Дэвид Бэндл

ДЛЯ ПРОФЕССИОНАЛОВ

ЗАЩИТА И БЕЗОПАСНОСТЬ В СЕТЯХ LINUX

ПИТЕР

Содержание

Моей любящей жене Сильвии и детям, Лизе и Ванессе

Предисловие

Основной причиной, по которой возникла идея создания данной книги, стало все возрастающее количество домашних пользователей, которые соединены с Интернетом 24 часа в сутки семь дней в неделю. Это происходит благодаря все большей доступности и надежности таких служб, как Road Runner, и каналов связи adsl/ xdsl, предлагаемых телефонными компаниями. К этой аудитории можно присовокупить также небольшие предприятия, осознавшие важность Web и пришедшие к выводу, что поддержка своих web-ресурсов собственными силами предприятия обойдется им дешевле, чем использование для этой цели услуг сторонних компаний. Однако при этом руководители подобных предприятий понимают, что они не могут позволить себе принять на работу специального сотрудника, который является экспертом в области безопасности.

Данная книга посвящена компьютерной безопасности в операционной среде Linux и ориентирована на домашних пользователей и небольшие предприятия, которые не могут позволить себе содержание полноценной хорошо оснащенной компьютерной службы. В книге содержатся базовые сведения о компьютерной безопасности. Многие из читателей могут скептически отнестись к предположению, что материал данной книги будет для них понятным и полезным. Многие считают, что если такие большие и серьезные организации, как NASA и другие, могут подвергнуться успешным атакам взломщиков, несмотря на то, что в таких организациях на обеспечение защиты тратятся огромные средства, а безопасность обеспечивают множество людей, которые действительно являются экспертами в этой области, то что же можно сказать о домашних компьютерах и сетях небольших предприятий; очевидно, что злоумышленники могут взломать их даже с большим успехом. Однако на самом деле проблемы безопасности, стоящие перед компьютерными отделами крупных организаций, значительно серьезнее, чем проблемы, стоящие перед домашними пользователями и небольшими предприятиями. Во-первых, компьютерная система крупной организации вроде NASA — это более крупная и более привлекательная для злоумышленников цель. Во-вторых, сеть крупной организации обладает существенно более сложной структурой, в нее входит значительно большее количество систем, становится очень сложно следить за тем, чтобы каждая из этих систем использовала наиболее свежее, наиболее неуязвимое программное обеспечение. В-третьих, крупные организации обеспечивают доступ к огромному количеству служб для огромного количества пользователей и при этом пытаются находиться на самом переднем фронте компьютерных технологий. Любой, кто пытается удержаться на этом рубеже, рано

или поздно оказывается под ударом и несет потери. Самой безопасной и самой хорошо защищенной системой следует считать компьютер, который еще не извлечен из продажной упаковки, не включен в сеть и заперт в потайной комнате. Однако вряд ли такую систему можно назвать полезной.

Данная книга поможет вам понять устройство вашей системы с точки зрения безопасности. Глава за главой, концепция за концепцией вы будете овладевать основными понятиями и принципами защиты компьютерных систем. Прочитав книгу, вы не сможете стать экспертом в области компьютерной защиты, однако у вас появится базовый набор знаний, благодаря которым вы сможете приступить к освоению более сложного материала.

Книга представляет собой руководство не только для новичков, но и для профессионалов, поскольку содержит сведения, не слишком часто освещаемые в литературе. Когда я писал данную книгу, я в основном работал с Caldera OpenLinux и использовал средства, ориентированные на Caldera OpenLinux, однако рассматриваемые концепции применимы в отношении всех разновидностей Linux.

Не только начинающие, но и многие профессионалы ошибочно полагают, что комплекты Linux различных поставщиков существенно отличаются друг от друга, как это было в далеком 1988 году, когда на рынке в одно и то же время существовали две похожие операционные системы: MS-DOS и DR-DOS. Однако в действительности это не так. Ядро Linux, компоновку которого вы можете выполнить самостоятельно, получается с использованием одних и тех же файлов исходного кода — этот код используется во всех комплектах Linux. Демон telnet рекомендуется отключить в любой системе Linux, вне зависимости от поставщика. Различные комплекты Linux отличаются друг от друга в основном процедурами установки, а также различными наборами средств администрирования. Однако при этом каждый из комплектов Linux использует одни и те же библиотеки, одни и те же серверные программы, одну и ту же базовую структуру файловой системы, кроме того, разным комплектам Linux свойственны одни и те же недостатки и уязвимые места.

Таким образом, если вас интересуют базовые сведения о безопасности Linux, если вы хотите досконально разобраться в том, что это такое, как она работает и как ее улучшить, значит, это — ваша

книга. Тот факт, что вы дочитали данное короткое предисловие до конца, лишний раз подтверждает вашу заинтересованность в изучении рассматриваемой здесь тематики.

Как организована книга

Книга состоит из четырех частей.

Часть 1. Ваш узел

Материал этой части поможет вам понять организацию защиты вашего сетевого узла. Вы узнаете о пользователях, разрешениях на доступ, защите файлов и о том, как работает ваш узел. В данном разделе рассказывается, как организована защита доступа к системе со стороны пользователей, авторизованных для подключения к системе, а кроме того, как защитить вашу систему в процессе начальной загрузки.

Часть 2. Ваша сеть

Материал во второй части книги объясняет принципы работы сетей, а также содержит сведения, необходимые вам для того, чтобы понять основные приемы атак на компьютерные системы. Прочитав данную часть, вы поймете, что именно интересует злоумышленников в вашей системе и что они разыскивают, пытаясь проникнуть в ваш компьютер. Поняв это, вы сможете более эффективно противодействовать им.

Часть 3. Брандмауэры и специальное программное обеспечение

В третьей части рассказывается об использовании брандмауэров и другого специализированного программного обеспечения, позволяющего поднять уровень защиты вашей системы.

Часть 4. Аудит системы безопасности

Четвертая часть рассказывает о том, как организовать наблюдение за вашей системой. Прочитав ее, вы узнаете, как настроить систему протоколирования, как определить, какие именно данные должны протоколироваться, как настроить местоположение файлов журналов и как осуществлять чтение этих журналов в последующем. Вы также узнаете о том, каким образом злоумышленники могут повлиять на содержимое журналов. Вы узнаете, как следить за действиями злоумышленников, как обнаруживать попытки несанкционированного проникновения в систему и как противодействовать им. Наконец, вы познакомитесь с программными средствами, которые злоумышленники используют для изучения уязвимых мест вашей системы.

Приложения

В конце книги содержатся два приложения. В первом из них вы найдете перечень утилит, связанных с безопасностью, которые могут оказаться полезными для вас (некоторые из них записаны на прилагаемом к данной книге компакт-диске). Второе приложение описывает программы, записанные на прилагаемом к книге компакт-диске.

Обозначения

Чтобы облегчить восприятие материала книги, я использую следующие обозначения и выделения:

- комбинации клавиш, которые следует нажимать одновременно, показываются с использованием символа «плюс» (+), например, Ctrl+P;

- *курсив* используется для смыслового выделения, а также в случае, если некоторый термин встречается в книге впервые;

- специальным шрифтом обозначаются имена файлов, программ, адреса Интернета, команды, элементы интерфейса и т. п.

Также в тексте встречаются врезки нескольких разновидностей:

СОВЕТ

Врезка типа «Совет» подсказывает вам, как лучше всего на практике использовать информацию, излагаемую в книге.

ПРИМЕЧАНИЕ

Врезка типа «Примечание» информирует вас о специальных случаях и исключениях, возникающих в процессе функционирования некоторого механизма.

ВНИМАНИЕ

Врезка типа «Внимание» предупреждает вас о возможных проблемах, с которыми вы можете столкнуться.

