователей утилиту pw:

```
$ pw user show -a
```

Утилитой pw суперпользователь добавляет, редактирует и удаляет учетные записи и группы. Чтобы завести в системе нового пользователя, необходимо вызвать pw с параметром useradd и названием учетной записи в качестве второго аргумента; чтобы удалить — с параметром userdel, а чтобы изменить его — с параметром usermod. Аналогичные параметры используются для добавления, изменения и удаления групп пользователей: groupadd, groupmod и groupdel соответственно. Например, следующая консольная команда создаст пользователя simpeuser:

```
# pw user add simpleuser -c “Simple User”
    -d /home/simpleuser -s /bin/csh
```

Во FreeBSD существуют альтернативные команды для работы с учетными записями. Команда adduser является интерактивной и предлагает администратору заводить пользователей в диалоговом режиме:

```
# adduser
```

С помощью интерактивной команды rmuser суперпользователь может удалять ненужные учетные записи:

```
# rmuser имя_пользователя
```

Аналогично командам adduser и rmuser, команда chpass предоставляет администратору интерактивный интерфейс для изменения параметров учетных записей:

```
# chpass имя_пользователя
```

Каждый пользователь может изменить свой пароль самостоятельно с помощью простой команды passwd, вызванной без параметров. Суперпользователь с помощью этой утилиты также изменяет пароли других пользователей системы, указывая их имена в качестве первого параметра:

```
# passwd user
```

Команда who выводит список пользователей, работающих с системой в настоящий момент, причем для удаленных пользователей выводится IP-адрес машины:

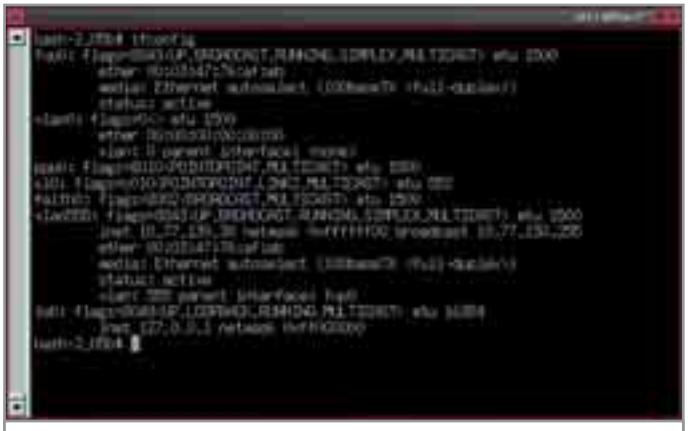
```
$ who
```

Для забывчивых пользователей предусмотрена следующая команда:

```
$ who am i
```

»

Интерактивная команда добавления пользователей — adduser



Сетевые интерфейсы конфигурируются программой ifconfig

```
$ ftp ftp://user:password@ftp.freebsd.
    org:21/pub/
```

Команда telnet нужна для консольного доступа к удаленной машине по сети по протоколу telnet. Команда принимает в качестве параметра имя или адрес удаленной машины. В том случае, если имя удаленного пользователя отличается от имени пользователя локального, команде необходимо с помощью ключа -l указать имя удаленной учетной записи:

```
$ telnet -l пользователь удаленный_сервер
```

Программа ssh аналогична команде telnet, но для удаленного доступа использует более безопасный протокол SSH. Использование ssh предпочтительнее, так как в отличие от утилиты telnet, которая передает пароль пользователя по сети в открытом виде, ssh пользовательский пароль перед передачей серверу предварительно шифрует:

```
$ ssh пользователь@удаленный_сервер
```

С помощью программы scp можно передавать по протоколу SSH-файлы между машинами сети. Использование scp предпочтительнее использования программы ftp по той же причине, по которой лучше использовать ssh, чем telnet: как авторизационную информацию, так и копируемые файлы scp передает по сети в зашифрованном виде. Примеры использования программы scp:

```
$ scp локальный_файл пользователь
    @удаленный_сервер:удаленная_
    директория
$ scp пользователь@удаленный_сервер:
    удаленный_файл локальная_
    директория
```

Перезагрузка и выключение компьютера

FreeBSD, как и другие современные операционные системы, не любит выключения компьютера без предупреждения. Перед выключением питания компьютера все пользовательские и системные процессы необходимо завершать штатным образом: первые — чтобы по возможности не потерять важные данные, находящиеся в процессе редактирования, вторые — чтобы при следующем запуске не оставалось блокировочных файлов и все системные компоненты стартовали корректно. Файловые системы должны быть размонтированы, чтобы не потерять информацию, находящуюся в кеше и еще не записанную на сам носитель. Для штатного завершения работы FreeBSD используются консольные команды halt и reboot. Вызванная без параметров суперпользователем команда halt корректно завершит работу операционной системы и выключит компьютер, команда reboot — перезагрузит его:

```
# halt
# reboot
```

Команда shutdown предоставляет суперпользователю больше возможностей: с ее помощью можно указать системе выполнить отложенную перезагрузку или выключение компьютера, указать сообщение, которое будет разослано на консоли всех пользователей, чтобы сообщить им о предстоящем выключении сервера и о необходимости завершить всю работу с системой. Следующие вариант команды shutdown через десять минут остановит FreeBSD и выключит питание у компьютера, предварительно послав всем пользователям предупреждение:

```
# shutdown -p +10:00 Finish your tasks in
    10 minutes, the server will be halted
```