ССЫЛКА

Врезка типа «Ссылка» указывает на другие места книги, в которых излагается материал, связанный с рассматриваемым.

Введение

Почему данная книга настолько важна

Каждый новый день все большее количество индивидуальных пользователей и предприятий открывают для себя операционную систему Linux. Каждый новый день все большее количество людей подключаются к Интернету, и все чаще для этой цели используется выделенное соединение. Предполагаемая анонимность Всемирной сети зачастую будит во многих пользователях Интернета самые низменные инстинкты. Иными словами, как это ни печально, с ростом числа пользователей Интернета во всемирной сети увеличивается также количество хулиганья. С этим придется иметь дело до тех пор, пока положение дел кардинальным образом не изменится и подобные отбросы общества не будут пойманы и посажены в тюрьму за совершенные ими преступления. В этом не может быть ошибки: проникновение в чужую систему или несанкционированное расходование пропускной способности чужих каналов связи (будь то преднамеренная атака Denial of Service или нежелательная «грязная» коммерческая электронная почта) является преступлением и должно преследоваться во всех уголках земного шара.

В настоящее время, когда недорогие выделенные соединения с Интернетом становятся атрибутом повседневной жизни, новички в мире Linux все больше нуждаются в простой доступной книге, посвященной вопросам компьютерной безопасности. Если вы покупаете новый дом и выясняется, что в нем нет ни одного замка, вы не стесняетесь позвонить слесарю, чтобы он пришел и установил вам замки на все двери. Так почему бы, приобретая новый сервер и подключая его к Интернету, не позаботиться должным образом о его защите?

Некоторые читатели, приобретая данную книгу, думают, что, прочитав ее и применив некоторые магические формулы, они могут на все сто процентов абсолютно надежно защитить себя от любых злонамеренных действий. К сожалению, никакой универсальной серебряной пули, поражающей всех ваших врагов, не существует. Никакие заклинания, молитвы и волшебные напитки не смогут полностью обезопасить вашу систему. Даже к самому лучшему замку со временем можно подобрать отмычку. Однако чем лучше замок, тем сложнее это сделать. Один из законов Мэрфи, адресованный солдатам, гласит: «если враг не может проникнуть к вам внутрь укрепления, значит, вы не сможете выйти наружу». Скорее всего, Мэрфи понятия не имел об Интернете и компьютерах, однако его закон очень подходит к информационным технологиям. Следует привыкать к бдительности. Помните, что кто-то пытается проникнуть в вашу систему, кто-то пытается обнаружить слабые места в вашей защите. Информация в данной книге затруднит ему его работу, и возможно, не достигнув успеха, он просто оставит злонамеренные попытки.

Летом 1999 года компания Microsoft подключила к Интернету собственный сервер IIS и предложила всем желающим попытаться взломать его. Этот сервер не продержался и трех часов. В другое время другими людьми к всемирной сети был подключен компьютер PowerPC с установленной на нем системой Linux. На сервере работала единственная доступная извне служба: web-сервер Apache. Всем желающим также предлагалось взломать этот сервер. Неделей позже хозяева сервера были вынуждены отключить его, но не потому, что систему удалось взломать. Отключить сервер пришлось потому, что в результате попыток взлома пострадали системы, принадлежащие провайдеру Интернета, который обеспечивал связь этого сервера с Всемирной паутиной. Некоторые из желающих взломать сервер попытались получить несанкционированный доступ к компьютерам, расположенным «близко» от сервера, для того чтобы проникнуть в этот сервер через задний ход. Однако взломанные системы принадлежали провайдеру, а атаки в отношении этих систем противоречили условиям соревнования. Сервер так и остался невзломанным, в то время как он в течение недели был доступен для всех пользователей Интернета. Этот сервер был действительно хорошо защищен. Прочитав данную книгу, вы станете лучше понимать, как именно можно достигнуть такого же уровня защиты.

В настоящее время все большее количество людей во все большей степени зависят от компьютеров. В свое время человек открыл для себя огонь, и теперь огонь стал жизненно необходим ему. Точно так же после появления компьютеров и телекоммуникаций наша жизнь во все большей степени зависит от компьютерных технологий. Никто не станет спорить с тем, что некоторые крупные предприятия целиком и полностью зависят от Интернета (на ум немедленно приходит всемирно известная служба электронной коммерции Amazon.com). Интернет — это не что иное, как набор соединенных между собой компьютеров. Таким образом, получается, что огромное количество людей и многочисленные предприятия в огромной степени зависят от того, насколько хорошо защищены эти компьютеры.

ПРИМЕЧАНИЕ

В данной книге в первую очередь обсуждается система Caldera OpenLinux (эта система записана также на прилагаемый к книге компакт-диск), однако большая часть сведений применима также и к любым другим разновидностям Linux. Если некоторая информация относится только к Caldera или только к ограниченному набору разновидностей Linux, об этом упоминается особо.

Что такое безопасность

Когда вы начинаете изучать компьютерную безопасность, у вас формируется особенное отношение к ней. В упомянутом ранее законе Мэрфи речь идет о солдатах на войне, однако вы не должны ошибаться на этот счет. С момента подключения вашего компьютера к Интернету вы становитесь либо другом, либо врагом по отношению ко всему остальному интернет-сообществу. Считайте, что начинается война, и вы переносите эту электронную войну в ваш дом или в ваш рабочий офис, а полем битвы является ваша компьютерная система или сеть.

Вы можете относиться к безопасности по-разному, однако игнорировать ее нельзя. Если вы закроете на нее глаза в надежде, что все обойдется, вы наверняка проиграете войну. Однако бояться ее тоже не стоит — компьютерная безопасность — это вовсе не черная магия. Вам потребуется небольшой набор начальных знаний и побольше любознательности. Вы должны задавать себе вопросы: какой механизм здесь работает, как и почему. Вы должны представить себя на месте злоумышленника. Если бы вы были взломщиком, как вы смогли бы проникнуть в систему? Что бы вы стали делать, оказавшись внутри? Помните, что ваша задача проще: вы знаете о вашей системе и о ее уязвимых местах больше, чем те, кто пытается «прощупать» ее снаружи.

Анализ задач безопасности

Анализ существующих рисков, возможно, является наиболее сложной частью вашей работы. Прежде всего следует определить, что именно подвержено риску. Вы можете рассматривать вашу информационную инфраструктуру несколькими разными способами. В частности, информационные системы можно рассматривать как комбинацию трех составных частей: оборудование, программное обеспечение и данные. Каждая из этих составляющих подвергается трем угрозам: фактическое разрушение, неавторизированная модификация и неавторизированное использование. Рассмотрим каждую из этих угроз подробнее.

Оборудование — это то, до чего вы можете физически дотронуться рукой, например монитор или центральный процессор. Сюда же следует отнести принадлежащие вам сетевые устройства, например маршрутизаторы, кабели и т. п. В эту же категорию включается нечто, чего нельзя потрогать, — доступная вам пропускная способность каналов связи. Пропускная способность включается в эту категорию потому, что ее сложно включить в какую-либо другую категорию.

Оборудование можно привести в негодность, то есть сломать. Это не обязательно должен быть удар молотком по системному блоку — некоторые программы обладают возможностью сжечь монитор, а иногда и графическую карту. Содержимое перепрограммируемой постоянной памяти (EEPROM) можно стереть. В среде Linux нет вируса, подобного вирусу CIH (Чернобыль), поражающему системы Windows и приводящему в негодность материнские платы, однако пользователь, обладающий привилегиями root, может запустить программу, которая окажет на систему аналогичный эффект.

Взломщику не обязательно обладать физическим доступом к системе для того, чтобы оказать на нее влияние или модифицировать ее. Используя удаленный доступ, злоумышленник может перевести сетевую карту Ethernet в режим прослушивания сети или нарушить процесс нормальной загрузки системы. Он может также внести изменения в таблицу маршрутизации, что приведет к изменению порядка функционирования сети.