Заключение

Здесь приведен далеко не полный список консольных команд, которые могут понадобиться администратору FreeBSD в повседневной работе и настройке системы. Но, зная базовые принципы работы ОС в целом и используя справку, пользователь уже сможет не чувствовать себя чужаком в командной строке. Несмотря на то, что графический интерфейс зачастую удобнее и нагляднее, работать с приложениями в консоли быстрее, а в некоторых ситуациях и надежнее. Поэтому не нужно бояться и избегать командной строки. Для успешной работы с FreeBSD консоль должна стать другом администратора.

■ ■ ■ Александр Юрьин

Дисковые квоты

По одной пачке в руки

Нельзя не упомянуть о знаменитом встроенном механизме дисковых квот FreeBSD. С помощью квотирования администратор может ограничить использование дискового пространства отдельными пользователями или их группами. Квотирование должно быть включено на уровне ядра. В системном конфигурационном файле /etc/rc.conf в переменной enable_quotas нужно установить «YES», а в файле /etc/fstab добавить параметры userquota и groupquota тем файловым системам, для которых этот механизм необходим. Далее, чтобы назначить собственно квоты, необходимо воспользоваться командой edquota, указав ей в качестве параметров ключ -u и имя пользователя, которому необходимо назначить лимиты (или, соответственно, ключ -g и название группы).

```
# edquota –u имя_пользователя
```

Команда edquota откроет установленный по умолчанию редактор, в котором для каждой файловой системы с включенной поддержкой квотирования прописаны стандартные ограничения дискового пространства. При необходимости эти значения можно изменить. Новые квоты вступят в действие сразу после закрытия редактора. Проверить действие квот можно с помощью команды quota, указав ей в качестве аргумента имя пользователя, или (при использовании ключа –g) название группы:

```
# quota имя_пользователя
```

```
$ ftp ftp://user:password@ftp.freebsd.
    org:21/pub/
```

Команда telnet нужна для консольного доступа к удаленной машине по сети по протоколу telnet. Команда принимает в качестве параметра имя или адрес удаленной машины. В том случае, если имя удаленного пользователя отличается от имени пользователя локального, команде необходимо с помощью ключа -l указать имя удаленной учетной записи:

```
$ telnet -l пользователь удаленный_сервер
```

Программа ssh аналогична команде telnet, но для удаленного доступа использует более безопасный протокол SSH. Использование ssh предпочтительнее, так как в отличие от утилиты telnet, которая передает пароль пользователя по сети в открытом виде, ssh пользовательский пароль перед передачей серверу предварительно шифрует:

```
$ ssh пользователь@удаленный_сервер
```

С помощью программы scp можно передавать по протоколу SSH-файлы между машинами сети. Использование scp предпочтительнее использования программы ftp по той же причине, по которой лучше использовать ssh, чем telnet: как авторизационную информацию, так и копируемые файлы scp передает по сети в зашифрованном виде. Примеры использования программы scp:

```
$ scp локальный_файл пользователь
    @удаленный_сервер:удаленная_
    директория
$ scp пользователь@удаленный_сервер:
    удаленный_файл локальная_
    директория
```

Перезагрузка и выключение компьютера

FreeBSD, как и другие современные операционные системы, не любит выключения компьютера без предупреждения. Перед выключением питания компьютера все пользовательские и системные процессы необходимо завершать штатным образом: первые — чтобы по возможности не потерять важные данные, находящиеся в процессе редактирования, вторые — чтобы при следующем запуске не оставалось блокировочных файлов и все системные компоненты стартовали корректно. Файловые системы должны быть размонтированы, чтобы не потерять информацию, находящуюся в кеше и еще не записанную на сам носитель. Для штатного завершения работы FreeBSD используются консольные команды halt и reboot. Вызванная без параметров суперпользователем команда halt корректно завершит работу операционной системы и выключит компьютер, команда reboot — перезагрузит его:

```
# halt
# reboot
```

Команда shutdown предоставляет суперпользователю больше возможностей: с ее помощью можно указать системе выполнить отложенную перезагрузку или выключение компьютера, указать сообщение, которое будет разослано на консоли всех пользователей, чтобы сообщить им о предстоящем выключении сервера и о необходимости завершить всю работу с системой. Следующие вариант команды shutdown через десять минут остановит FreeBSD и выключит питание у компьютера, предварительно послав всем пользователям предупреждение:

```
# shutdown -p +10:00 Finish your tasks in
    10 minutes, the server will be halted
```

Заключение

Здесь приведен далеко не полный список консольных команд, которые могут понадобиться администратору FreeBSD в повседневной работе и настройке системы. Но, зная базовые принципы работы ОС в целом и используя справку, пользователь уже сможет не чувствовать себя чужаком в командной строке. Несмотря на то, что графический интерфейс зачастую удобнее и нагляднее, работать с приложениями в консоли быстрее, а в некоторых ситуациях и надежнее. Поэтому не нужно бояться и избегать командной строки. Для успешной работы с FreeBSD консоль должна стать другом администратора.

■ ■ ■ Александр Юрьин

Дисковые квоты

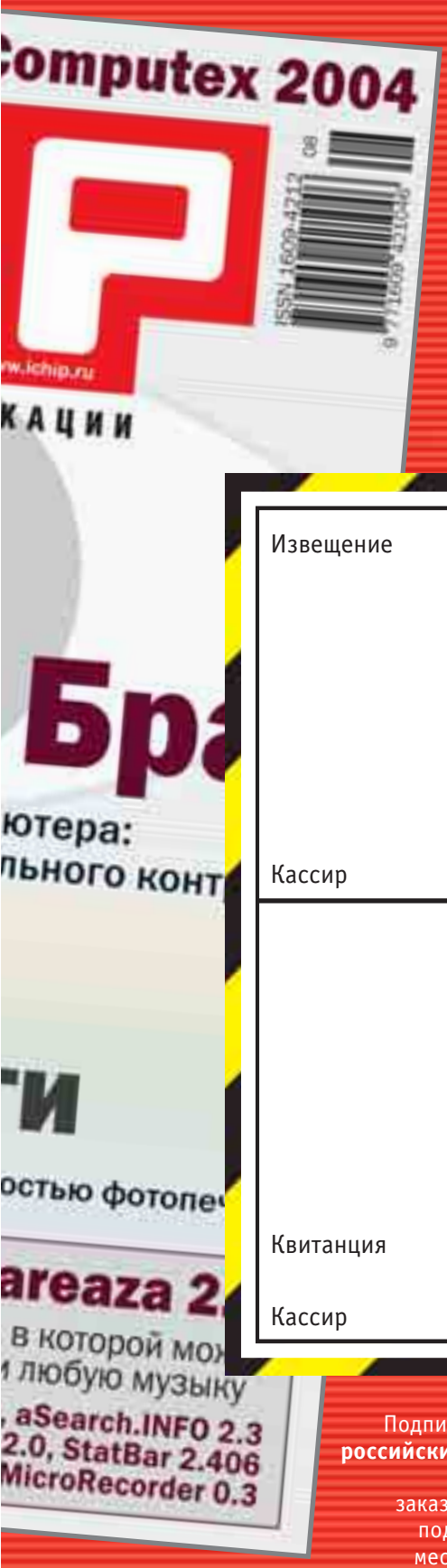
По одной пачке в руки

Нельзя не упомянуть о знаменитом встроенном механизме дисковых квот FreeBSD. С помощью квотирования администратор может ограничить использование дискового пространства отдельными пользователями или их группами. Квотирование должно быть включено на уровне ядра. В системном конфигурационном файле /etc/rc.conf в переменной enable_quotas нужно установить «YES», а в файле /etc/fstab добавить параметры userquota и groupquota тем файловым системам, для которых этот механизм необходим. Далее, чтобы назначить собственно квоты, необходимо воспользоваться командой edquota, указав ей в качестве параметров ключ -u и имя пользователя, которому необходимо назначить лимиты (или, соответственно, ключ -g и название группы).

```
# edquota –u имя_пользователя
```

Команда edquota откроет установленный по умолчанию редактор, в котором для каждой файловой системы с включенной поддержкой квотирования прописаны стандартные ограничения дискового пространства. При необходимости эти значения можно изменить. Новые квоты вступят в действие сразу после закрытия редактора. Проверить действие квот можно с помощью команды quota, указав ей в качестве аргумента имя пользователя, или (при использовании ключа –g) название группы:

```
# quota имя_пользователя
```



Кто хочет знать, читает CHIP

Подписка!

Извещение	ИНН 7705056238 ЗАО "Издательский дом "Бурда" р/сч № 40702810900020106298 в Сбербанке России г. Москва к/сч № 30101810400000000225 в ОПЕРУ Моск. ГТУ Банка России БИК 044525225 Плательщик Адрес Назначение платежа / Сумма ***Журнал CHIP _____ номеров*** Подпись плательщика
Кассир	
Квитанция	ИНН 7705056238 ЗАО "Издательский дом "Бурда" р/сч № 40702810900020106298 в Сбербанке России г. Москва к/сч № 30101810400000000225 в ОПЕРУ Моск. ГТУ Банка России БИК 044525225 Плательщик Адрес Назначение платежа / Сумма ***Журнал CHIP _____ номеров*** Подпись плательщика
Кассир	

Подписку можно оформить **на любой российский адрес.** В стоимость подписки включена доставка журналов заказными бандеролями. При оплате подписки до 15-го числа текущего месяца вы будете получать номера со следующего месяца.

Цена за:
6 номеров 570 рублей
12 номеров 1140 рублей

Для оформления подписки заполните платежный документ и оплатите свой заказ через отделение Сбербанка.
При заполнении бланка разборчиво укажите фамилию, имя, отчество полностью, почтовый индекс и адрес получателя.
В графе «Назначение платежа» напишите количество номеров издания. В графе «Сумма» проставьте сумму за выбранное вами количество номеров.