Неавторизированное использование оборудования не только лишает вас возможности воспользоваться ресурсами, за которые вы платите деньги (производительность процессора, память, пропускная способность каналов связи), из-за подобного использования вы можете быть внесены в черный список. Ваш провайдер может отключить вас от сети за то, что действия злоумышленника, выполняемые от лица вашей системы, нарушают устанавливаемые провайдером условия обслуживания. Некоторые домены могут отказать вам в приеме почты потому, что ваш узел был использован для несанкционированной рассылки «грязной» почты. Узлы FTP и Web могут блокировать вас из-за того, что какой-то компьютерный хулиган воспользовался вашей системой для сканирования других систем. Все это может стать причиной в лучшем случае неудобств, а в худшем случае — судебных разбирательств.

Анализ трех упомянутых угроз в отношении программного обеспечения выполняется проще. Программы можно стереть — в результате вы теряете функциональность вашей системы до тех пор, пока не восстановите их. Программы можно также изменить таким образом, чтобы они функционировали не так,

как вы ожидаете. Например, программы, подвергшиеся модификации, могут обеспечивать несанкционированный доступ для неавторизованных пользователей или перехватывать пароли и передавать их злоумышленникам. Неавторизированное использование программ — это последняя из трех угроз в отношении программного обеспечения. Эта угроза прежде всего имеет отношение к использованию коммерческих программ с лицензионными ограничениями. Вы несете ответственность за выполнение условий лицензионного соглашения, связанного с использованием той или иной программы, однако злоумышленники могут стать причиной нарушения этих условий. В результате неавторизированного использования программ также расходуется производительность процессора, что неблагоприятно влияет на работу вашей системы.

Данные являются наиболее уязвимой составляющей вашей компьютерной системы. Из трех рассмотренных элементов данные сложнее всего заменить. Разрушение данных означает как минимум потерю времени, которое понадобится вам, чтобы восстановить данные из резервной копии и заново воссоздать те данные, которые вы не успели сохранить в резервной копии. Несанкционированное изменение данных может привести к получению неправильных результатов их обработки. Неавторизированное использование может стать причиной потери закрытых, частных и индивидуальных данных, передачу таких данных в руки тех, кто не имеет на них никаких прав.

Анализируя каждую из рассмотренных ранее угроз, вы должны думать как о внешних, так и о внутренних пользователях. Внутренние пользователи — это пользователи, которые обладают законным правом авторизированного доступа к вашим внутренним системам. Работая с системами, они могут превысить свои полномочия либо специально, либо непреднамеренно. Помните, что большая часть нежелательных действий выполняется внутренними пользователями непреднамеренно. Хорошая система безопасности способна уменьшить количество проблем, которые возникают в результате действий внутренних пользователей. На основании анализа единичного инцидента сложно судить, была ли это нечаянная неосторожность или злонамеренный акт, однако, изучив общий порядок дел, вы можете с большей уверенностью локализовать проблему.

Разработка политики безопасности

Идентифицировав основные угрозы, нависшие над вашей системой, вы можете приступить к разработке политики защиты. Общая политика защиты системы может быть либо открытой, либо закрытой. Если говорить более подробно, общая политика может утверждать одно из двух: либо разрешить все, что явно не запрещено, либо запретить все, что явно не разрешено.

В последующих главах будет рассказано о том, как политика безопасности воплощается в жизнь. Я расскажу вам о пользователях и группах, и вы сможете определить, что для вас приемлемее: использование группы по умолчанию либо частные пользовательские группы. Вы узнаете о сетях и службах. О том, как отключить службы, в которых вы не нуждаетесь, и о том, как убедиться в том, что функционирующие службы настроены корректно. Вы узнаете о том, как взглянуть на вашу систему извне и увидеть ее с точки зрения злоумышленника, пытающегося взломать вашу защиту. Вы также узнаете о том, как обнаружить вторжение и что делать для того, чтобы восстановить вашу систему и обезопасить ее от повторного вторжения. Вопрос об оценке убытков, связанных с вторжением, в данной книге не рассматривается, однако вы не должны забывать о возможных последствиях атаки. Вы должны оценить возможные потери времени и информации.

Из книги вы узнаете об использовании пакетных фильтров. Вы научитесь настраивать фильтры для отбрасывания, отклонения и, когда это необходимо, приема пакетов. Пакетные фильтры являются встроенным в Linux программным обеспечением брандмауэра. Здесь я хочу лишний раз подчеркнуть, что брандмауэр не является совершенным средством защиты. Брандмауэр не может гарантировать вам, что ни один злоумышленник никогда не проникнет в вашу систему. Брандмауэр просто дает вам дополнительное время, которое вы можете использовать для обнаружения атаки и соответствующих ответных действий. Если взломщик, действительно хорошо знающий свое дело, проникнет в вашу систему, его присутствие очень сложно будет обнаружить вплоть до того момента, когда будет слишком поздно. На самом деле злоумышленник может замаскировать свое присутствие так, что вы вообще не сможете его обнаружить, несмотря на то, что он будет продолжать использовать вашу систему. Время является как вашим другом, так и вашим врагом.

Однако помимо чисто технических вопросов, которые я затронул в предыдущих абзацах, вы не должны упускать из виду также некоторые другие весьма важные вопросы. В данной книге вы не найдете юридических советов. Для получения подобной информации будет лучше обратиться к профессиональному адвокату, обладающему опытом ведения дел в компьютерной области. Однако вы должны определить масштаб нанесенного вам ущерба. Намерены ли вы инициировать судебное

разбирательство? Обладаете ли вы достаточно убедительными уликами? Помните, что файлы журналов можно модифицировать, поэтому журналы взломанной системы нельзя считать достоверным источником сведений.

Если для преследования злоумышленника вы намерены обратиться в компетентные органы, то обладаете ли вы правом обращения в полицию? Входит ли это в вашу компетенцию? Если нет, то кто в вашей организации обладает таким правом? Как вы или ваше руководство намерены вести себя в подобной ситуации? Насколько чувствительна ваша организация к последующей огласке возникшего инцидента?

Вы также должны составить хотя бы приблизительный план действий на случай возникновения подобной чрезвычайной ситуации. Планируете ли вы немедленно блокировать действия нарушителя и восстановить защиту системы или вы намерены не трогать злоумышленника в течение определенного времени и при этом попытаться определить адрес и физическое местоположение системы, с которой осуществляется атака? Будет ли у вас в запасе время, достаточное для слежения за взломщиками?

Ваша политика должна включать в себя также и другие правила, например, касательно паролей (через какое время пароли должны меняться и т. п.), правила обработки электронной почты (архивируете ли вы всю корпоративную почту или предписываете обязательное уничтожение всех писем через определенный период времени?). Очевидно, что если вы являетесь домашним пользователем, вы можете лишь поверхностно прикинуть приемлемые для вас ответы на эти вопросы, однако если вы работаете в организации, вы должны тщательно продумать и изложить на бумаге политику вашей сети, которая должна включать в себя упомянутые, равно как и многие другие, соображения. Дополнительную информацию по этой теме можно обнаружить в RFC-2196 и RFC-2504.

Часть I
Ваш узел

1 Пользователи, группы и безопасность

В данной главе рассматриваются следующие вопросы: - пользователи;

- различия между привилегированными и непривилегированными пользователями;
- файлы входа в систему;
- файл /etc/passwd;
- файл /etc/shadow;
- файл /etc/gshadow;
- файл /etc/login.defs;
- модификация сведений об устаревании паролей;
- PAM.