Адрес для писем: 125284
Москва, а/я 125

Телефоны для справок:
ЗАО «Бурда Директ» (095) 916-5706
E-mail: abo@burdadirect.ru
ЗАО «АПР» (095) 101-2537

Подписка через Интернет:
www.burdadirect.ru,
www.pressa.apr.ru/index/44077
Подписной индекс **44077**
в Объединенном каталоге «Пресса России»
и каталоге «Роспечать»
Подписной индекс **99006** в каталоге «МАП»
Распространение и подписка в **Белоруссии:**
УП «РЭМ-Инфо», Минск, тел. (017) 291-9891/98,
подписной индекс в каталоге Белпочты **44077**

Вмешательство на расстоянии

Основы работы с SSH

Одно из основных преимуществ Unix-серверов — возможность удаленного администрирования даже через очень медленные линии связи. Следование идеологии работы и администрирования через интерфейс командной строки гарантирует, что вы сможете выполнить любое действие, необходимое для управления сервером.

Изначально для выполнения действий на удаленных Unix-машинах были использованы telnet и наборы программ rlogin, rsh и rcp. В основу защищенности этих продуктов, созданных очень давно, был положен принцип защиты сетевого транспорта. Время показало, что можно «украсть» чужой IP-адрес, можно перенаправить клиента на свой хост, взломав DNS-сервер или маршрутизатор, можно «прослушивать» трафик, находясь в том же сегменте сети, где проходят пакеты от клиента к серверу, и наконец, информацию можно снять даже удаленно прямо с сетевого кабеля, считывая электромагнитное излучение. Поэтому на сегодняшний день фактическим стандартом стало использование для администрирования Unix-серверов SSH — Secure SHell, которая предоставляет все преимущества SSL и еще множество дополнительных возможностей.

Ниже мы рассмотрим основные свойства SSH, используя в качестве примера свободно распространяемый пакет OpenSSH (http://www.openssh.org), который в настоящее время присутствует во всех дистрибутивах.

Шифрование трафика криптостойкими алгоритмами

Использование «слабых» алгоритмов возможно, но обычно запрещается администратором сервера, чтобы гарантировать надежное шифрование передаваемых данных. Пользователь может выбирать между стандартизованным 3DES, быстрым Blowfish и редко применяемыми CAST128 и Arcfour.

Авторизация

Авторизация, вернее, ее разновидность, основанная на криптографии с открытым »

» ключом (public-key cryptography), гарантирует, что мы связываемся именно с тем хостом, с которым нам необходимо. Это особенно важно, учитывая, что IP-адрес сервера может быть «украден» (IP Spoofing), или маршрутизация и DNS-зона могут быть изменены взломщиком с целью перенаправления запросов к серверу на свой хост, что позволило бы, например, перехватывать пароли, симулируя логин. Криптография с открытым ключом, разработанная Уайтфильдом Диффи, использует асимметричное шифрование, то есть пару ключей, обладающую следующими свойствами: что-либо, зашифрованное одним из ключей, может быть расшифровано с помощью другого; имея один ключ из пары, называемый открытым, невозможно получить другой — секретный.

В SSH проверяется соответствие ключа IP-адресу сервера и его доменному имени, и в случае несовпадения пользователю выдается соответствующее предупреждение.

Кроме того, не стоит забывать о регулярной смене ключа для шифрования трафика во время сессии. По умолчанию новый ключ создается один раз в час. Это значительно снижает возможность эффективной расшифровки перехваченных данных.

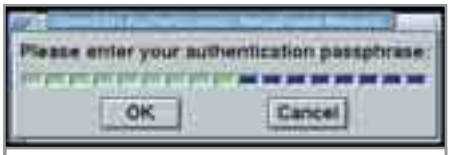
Защищенный туннель для протокола X11

SSH автоматически устанавливает переменную DISPLAY на удаленном хосте и создает туннель для X11 соединений. Это позволяет прозрачно для пользователя запускать на удаленном хосте графические приложения, которые при этом будут отображаться на локальной рабочей станции. Для включения этой возможности необходимо, чтобы в sshd_config на сервере параметр X11Forwarding был установлен в значение «YES», а при соединении с сервером у SSH-клиента либо в командной строке (ключ -X для OpenSSH), либо в конфигурационном файле (~/.ssh/config) необходимо эту возможность разрешить. При этом SSH-сервер открывает TCP-сокет с номером порта, вычисляемым как 6000 (стандартный порт для X11 при DISPLAY=:0) + X11DisplayOffset (взятом из sshd_config).

Перенаправление указанных TCP/IP-портов

При перенаправлении через защищенный туннель принцип работы практически тот же, что и при создании X11-туннеля. Поддерживаются два вида туннелей: локальный (local) и удаленный (remote).