Linux не ограничивает вас одним-единственным способом сделать нечто, напротив, как правило, одну и ту же задачу можно решить несколькими способами. Какие-то из этих способов нельзя считать в полной мере правильными, выбор же среди остальных в общем случае является делом личных предпочтений. Некоторые способы существенно облегчают бремя администрирования, а это напрямую влияет на безопасность вашей системы. Чем больше времени уходит у вас на возню с учетными записями пользователей, тем меньше внимания вы уделяете вопросам безопасности.

В основе безопасности Linux лежат концепции пользователей и групп. Все решения о том, что разрешается или не разрешается делать пользователю, принимаются на основании того, кем является вошедший в систему пользователь с точки зрения ядра операционной системы.

Общий взгляд на пользователей

Linux является многозадачной многопользовательской системой. Это означает, что с этой системой могут работать много пользователей, причем одновременно. Естественно, эти пользователи не смогут одновременно работать с одной и той же клавиатурой и монитором, а вот войти в систему при помощи telnet, ftp, http и т. д. они вполне могут. В обязанности операционной системы входят изоляция и защита пользователей друг от друга. Система следит за каждым из пользователей и, исходя из того, кем является этот пользователь, определяет, можно ли предоставить ему доступ к тому или иному файлу или разрешить запуск той или иной программы.

При создании нового пользователя ему ставится в соответствие уникальное имя (которое иногда называют *регистрационным именем пользователя)*. Широко распространено ошибочное мнение, будто система следит за пользователем и определяет его привилегии исходя из его имени. Однако это не совсем так. Имя пользователя, безусловно, важно, но как вы увидите далее из этой главы, идентификатор пользователя еще важнее. Имя пользователя используется для удобства — запомнить символьное имя проще, чем запомнить численный идентификатор.

ПРИМЕЧАНИЕ

Система определяет привилегии пользователя на основании идентификатора пользователя (user ID, UID). В отличие от имени пользователя, UID может и не быть уникальным, в этом случае для сопоставления ему имени пользователя берется первое найденное имя, UID которого совпадает с данным.

Каждому новому регистрируемому в системе пользователю ставятся в соответствие определенные элементы системы. Обычно это домашний каталог и командная оболочка. Домашний каталог можно сравнить с комнатой в доме. Она отдается в полное распоряжение пользователя, который волен делать в ней все, что ему захочется. Однако за пределами комнаты его всевластие заканчивается. Будучи только гостем ъ доме, пользователь может наводить беспорядок или же убираться в других комнатах лишь с разрешения владельца дома, то есть операционной системы.

Привилегированные и непривилегированные пользователи

При добавлении нового пользователя в систему ему выделяется специальный номер, называемый *идентификатором пользователя* (user ID, UID). В Caldera OpenLinux выделение идентификаторов новым пользователям начинается с 500 и продолжается в сторону больших чисел, вплоть до 65 534. Номера до 500 зарезервированы для системных учетных записей. В других комплектах Linux может использоваться другое распределение системных и обычных идентификаторов: например, идентификаторы обычных пользователей могут начинаться с номера 1000, а не с 500. Номер идентификатора, на котором заканчиваются системные пользователи и начинаются пользователи обычные, определяется поставщиком комплекта Linux.

В общем и целом идентификаторы с номерами, меньшими 500, ничем не отличаются от остальных идентификаторов. На самом деле все номера от 1 до 65 534 равнозначны и не несут какой-либо предопределенной смысловой нагрузки. Тот факт, что номера до 500 зарезервированы для системы, просто является общепринятым соглашением. Часто программе для нормального функционирования требуется специальный пользователь с полным доступом ко всем файлам. Примером таких программ служат системы управления базами данных. В таком случае используется идентификатор из числа зарезервированных. Эти номера соответствуют непривилегированным учетным записям, пользователи которых не обладают никакими специальными привилегиями.

Однако нумерация идентификаторов начинается с 0 и продолжается до 65 535. UID 0 — это особенный UID. Любой процесс или пользователь с нулевым идентификатором является привилегированным. Такой человек или процесс имеет неограниченную власть над системой. Ничто не может служить для него запретом. Возвращаясь к аналогии с домом, учетная запись root (учетная запись, UID которой равен 0), также называемая учетной записью *суперпользователя,* делает вошедшего с ее использованием если не владельцем дома, то как минимум его доверенным лицом. Имея нулевой UID, можно бить окна, ломать стены или вообще поджечь весь дом — и никто не сможет воспрепятствовать этому. Администраторы со стажем знают, что работать с системой от имени этой учетной записи следует лишь в случае абсолютной необходимости и ровно столько, сколько нужно для выполнения потребовавших ее использования задач. То есть чем меньше вы работаете с системой от имени пользователя root, тем ниже риск вторжения в вашу систему. Далее мы еще вернемся к этому вопросу.

Остается UID, равный 65 535. Он тоже не из обычных. Этот UID принадлежит пользователю nobody *(никто)*. Прежде чем писать мне письма с обвинениями в некомпетентности, позвольте мне объясниться. Как можно заметить (а таковая возможность вскоре представится), в OpenLinux UID пользователя nobody равен 65 534. Номер 65 535 рассматривается операционной системой как UID, вообще не имеющий каких-либо привилегий. Создайте пользователя поопе, воспользовавшись программой useradd: useradd -m -u 65535 noone

Посмотрите внимательно на свойства домашнего каталога пользователя поопе. Видите странность: его владельцем является не поопе, как того можно было бы ожидать (команда useradd, создав домашний каталог пользователя, всегда делает его владельцем созданного каталога), а root? Если сказать su -noone, то тогда вы сможете читать файлы каталога, но не сможете создать или удалить ни одного файла, поскольку поопе не владеет ими. Дабы избежать сложностей с номером 65 535, пользователю nobody поставлен в соответствие номер 65 534, а 65 535 попросту не используется. (Не забудьте удалить пользователя поопе и его домашний каталог: userdel поопе; rm -Rf /home/noone.)

Когда-то одним из способов взлома системы было создание пользователя с идентификатором 65 536, в результате чего он получал привилегии суперпользователя. Действительно, если взять любой UID и перевести соответствующее число в двоичную форму, то получится комбинация из шестнадцати двоичных разрядов, каждый из которых равен либо 0, либо 1. Подавляющее количество идентификаторов включают в себя как нули, так и единицы. Исключением является нулевой UID суперпользователя, состоящий из одних нулей, и UID nobody, равный 65535 и состоящий из 16 единиц, то есть 1111111111111111. Число 65 536 нельзя разместить в 16 разрядах — для представления этого числа в двоичной форме требуется использовать уже 17 разрядов. Самый старший разряд будет равен единице (1), все остальные равны нулю (0). Так что же происходит при создании пользователя с идентификатором длиной в 17 двоичных разрядов — 10000000000000000? Теоретически, пользователь с нулевым идентификатором: поскольку под идентификатор отводится лишь 16 двоичных разрядов, 17 разряд хранить негде, и он отбрасывается. Стало быть, единственная единица идентификатора теряется, и остаются одни нули, а в системе появляется новый пользователь с идентификатором, а значит, и привилегиями, суперпользователя. Но это было давно, теперь же в Linux нет программ, которые позволили бы вам установить UID в 65 536. Так просто современные системы уже не взломаешь. Тогда какая польза от всего этого разговора? Большая, если преуспеть в замене программы, не позволяющей устанавливать этот UID, на программу, которая позволяет. Кроме того, как мы еще увидим, подобный метод взлома можно использовать и в других ситуациях.

ПРИМЕЧАНИЕ

Пользователей с идентификаторами, превышающими 65 536, создавать можно, но использовать их без подмены /bin/login не получится.