Локальный туннель можно установить, выполнив команду ssh -L port:host:hostport user@server. При этом трафик на указанный port на локальной (клиентской) машине через SSH-туннель будет перенаправляться на server и уже с него будет устанавливаться соединение на порт port хоста host. Удаленный туннель устанавливается аналогичным способом, с той лишь разницей, что перенаправляется порт на удаленном сервере на сторону клиента, и соединения, сделанные на сервере server на локальный порт port, будут перенаправляться на клиента, и оттуда на указанный порт port хоста host. Синтаксис команды ssh -R port:host:hostport user@server. Пример:



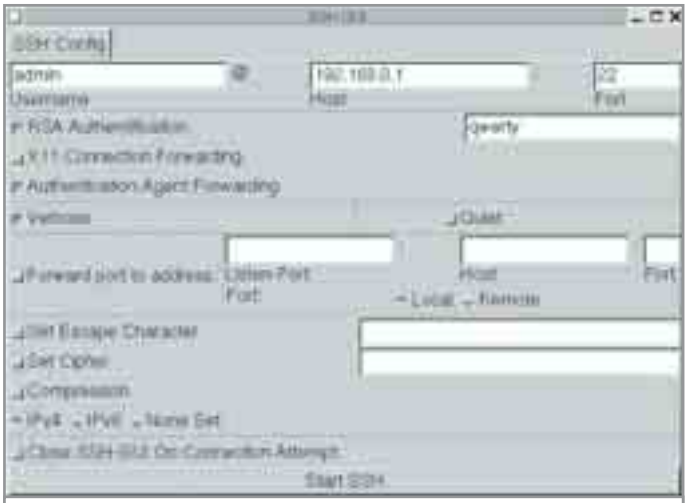
Окно аутентификации OpenSSH выполнено весьма оригинально

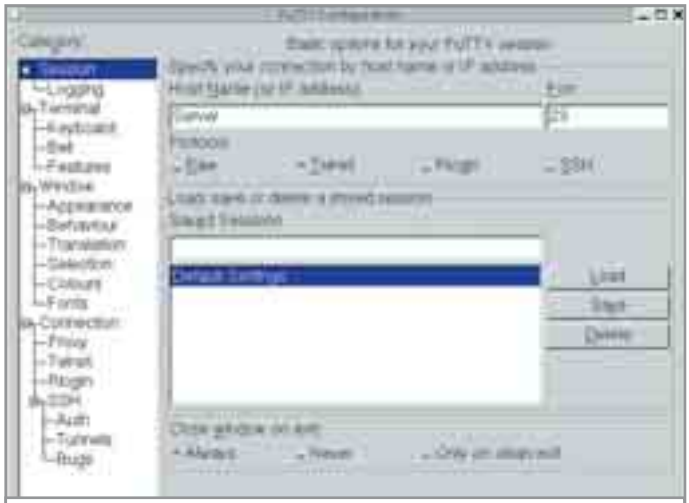
```
ssh -fN -L 3128:localhost:3128 myserver
```

Теперь можно в настройках браузера на локальной машине указать в качестве прокси-сервера localhost:3128. При этом на самом деле будет использоваться прокси, находящийся на сервере myserver. Для прокси-сервера соединения будут выглядеть приходящими с адреса 127.0.0.1 (localhost). Ключ -fN указан для того, чтобы SSH сразу ушла в фоновый режим, так как в данном случае нам не требуется выполнять никаких команд.

Опциональное сжатие передаваемых данных

При этом используется тот же алгоритм, что и в gzip. Сжатие включается ключом SSH-клиента и действует для всех передаваемых или получаемых данных (включая туннели) в пределах данной сессии. Это может »


Графическая оболочка для SSH-клиента доступна в коллекции портов /usr/ports/security/ssh-gui


Клиент PuTTY работает с несколькими протоколами и существует как под Unix, так и под Windows

» значительно ускорять передачу данных при использовании медленных модемных или низкоскоростных ADSL-линий.

Особенности авторизации

Рассмотрим подробнее систему аутентификации в SSH, основанную на криптографии с открытым ключом, поскольку правильное ее применение очень удобно при повседневном администрировании.

Telnet и rlogin при установлении соединения всегда запрашивали имя пользователя и пароль. Это защищало от неавторизованного входа в систему, но каждый раз требовало ввода пароля администратором, что было неудобно, или хранения его в скриптах, что отнюдь не повышало защищенность сервера. В любом случае, пароль передавался по сети в открытом виде, что позволяло его легко перехватить.

Rsh, напротив, использовал для авторизации только IP-адрес удаленного хоста, что было очень удобно в скриптах, но открывало широкое поле деятельности для всевозможных взломщиков.

SSH реализует и классическую схему авторизации, когда сервер запрашивает пароль у клиента, и подобную rsh авторизацию, срабатывающую при установлении «личности» сервера. Представим однако, что удаленный сервер был взломан, и хакер смог подменить sshd на свой, сохраняющий все введенные пароли в файл. Несмотря на то, что все данные шифруются при передаче по сети, на стороне сервера они все равно расшифровываются, и дальше пароль сравнивается с тем, что хранится на сервере. Проблема усугубляется тем, что администраторы, как правило, используют один и тот же пароль сразу на нескольких серверах, поэтому хакеру, получившему пароль от одного сервера и проанализировавшему ~/.ssh/known_hosts, становятся доступными и другие хосты, а находясь в /etc/sudoers и имея доступ суперпользователя на новом сервере-жертве, он может подменить sshd и там, постоянно расширяя область своего проникновения.

Третий способ авторизации в SSH позволяет элегантно и эффективно решить эту »

Модемный терминал

Из дома – в офисную локальную сеть

Достаточно часто администраторам необходимо дозвониться до сервера по модему и получить доступ к локальной сети и Интернету, используя свое офисное интернет-соединение. Для этого на сервере необходимо настроить модемный доступ. Итак:

► Подключаем модем, определяем, на каком порту он виден, и проверяем его работоспособность, например, с помощью интерактивного режима программы ppp.

► Устанавливаем mgetty из портов:

```
cd /usr/ports/comms/mgetty+sendfax &&
make install all
```

Можно использовать установки по умолчанию, поскольку они вполне работоспособны. Указываем mgetty, какой скрипт (login program) использовать для входа. Создайте файл под названием /etc/ppp/ppp-shell, содержащий следующее:

```
#!/bin/sh
IDENT=`echo $0 | sed -e 's/^.*-\(.*\)$/\1/'`
CALLEDAS="$IDENT"
TTY=`tty`
if [ x$IDENT = xdialup ]; then
IDENT=`basename $TTY`
fi
echo "PPP for $CALLEDAS on $TTY"
echo "Starting PPP for $IDENT"
exec /usr/sbin/ppp -direct $IDENT
```

Этот скрипт должен быть исполняемым. Теперь создайте на этот скрипт символическую ссылку с именем ppp-dialup с помощью следующей команды:

```
# ln -s ppp-shell /etc/ppp/ppp-dialup
```

Используйте этот скрипт в качестве оболочки для удаленных пользователей. Ниже приведен пример записи в /etc/password для удаленных пользователей PPP с именем пользователя pchilds .

```
pchilds:*:1011:300:Peter Childs
        PPP:/home/ppp:/etc/ppp/ppp-dialup
```

Создайте каталог /home/ppp, доступный для чтения и содержащий следующие файлы нулевой длины:

```
-r--r--r-- 1 root wheel 0 May 27 02:23
        .hushlogin
-r--r--r-- 1 root wheel 0 May 27 02:22
        .rhosts
```

Это предотвратит отображение /etc/motd. Настройка и компиляция mgetty с параметром AUTO_PPP позволяет mgetty определять LCP-фазу PPP-соединений и автоматически порождать оболочку ppp. Однако, поскольку стандартный метод «логин/пароль» не используется, необходима аутентификация пользователей через PAP или CHAP.

В этом разделе предполагается, что пользователь успешно настроил, скомпилировал и установил версию mgetty с параметром AUTO_PPP (v0.99beta или более позднюю).

Убедитесь, что в файле /usr/local/etc/mgetty+sendfax/login.config имеется следующая строка:

```
/AutoPPP/ — - /etc/ppp/ppp-pap-dialup
```

Это укажет mgetty запускать скрипт ppp-pap-dialup для обнаруженных соединений PPP-типа.

Создайте файл /etc/ppp/ppp-pap-dialup, содержащий следующее (этот файл должен быть выполняемым):

```
#!/bin/sh
exec /usr/sbin/ppp -direct pap$IDENT
```

» и несколько сопутствующих проблем парольной аутентификации. В основу так называемой Public Key Authentication — авторизации по открытому ключу — положен принцип шифрования с открытым ключом, используемый и в SSL, и в PGP. Очевидно, что «кража» открытого ключа — единственного куска аутентификационных данных, передаваемого на сервер, — не даст хакеру абсолютно ничего. Рассмотрим данную технику работы подробнее.

Шифрование с открытым ключом

► Создаем пару «открытый/закрытый ключ» с помощью утилиты ssh-keygen:

```
user@myhost:~$ ssh-keygen -b 2048 -f
    mykey -t dsa
Generating public/private dsa key pair
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in mykey.
Your public key has been saved in
    mykey.pub.
The key fingerprint is:
4d:f8:b7:40:de:af:68:82:d5:bf:81:98:12:1b:
    4b:79 user@myhost
user@myhost:~$
```

параметр -b указывает размер ключа в битах, стандартные рекомендуемые значения — 512, 1024, 2048;

-t — тип ключа. Для второй версии протокола можно выбирать rsa или dsa, для устаревшей первой используется значение rsa1;

-f <filename> — имя файла, куда будет записан закрытый ключ. Открытый будет сохранен в соответствующий файл с расширением .pub;

Ключ -f можно и не указывать; в этом случае будет создан ключ, используемый ssh по умолчанию.

► Каким-либо способом (например, через тот же SSH, используя парольную аутентификацию) добавляем открытый ключ в файл $HOME/.ssh/authorized_keys, где $HOME — домашний каталог пользователя, под именем которого мы хотим заходить на сервер. Очень удобно выполнять эту проце- »

Для каждой линии, включенной в /etc/ttys, создайте соответствующую запись в /etc/ppp/ppp.conf. Она будет отлично сочетаться с тем, что было создано выше.

```
pap:
 enable pap
 set ifaddr 203.14.100.1 203.14.100.20-
     203.14.100.40
 enable proxy
```

Для каждого пользователя, входящего в сеть по этому методу, в файле /etc/ppp/ppp.secret должна присутствовать запись с логином/паролем. Правда, есть и альтернативный вариант: для аутентификации пользователей по PAP через /etc/password необходимо использовать следующий параметр:

```
enable passwdauth
```

Если вы хотите присвоить некоторым пользователям статический IP, задайте его в качестве третьего аргумента в /etc/ppp/ppp.secret.