Любой взломщик обязательно постарается получить привилегии суперпользователя. Как только он их получит, дальнейшая судьба вашей системы полностью будет зависеть от его намерений. Возможно, он, удовлетворившись самим фактом взлома, не сделает с ней ничего плохого и, послав вам письмо с описанием найденных им дыр в системе безопасности, навсегда оставит ее в покое, а возможно, и нет. Если намерения взломавшего вас хакера не столь чисты, то тогда лучшее, на что можно надеяться, — это выведение системы из строя. Вы наверняка слышали о подменах хакерами web-страниц известных сайтов на свои страницы, так вот это лишь одна из возможных игр. Ваша система может быть нужна злоумышленнику в качестве прикрытия для проведения с нее атак на другие системы, чтобы все следы этих атак вели к вашей системе. На скольких языках вы можете произнести словосочетание «судебное разбирательство»? Как вы думаете, почему случаи поимки властями взломщиков, причинивших серьезный ущерб, столь редки? В том числе и потому, что они используют для взлома чужие системы и тщательно заметают все следы к своей системе.

Файл /etc/passwd

Желающий войти в систему должен ввести имя пользователя и пароль, которые проверяются по базе данных пользователей, хранящейся в файле /etc/passwd. В нем, кроме всего прочего, хранятся пароли всех пользователей (при использовании теневых паролей из него можно узнать, в каком другом файле они хранятся). При подключении к системе введенный пароль сверяется с паролем, соответствующим данному имени, и в случае совпадения пользователь допускается в систему, после чего запускается программа, указанная для данного имени пользователя в файле паролей. Если это командная оболочка, пользователь получает возможность вводить команды.

Чтобы понять, как это работает, давайте посмотрим на файл passwd. Находится он в каталоге /etc и может быть прочитан кем угодно. Последнее необходимо, чтобы могли работать программы, проверяющие имена и идентификаторы пользователей. Когда-то в этом файле хранились пароли пользователей, отсюда его имя. Взгляните на листинг 1.1. Это файл passwd в старом стиле. Давайте пройдемся по всем его полям и выясним их назначение.

Листинг 1.1. Файл /etc/passwd в старом стиле

```
root::1i DYwrOmhmEBU: 0:0: root:: /root: /bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/adm:
lp:*:4:7:lp:/var/spool/lpd:
sync:*:5:0:sync:/sbin:/bin/sync
shutdown:*:6:11:shutdown:/sbin:/sbin/shutdown
halt:*:7:0:halt:/sbin:/sbin/halt
mail:*:8:12:mail:/var/spool/mail:
news:*:9:13:news:/var/spool/news:
uucp:*:10:14:uucp:/var/spool/uucp:
operator:*:11:0:operator:/root:
games:*:12:100:games:/usr/games:
gopher:*:13:30:gopher:/usr/1ib/gopher-data:
ftp:*:14:50:FTP User:/home/ftp:
man:*:15:15:Manuals Owner:/:
majordom:*:16:16:Majordomo:/:/bin/false
postgres:*:17:17:Postgres    User:/home/postgres:/bin/bash
mysql:*:18:18:MySQL    User:/usr/local/var:/bin/false
silvia:1iDYwrOmhmEBU:501:501:Silvia    Bandel:/home/silvia:/bin/bash
nobody:*:65534:65534:Nobody:/:/bi n/false
david:1iDYwrOmhmEBU:500:500:David   A.   Bandel :/home/david:/bin/bash
```

Файл паролей имеет жестко заданную структуру. Можно заметить, что содержимое файла представляет собой таблицу. Каждая строка файла — это запись таблицы. Каждая запись состоит из нескольких полей. Поля файла passwd, как принято и в некоторых других служебных таблицах в Unix, разделяются двоеточием, поэтому двоеточия нельзя использовать ни в одном из полей. Всего имеется семь полей: имя пользователя, пароль, идентификатор пользователя, идентификатор группы, поле GECOS (оно же поле комментариев), домашний каталог и командная оболочка входа в систему.

Подробнее о /etc/passwd

В первом поле указывается имя пользователя. Оно должно быть уникальным — нельзя, чтобы два пользователя системы имели одно и то же имя. Если вы попробуете добавить в файл двух пользователей с одинаковыми именами (для этого вам придется вручную отредактировать файл, так как программы useradd, coastool и LISA не позволят вам сделать этого), то любая программа, начав поиск по имени пользователя, закончит его на первом из этих двух имен и никогда не доберется до второго. А значит, второй пользователь просто не сможет войти в систему, поскольку при входе всегда будет проверяться пароль и назначаться UID первого пользователя. Стало быть, второй пользователь как бы и не существует. Поле имени является единственным полем, значение которого должно быть уникальным. Во втором поле хранится пароль пользователя. Для того чтобы обеспечить защиту системы, пароль хранится в хэшированном виде. Термин «хэширован-ный» в данном контексте означает «зашифрованный». В случае с Linux пароль шифруется по алгоритму DES (Data Encryption Standard). Длина хэшированно-го пароля в этом поле всегда равна 13 символам, причем некоторые из символов, такие как двоеточие и одинарная кавычка, никогда не встречаются среди них (см. раздел «Файл /etc/shadow»). Любое другое значение поля, отличное от правильного хэшированного 13-символьного пароля, делает невозможным вход данного пользователя в систему, за одним чрезвычайно важным исключением: поле пароля может быть пустым. Например, одна из записей файла /etc/passwd может выглядеть следующим образом:

```
david::500:500:David A. Bandel:/home/david:/bin/bash
```

Во втором поле не стоит ничего, даже пробела, это означает, что соответствующему пользователю не нужен пароль для входа в систему. Другими словами, данный пользователь может войти в систему, не указывая вообще никакого пароля. Скорее всего, это не то, что вам нужно. Поэтому файл /etc/passwd всегда следует проверять на наличие записей с пустыми паролями. Если изменить пароль, хранящийся в поле, добавив к нему какой-либо символ, например одинарную кавычку, то данная учетная запись окажется заблокированной, а соответствующий пользователь более не сможет войти в систему. Дело в том, что после добавления в 14-символьный хэшированный пароль нелегального символа система отказывалась аутентифицировать пользователя с таким паролем. Это достаточно старый прием, и в настоящее время его практически не используют. Современные системы для подключения пользователей к системе используют механизм /etc/shadow (о нем чуть позже), который предусматривает более правильные способы блокирования учетной записи без изменения пароля. Однако во времена, когда механизм теневых паролей еще не был разработан, системные администраторы, в чьем ведении находились системы с большим числом пользователей (а когда пользователей много, всегда найдутся те, кто не прочь попытаться взломать чего-нибудь на досуге), зачастую сами блокировали пароль пользователя root, помещая одинарную кавычку перед первым его символом. Вместо заблокированной таким образом учетной записи заводилась другая учетная запись с привилегиями суперпользователя, которой, как правило, назначали имя toor или tuber.

В настоящее время длина пароля ограничена восьмью символами. Пользователь может вводить и более длинные пароли, однако значимыми будут только первые восемь символов. Первые два символа хэшированного пароля являются *затравкой* (salt). (Затравкой называется число, используемое для инициализации алгоритма шифрования. При каждой смене пароля затравка выбирается случайным образом.) В результате число всех возможных перестановок достаточно велико, поэтому выяснить, есть ли в системе пользователи с одинаковыми пароля-ми, простым сравнением хэшнрованных паролей нельзя. Вообще говоря, алгоритм шифрования паролей обладает любопытным свойством. Конечно же, вы не сможете расшифровать зашифрованный пароль, однако вы можете взять любое слово, зашифровать его и сравнить с зашифрованным паролем. Из-за этой особенности алгоритма шифрования весьма распространенным способом взлома хэширован-ного пароля является атака по словарю.

ПРИМЕЧАНИЕ

Атака по словарю (dictionary attack) относится к методам взлома паролей грубой силой и подразумевает использование словаря и известной затравки. Атака состоит в переборе всех слов словаря, шифрования их с данной затравкой и сравнении результата со взламываемым паролем. При этом кроме слов из словаря обычно рассматриваются и некоторые их модификации, например, все буквы заглавные, только первая буква заглавная и добавление чисел (обычно только 0-9) в конец всех этих комбинаций. Подобным образом можно взломать достаточно много легко угадываемых паролей.

Из листинга 1.1 видно, что у пользователей root, silvia и david один и тот же хэ-шированный пароль. Это сделано мною преднамеренно, на практике такого не происходит. Признаюсь сразу же: так выглядит хэшированный пароль «silvia». Подобная очевидность пароля (особенно для пользователя silvia) ничем не лучше отсутствия пароля как такового. Столь простой пароль находится любой приличной программой взлома примерно за 3 наносекунды, однако я уже сообщил вам его, так что можете считать, что я сэкономил вам время.

ССЫЛКА
Об использовании паролей в системе безопасности рассказывается в главе 2.

В третьем поле указывается идентификатор пользователя. Об идентификаторах уже говорилось ранее, поэтому не буду повторяться. Идентификатор пользователя не обязан быть уникальным, как думают некоторые. В частности, кроме пользователя root, может быть сколь угодно других пользователей с нулевым идентификатором, и все они будут обладать привилегиями суперпользователя. Таким образом, в примере, что я приводил три абзаца тому назад, добавление нового пользователя для целей администрирования делалось именно путем присвоения ему (пользователю toor, или tuber, или какому-нибудь другому) нулевого идентификатора.

Однако данный подход не свободен от определенных проблем. Предположим, вы так и сделали: создали нового пользователя с нулевым идентификатором и заблокировали пользователя root недопустимым символом в хэшированном пароле. После чего вошли в систему уже не как root, а как новый пользователь. А войдя и поработав некоторое время, решили отлучиться на несколько минут и, не желая выходить из системы, не подумали и заблокировали терминал. Досадно, конечно, но разблокировать терминал по возвращении вам не удастся. Причина в том, что программе блокировки терминала важен лишь UID пользователя, заблокировавшего терминал, который в данном случае равен нулю. При поиске пользователя с нулевым идентификатором первым будет найден заблокированный пользователь root. Однако ничего не знающая об этом программа блокировки не станет смотреть далее и запросит пароль пользователя root Тупик. Итак, у данного подхода есть свои тонкости, не забывайте про них. Во второй части этой книги, когда речь пойдет о безопасности NFS, мы еще вернемся к вопросу взаимосвязей между идентификаторами и именем пользователя.

Четвертое поле содержит идентификатор группы (Group ID, GID). Группа, указанная в этом поле, называется *первичной группой пользователя* (primary group). Пользователь может принадлежать к нескольким группам, но одна из них обязательно должна быть первичной группой. Более подробно о группах будет рассказано в этой и в следующих двух главах.

Пятое поле теперь называют полем комментариев, но первоначальное его название — GECOS, от «GE Consolidated Operating System». При запросе информации о пользователе через finger или иную программу содержимое данного поля теперь возвращается как истинное имя пользователя. Поле комментариев может быть пустым.

Шестое поле задает домашний каталог пользователя. У каждого пользователя должен быть свой домашний каталог. Обычно пользователь, войдя в систему, оказывается в своем домашнем каталоге, но если такового не существует, то он попадает в корневой каталог. Назначая домашний каталог для некоторых учетных записей, следует быть внимательным. Например, домашним каталогом пользователя ftp следует сделать корневой каталог дерева каталогов, доступных через анонимный ftp, а не просто корневой каталог.

Седьмое и последнее поле задает командную оболочку входа в систему. Не всякую оболочку можно указать в этом поле. В зависимости от настроек системы в нем может быть указана только оболочка из списка допустимых оболочек. В OpenLinux список допустимых оболочек находится по умолчанию в файле /etc/shells (см. обсуждение PAM далее в этой главе).

Как упоминалось ранее, время, когда пароли пользователей хранились в файле /etc/passwd, безвозвратно ушло в прошлое. Даже несколько лет назад вычислительных мощностей компьютера хватало на то, чтобы, сравнив хэшированный словарь с копией файла passwd, взломать приличное количество паролей, включая суперпользователя, всего за несколько дней, а то и часов. Теперь подобная процедура занимает всего несколько минут. Поэтому возникла потребность в ограничении доступа к хэшированным паролям. Появился механизм теневых паролей.

Файл /etc/shadow

Владельцем файла /etc/shadow является пользователь root и только он имеет право читать этот файл. Для его создания нужно взять имена пользователей и хэши-рованные пароли из файла passwd и поместить их в файл shadow, заменив при этом все хэшированные пароли в файле passwd символами x. Если вы посмотрите на файл passwd вашей системы, то увидите, что на месте хэшированных паролей там стоят символы x. Данный символ указывает системе на то, что пароль следует смотреть не здесь, а в файле /etc/shadow. Переход от простых паролей к теневым и обратно осуществляется посредством трех утилит. Для перехода к теневым паролям сначала запустите утилиту pwck. Она проверяет файл passwd на предмет всяких аномалий, из-за которых следующий шаг может закончиться неудачей или попросту зациклиться. После того как отработает pwck, запустите утилиту pwconv для создания /etc/shadow. Обычно это делается после ручного обновления файла /etc/passwd. Для возвращения к обычным паролям запустите pwuncov.

Файл теневых паролей во многих отношениях схож с файлом обычных паролей. В частности, первые

два поля этих файлов одинаковы. Но помимо этих полей в нем, естественно, есть и дополнительные поля, отсутствующие в файле обычных паролей. Листинг 1.2. показывает содержимое типичного файла /etc/shadow.

Листинг 1.2. Файл /etc/shadow

```
root:1iDYwrOmhmEBU:10792:0:: 7:7::
bin:*:10547:0::7:7::
daemon:*:10547:0::7:7::
adm:*:10547:0::7:7::
lp:*:10547:0::7:7::
sync:*:10547:0::7:7::
shutdown:U:10811:0:-1:7:7:-1:134531940
halt:*:10547:0::7:7::
mail:*:10547:0::7:7::
news:*:10547:0::7:7::
uucp:*:10547:0::7:7::
operator:*:10547:0::7:7::
games:*: 10547:0: :7:7::
gopher:*:10547:0::7:7::
ftp:*:10547:0::7:7::
man:*:10547:0::7:7::
majordom:*:10547:0::7:7::
postgres:*:10547:0::7:7::
mysql:*:10547:0::7:7::
si1via:1iDYwrOmhmEBU:10792:0:30:7:-l::
nobody:*:10547:0::7:7::
david:1iDYwrOmhmEBU:10792:0::7:7::
```

Подробнее о /etc/shadow

Назначение первого поля файла shadow такое же, как и у первого поля файла passwd.

Второе поле содержит хэшированный пароль. Реализация теневых паролей в OpenLinux допускает хэшированные пароли длиной от 13 до 24 символов, однако программа шифрования паролей crypt умеет выдавать только 13-символь-ные хэшированные пароли. Символы, используемые в хэше, берутся из набора, состоящего из 52 букв алфавита (строчных и прописных), цифр 0-9, точки и наклонной черты вправо (/). Итого выходит 64 символа, допустимых в поле хэши-рованного пароля.

Затравка, таким образом, которая, как и ранее, представляет собой первые два символа, может выбираться из 4096 возможных комбинаций (64x64). Для шифрования используется алгоритм DES с 56-битным ключом, то есть пространство ключей этого алгоритма насчитывает 2^{56} ключей, что приблизительно равно 72 057 590 000 000 000 или 72 квадрильонам. Само по себе число может выглядеть впечатляющее, однако перебрать все ключи из пространства такого размера можно на самом деле за весьма короткое время. Относительно недавно совместными усилиями компьютеров, разбросанных по всему миру, подобный 56-битный код был взломан менее чем за два часа. (Время от времени в Интернете проводятся специальные акции по расшифровке сообщения, зашифрованного определенным алгоритмом, с целью проверки его криптостойкости, принять участие в которых может любой желающий, согласный выделить часть вычислительных мощностей своего компьютера для перебора определенной части пространства ключей.) Коль скоро речь зашла о переборе, замечу, что программы, реализующие атаку по словарю, обычно перебирают лишь ту малую часть пространства ключей, которая используется людьми (то есть слова из словаря и их модификации), причем с потрясающей скоростью. Поэтому единственная вещь, делающая пароли действительно защищенными, это невозможность прочтения файла паролей никем, кроме суперпользователя.