► Настраиваем ppp.conf. Вписываем туда "enable proxy" и проверяем, что /etc/rc.conf содержит необходимую строку gateway_enable="YES"

Вам потребуется добавить раздел для каждого из пользователей со статическими IP-адресами.

```
User1:
 set ifaddr 203.14.100.1 203.14.101.1
     255.255.255.255
User2:
 set ifaddr 203.14.100.1 203.14.102.1
     255.255.255.255
User3:
 set ifaddr 203.14.100.1 203.14.103.1
     255.255.255.255
```

Если необходимо, файл /etc/ppp/ppp.linkup должен также содержать информацию о маршрутизации для каждого пользователя со статическим IP-адресами. Ниже приведен такой пример.

```
User1:
 add 203.14.101.0 netmask 255.255.255.0
     HISADDR
User2:
 add 203.14.102.0 netmask 255.255.255.0
     HISADDR
User3:
 add 203.14.103.0 netmask 255.255.255.0
     HISADDR
```

► Настраиваем /etc/ppp/ppp.secret. Этот файл применяется для аутентификации пользователя во время инициализации ppp-соединения. Чтобы задействовать пользовательский пароль из /etc/passwd, поставьте "*" в поле AUTHKEY. Для его использования в секции сервера в ppp.conf должна быть включена опция passwdauth. Эта функция, хоть и удобна, но работает только с PAP-авторизацией.

► Проверяем /etc/ttys. Во время установки mgetty уже должна была добавить свои записи в конец файла, поэтому достаточно просто убедиться, что файл устройства указан корректно. По умолчанию строчка выглядит как "cuaa0 "/usr/local/sbin/mgetty" dialup on"

► Предлагаем init перечитать /etc/ttys следующей командой: kill -HUP 1.

Сервер готов принимать входящие звонки. Пользователь при входе будет получать адрес из внутренней сети и использовать доступные ресурсы.

Для получения дополнительной информации вы всегда сможете использовать FreeBSD Handbook — полную документацию по этой системе. Адрес русской версии handbook —http://www.freebsd.org/doc/ru/books/handbook/.

» дуру с помощью входящей в состав OpenSSH программы ssh-copy-id:

```
user@myhost:~$ ssh-copy-id -i mykey.pub
      user@server

Password:
/usr/bin/X11/xauth: creating new
      authority file /home/user/.Xauthority
Now try logging into the machine,
      with "ssh 'user@server'", and check in:

.ssh/authorized_keys

to make sure we haven't added extra
      keys that you weren't expecting.

user@myhost:~$
```

Если ssh-copy-id для вашего SSH-клиента нет, можно перенести ключ следующей командой: cat mykey.pub | ssh user@ server 'cat >> ~/.ssh/authorized_keys'.

► Опционально подгружаем закрытый ключ из пары в ssh-agent через ssh-add:

```
user@myhost:~$ ssh-add mykey
Enter passphrase for mykey:
Bad passphrase, try again for mykey:
Identity added: mykey (mykey)
```

Установка пароля для шифрования закрытого ключа позволяет предотвратить его использование в случае кражи с носителя, где он находится. Согласитесь, что помнить всего один пароль и обеспечить сохранность одного ключа значительно легче, чем пароли и ключи десятков серверов, которые вы администрируете. Пароль выбирается при создании ключа командой ssh-keygen; с ее же помощью ключ можно перекодировать под другим паролем.

Конечно, SSH-agent использовать необязательно, но в этом случае SSH будет запрашивать пароль для ключа каждый раз при входе на сервер. Можно также сделать закрытый ключ, не зашифрованный каким-либо паролем, что позволяет использовать SSH в скриптах для выполнения команд на удаленных машинах. Тут возникает закономерный вопрос: «А что, если этот ключ украдут? Ведь тогда взломщик получит доступ к серверу с любого компьютера!». Для предотвращения этого в SSH предусмотрено несколько мер.

Первая из них — ограничение возможности использования ключа конкретными адресами. Для этого после добавления открытого ключа в authorized_keys на сервере откройте его в редакторе и добавьте параметр from="hostname_mask". В качестве hostname_mask можно использовать доменное имя клиента, или его IP-адрес. Также поддерживаются wildcards: можно написать from="*.mycompany.ru", и тогда ключ будет действителен для всех хостов в зоне mycompany.ru. В параметре from можно прописать сразу несколько значений, разделяя их запятыми.

```
user@server$ cat $HOME/.ssh/
      authorized_keys
from="*.mycompany.ru" ssh-rsa
      AAAAB3GzaN4... public-key1
```

Вторая мера — это ограничение команд, которые можно выполнять, пользуясь данным ключом. Они указываются аналогично from, используя параметр command="/path/to/program args":

```
user@server$ cat $HOME/.ssh/
      authorized_keys
command="dump /home" ssh-rsa
      AAAAB3GzaN4... public-key1
```

Предусмотрены еще и ограничение возможности создания туннелей, резервирования устройства виртуального терминала и принудительное выставление некоторых переменных окружения. Необходимые параметры для этого можно найти в man sshd.