С третьего поля начинается информация об устаревании пароля (см. раздел «Изменение информации об устаревании пароля» далее в этой главе). В нем хранится число дней, прошедших с 1 января 1970 года до дня последнего изменения пароля. В табл. 1.1. показаны числа, которые последовательно появятся в этом поле при изменении пароля первого числа каждого месяца в течение 2000-2005 годов.

Таблица 1.1. Значения, которые можно наблюдать в третьем поле файла shadow, если менять пароль первого числа каждого месяца

Месяц/Год	2000	2001	2002	2003	2004	2005
Январь (31)	10957	11323	11688	12053	12418	12784
Февраль (28/29)	10988	11354	11719	12084	12449	12815
Март (31)	11017	11382	11747	12112	12478	12843
Апрель (30)	11048	11413	11778	12143	12509	12874
Май(31)	11078	11443	11808	12173	12539	12904
Июнь(30)	11109	11474	11839	12204	12570	12935
Июль (31)	11139	11504	11869	12234	12600	12965
Август(31)	11170	11535	11900	12265	12631	12996

Сентябрь(30)	11201	11566	11931	12296	12622	13027
Октябрь (31)	11231	11596	11961	12326	12692	13057
Ноябрь(30)	11262	11627	11992	12357	12723	13088
Декабрь (31)	11292	11657	12022	12387	12753	13118

Четвертое поле задает минимальное число дней, которые должны пройти, прежде чем можно будет вновь изменять пароль. Пока со дня последнего изменения пароля не пройдет столько дней, сколько указанно в этом поле, вновь изменять пароль нельзя.

Пятое поле задает максимальное число дней, в течение которых можно использовать пароль, после чего он подлежит обязательной смене. При положительном значении этого поля попытка пользователя войти в систему после истечения срока действия пароля приведет к тому, что команда password будет запущена не как обычно, а в режиме обязательной смены пароля.

Значение из шестого поля определяет, за сколько дней до окончания срока действия пароля следует начать выдавать предупреждение об этом. Получив предупреждение, пользователь может начать придумывать новый пароль, который был бы легко запоминаем и достаточно надежен с точки зрения безопасности.

Седьмое поле задает число дней, начиная со дня обязательной смены пароля, по истечении которых данная учетная запись блокируется. Иначе говоря, если по прошествии указанного количества дней, отсчитываемых со дня обязательной смены пароля, пользователь не зайдет в систему и не изменит свой пароль, то его учетная запись будет заблокирована.

В предпоследнем поле хранится день блокировки учетной записи.

Последнее поле зарезервировано и не используется.

Последнее замечание

И еще одна вещь, прежде чем мы двинемся дальше. В файлах passwd и shadow вы можете встретить записи, подобные этой:

```
+:x:0:0:::
ИЛИ +:*:0:0::-1:-1::
```

Такие записи могут встречаться лишь в самом конце этих файлов, после всех обычных записей. Если вы не используете службу NIS (Network Information Services, ранее эта служба называлась «Yellow Pages», однако имя было изменено из-за конфликта с торговой маркой British Telecom), то их следует удалить. Тем, кому очень хочется или необходимо использовать NIS, следует знать, что NIS является чрезвычайно небезопасным протоколом со слаборазвитой системой авторизации. NIS+ несколько лучше своего предшественника, однако реализации сервера NIS+ для Linux (пока) нет. Работать с NIS означает подвергать свою систему большому риску, от которого лучше воздержаться. Мы больше не будем обсуждать здесь проблемы безопасности NIS, оставив их для отдельной книги.

Файл /etc/groups

Файл групп по своей природе похож на файл паролей. Из последующих глав вы узнаете больше о значимости файла групп, в этой же главе мы ограничимся обсуждением его структуры. Пример файла групп приведен в листинге 1.3.

Листинг 1.3. Пример файла /etc/group

```
root::0:
wheel::10:
bin::1:bin,daemon
daemon::2:bin,daemon
sys::3:bin,adm
adm::4:adm,daemon
tty::5:
disk::6:
lp::7:daemon,lp
mem::8:
kmem::9:
operator::11:
mail::12:mail
news:: 13: news
uucp::14:uucp
man::15:
majordom::16:
database::17:
```

```
mysql::18:
games::20:
gopher::30:
dip::40:
utmp::45:
ftp::50:
silvia::501:silvia  nobody::65534:  users::100:david,silvia
david::500:david
```

Подробнее о /etc/group

Каждая запись файла /etc/group состоит из четырех полей, разделенных двоеточиями. Первое поле задает имя группы. Подобно имени пользователя, оно облегчает жизнь системных администраторов (символьные имена запоминаются куда легче числовых идентификаторов).

Второе поле обычно всегда пустое, так как механизм паролей для групп обычно не используется, однако если данное поле не пусто и содержит пароль, то к группе может присоединиться любой пользователь. Для этого нужно выполнить команду newgrp с именем группы в качестве параметра, после чего ввести правильный пароль. Могу предположить, что некоторые из читателей скажут: «Ха, вот автор и попался. У группы root поле пароля пустое, а значит, кто угодно может присоединиться к ней, не вводя никакого пароля. И где здесь безопасность?» Не совсем так. Если пароль для группы не задан, то присоединиться к ней могут только пользователи, перечисленные в списке членов группы (см. описание четвертого поля чуть далее).

Третье поле задает идентификатор группы (Group ID, GID). Смысл его такой же, как и у идентификатора пользователя. Удобней, когда он уникальный, хотя это и необязательно.

Последнее поле представляет собой список имен пользователей, принадлежащих к группе (пробелы между запятой и первым или последним символом имени обычно игнорируются, но для гарантированной работы всех программ лучше не допускать их появления). Имена пользователей перечисляются через запятую без пробелов. Указывать пользователя в списке членов его первичной группе не обязательно, однако это не повредит и может уберечь от некоторой путаницы в дальнейшем. Первичная группа пользователя указывается (в обязательном порядке) в файле passwd и назначается при подключении пользователя к системе исходя из этой информации. Соответственно, если изменить первичную группу пользователя в файле passwd, то пользователь более не сможет присоединиться к своей бывшей первичной группе.

Файл /etc/gshadow

По умолчанию OpenLinux устанавливается без поддержки теневых групповых паролей. Это потому, что хороший системный администратор групповые пароли не использует. Причина проста: групповые пароли трудно хранить в секрете. Если вы практикуете групповые пароли, то можете справедливо полагать, что они известны всем. Важность неразглашения личного пароля до пользователя донести легко, но вот хранить в секрете пароль «который итак все уже знают» пользователи очень быстро устают. Прибавьте к этому потребность пользователей в легко запоминаемых групповых паролях, и вы получите систему, о безопасности которой говорить не приходится.

Но если вы согласны оставить в стороне осторожность и хотите задействовать механизм групповых паролей, то следует использовать теневые, а не обычные пароли. Причины те же, что упоминались при обсуждении теневых паролей для пользователей.