Как мы уже говорили, чтобы не вводить пароль при каждом входе на сервер, можно воспользоваться ssh-agent. Это специальная программа, хранящая в своей памяти закрытые ключи в расшифрованном виде и предоставляющая их через Unix-сокет приложениям по запросу. При запуске SSH-agent выставляет несколько переменных окружения, позволяющих программам найти его сокет, поэтому обычно он запускается во время инициализации сессии X, и все процессы пользователя далее являются его потомками, наследуя переменные окружения.

Добавление закрытых ключей осуществляется с помощью команды ssh-add, в качестве параметра которой передается путь до файла с закрытым ключом.

```
user@myhost:~$ ssh-add mykey
Enter passphrase for mykey:
Identity added: mykey (mykey)
user@myhost:~$
```

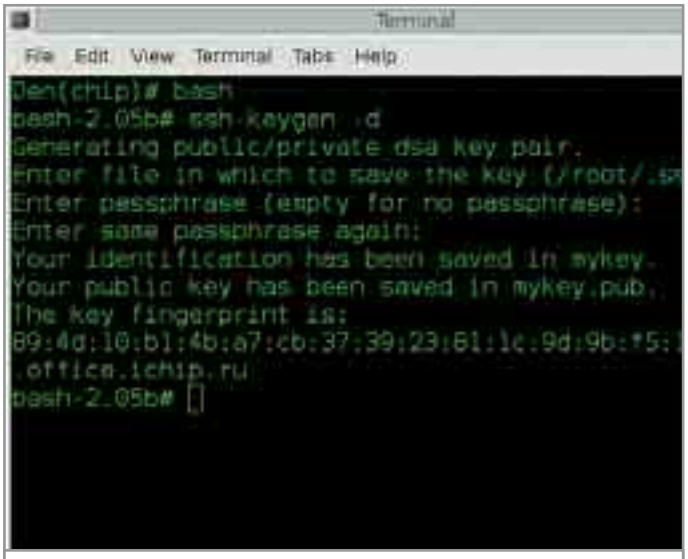
Если ssh-add запускается в интерактивном режиме (например, в xterm или консоли), то запрос на пароль для ключа выдается в тот же терминал. Когда же, например, »

Пакет Webmin

Администрирование через браузер

Существует возможность администрировать сервер, используя любой веб-браузер. Отличный пакет Webmin (http://www.webmin.com/) после небольшой настройки позволяет удаленно управлять сервером: заводить пользователей, принимать и передавать файлы, конфигурировать несколько десятков популярных приложений, таких как DNS–сервер, BIND, веб-сервер Apache, Squid, Samba, записывать компакт-диски, настраивать DHCP, почтовые сервисы, принт-сервер, SSH, firewall и т. д.

Webmin выполнен в виде простого веб-сервера и набора CGI-приложений, написанных на Perl. Это позволяет при необходимости самостоятельно добавлять необходимые функциональные возможности. Его интерфейс и модули практически полностью переведены на русский язык. Распространяется Webmin под BSD-подобной лицензией, что позволяет свободно применять, модифицировать и распространять его как для некоммерческого, так и для коммерческого использования.



Генерирование ключа с помощью команды ssh-keygen

Webmin — программа для администрирования через браузер

» обрабатывается ~/.xsession, ssh-add для получения пароля запускает программу, указанную в переменной окружения SSH_ASKPASS. По умолчанию это ssh-askpass — небольшая программа под X-Window System, входящая в пакет OpenSSH.

► Заходим на сервер точно так же, как и раньше, за исключением того, что у нас теперь не спрашивают пароль.

SSH на Windows-платформе

До сих пор мы обсуждали только OpenSSH, однако под Windows его использование может быть не всегда удобно, так как требует наличия Cygwin-окружения. Поэтому большую популярность завоевал другой бесплатный клиент SSH1/2– PuTTY (http:// www.chiark.greenend.org.uk/~sgtatham/putty/). Он предоставляет администратору простой и понятный графический интерфейс и полный набор утилит, которые мы обсуждали выше, включая программу для создания и импортирования/экспортирования ключей и аналог SSH-agent. Прилагаемая документация подробно описывает методику их использования под Windows, так что после прочтения этой статьи вы сможете разобраться в ней без особого труда. В сочетании с каким-либо X-сервером под Windows (Exceed от Humminbird, XFree86 из Cygwin) можно, находясь на рабочей станции с Windows, даже запускать на сервере графические программы конфигурирования, а при поддержке X-сервером OpenGL — и 3D-программы.

Если все еще остались вопросы...

Мы рассмотрели некоторые возможности работы с OpenSSH. Этот бесспорно важный и мощный пакет поддерживается проектом OpenBSD и основан на SSH v1.2.12 со всеми последними исправлениями и обновлениями, совместим с протоколами SSH версий 1 и 2. OpenSSH включен в базовую систему начиная с FreeBSD 4.0.

Для получения дополнительных сведений о работе OpenSSH вы всегда можете обратиться на страницы русской версии FreeBSD Handbook по адресу http://www.freebsd.org/doc/ru/books/handbook/ openssh.html, либо на страницы разработчика по адресу http:// www.openssh.com. Там же вы найдете последние версии этой полезной программы и свежие новости, что позволит вам постоянно поддерживать безопасность вашего сервера на должном уровне. ■ ■ ■ Константин Стародубцев