Для использования групповых теневых паролей нужно создать файл gshadow. Переход от системы без теневых групповых паролей (без файла gshadow) к системе с таковыми (с файлом gshadow) и обратно осуществляется аналогично тому, как это делается для файла passwd. Предосторожности ради сначала запустите утилиту grpck, которая проверит файл /etc/group на наличие ошибок, не позволяющих перейти к следующему шагу. Далее запустите утилиту grpconv. После ее выполнения в файле /etc/group в поле хэшированного пароля появится уже знакомый вам символ x, а сами пароли перекочуют в созданный ею файл /etc/gshadow. Записи файла gshadow состоят из четырех полей, разделенных запятой. Первое поле задает имя группы, второе — хэшированный пароль (вся изложенная ранее информация о хэшированных паролях остается верной и для групповых паролей). Третье поле зарезервировано. Четвертое поле представляет собой список членов группы. Подобно файлу shadow, для возвращения к обычному файлу групп запустите grpuncov.

Файл /etc/login.defs

Добавить нового пользователя в систему можно несколькими способами. В Open-Linux для этого используются следующие программы: новая программа coastooL, более старая программа LISA, а также программа useradd. Подойдет любая из них. Утилита COAS использует свой собственный файл, который будет рассмотрен в дальнейшем. А программы useradd и LISA берут информацию о значениях по умолчанию для полей файлов passwd и shadow из файла /etc/login.defs. Содержимое этого файла в сокращенной форме показано в листинге 1.4 (комментарии переведены на русский язык).

Листинг 1.4. Сокращенный файл /etc/login.defs

```
# Максимальное количество дней, в течение которого разрешается использовать пароль:
# (-1 - смена пароля не обязательна)
PASS_MAX_DAYS -1
Минимальное количество дней между сменами пароля:
PASS_MIN_DAYS O
# За какое количество дней до даты смены пароля должно выдаваться предупреждение:
PASS_WARN_AGE 7
# Какое количество дней должно пройти после истечения допустимого срока
# использования пароля, прежде чем учетная запись будет блокирована:
PASS_INACTIVE -1
# Форсировать истечение срока использования пароля в заданный день:
# (дата идентифицируется количеством дней после 70/1/1, -1 = не форсировать)
PASS_EXPIRE -1
###

# Значения полей создаваемой учетной записи для программы useradd
# группа по умолчанию:
GROUP 100

# домашний каталог пользователя: %s = имя пользователя)
HOME /home/%s
# командная оболочка по умолчанию:
SHELL /bin/bash
# каталог, в котором расположен скелет домашнего каталога:
SKEL /etc/skel
# минимальное и максимальное значения для автоматического выбора gid в groupadd
GID_MIN 100
GID_MAX 60000
```

Содержимое этого файла задает значения по умолчанию для полей файлов passwd и shadow. Если не переопределить их из командной строки, будут использованы именно они. Как отправная точка, эти значения вполне подойдут, однако для реализации устаревания паролей некоторые из них нужно будет изменить. Значение, равное -1, означает отсутствие ограничений. Не бойтесь подстраивать первоначальные значения под собственные нужды. Но какие значения следует использовать? Это постепенно станет ясно из этой и следующей глав.

В программе COAS дистрибутива Caldera используется графический интерфейс пользователя (для консоли текстовый, но все равно основанный на кнопках и системах меню). Для администрирования учетных записей и групп выберите System Administration (администрирование системы) > Account Administration (администрирование учетных записей) и далее используйте ниспадающие меню.

Изменение информации об устаревании пароля

Для изменения информации об устаревании пароля для одного или двух пользователей можно воспользоваться командой chage (change aging — изменить устаревание). Непривилегированные пользователи могут запускать chage только с параметрами -l и собственным именем пользователя, то есть запрашивать информацию об устаревании только собственного пароля. Для изменения информации об устаревании достаточно (и необходимо) указать имя пользователя, остальные параметры будут запрошены в диалоговом режиме. Вызов chage без параметров выдаст краткую справку об использовании.

В качестве примера рассмотрим листинг 1.5. Напомню, что значения даются в днях.

Листинг 1.5. Информация об устаревании пароля пользователя david

```
# chage -1 david
Minimum: 2
Maximum: 90
```

```
Warning:  7
Inactive:  14
Last Change:          Aug 11, 1999
Password Expires:     Nov 09,  1999
Password Inactive:    Nov 23, 1999
Account Expires:      Never
```

Из листинга 1.5 видно, что пользователь david сможет сменить пароль лишь по прошествии двух дней со дня его последней смены. Количество дней, по истечении которых пользователь будет вынужден сменить пароль, равно 90. Предупреждения о скором устаревании пароля он будет получать в течение последних семи дней из этих девяноста. Если пользователь не сменит свой пароль в 14-дневный период с момента, когда срок использования пароля истечет, учетная запись этого пользователя будет заблокирована, и он уже не сможет изменить пароль (равно как и войти в систему).

Рассмотрим ситуацию со временными сотрудниками, нанятыми лишь на летний период. Во-первых, датой устаревания паролей всех таких учетных записей следует сделать 1 сентября. Далее можно отключить возможность смены пароля, сделав минимально допустимое число дней до смены пароля большим, чем максимально допустимый возраст пароля. Поскольку менять пароль таким пользователям не придется, то можно отключить (установив в ноль) и вывод предупреждения. Как видите, манипулируя параметрами устаревания пароля, можно настраивать учетные записи под самые различные нужды.

ССЫЛКА

Более подробно об использовании механизма устаревания паролей будет рассказано в главе 2.

Программа COAS может использоваться с целью изменения параметров устаревания паролей для каждой из учетных записей по отдельности. При этом значения указываются в днях. Интерфейс программы очевиден.

ПРИМЕЧАНИЕ -

Для получения информации об устаревании пароля пользователя или форсирования этого процесса можно воспользоваться командой expiry.

Система безопасности PAM

Один из недостатков файла теневых паролей заключается в том, что все приложения, так или иначе работающие с паролями, должны быть (заново) скомпилированы с поддержкой теневых паролей, иначе они будут неработоспособны. Однако компания Sun предложила решение этой проблемы, впервые появившееся в операционной системе Solaris 2.3, которое затем было адаптировано и для Linux. Система называется PAM (Pluggable Authentication Modules) и состоит из модулей и библиотек. Приложению, компонованному с использованием этих библиотек, не нужно подстраиваться под систему авторизации, используемую в системе в данный момент, так как PAM берет все эти вопросы на себя. Такой подход делает систему намного более гибкой.

Основная идея PAM состоит в том, что всегда можно написать новый модуль безопасности, который бы обращался к файлу или устройству за информацией и возвращал результат выполнения процедуры авторизации: УСПЕХ (SUCCESS), НЕУДАЧА (FAILURE) или ИГНОРИРОВАТЬ (IGNORE). А PAM, в свою очередь, возвратит УСПЕХ (SUCCESS) или НЕУДАЧА (FAILURE) вызвавшей ее службе. Таким образом, неважно, какие пароли, теневые или обычные, используются в вашей системе, коль скоро в ней есть PAM: все поддерживающие PAM программы будут прекрасно работать и с теми и другими. Более того, вы можете сравнительно легко добавлять и другие механизмы авторизации. Допустим, у вас есть устройство распознавания речи, способное сравнивать голоса с эталоном из файла. Тогда чтобы получить авторизацию на основе распознавания речи, достаточно написать модуль, который бы сравнивал голоса с помощью устройства и возвращал одно из трех значений: УСПЕХ (SUCCESS), ИГНОРИРОВАТЬ (IGNORE) или НЕУДАЧА (FAILURE) в зависимости от результата. Аналогично можно добавить авторизацию на основе сканирования сетчатки глаза, смарт-карты и т. д. Дело лишь в соответствующем модуле. Модули можно комбинировать. Например так, чтобы для успешной авторизации требовалось прохождение нескольких процедур авторизации или же чтобы было достаточно прохождения любой из них. Тем самым можно создавать достаточно сложные системы авторизации: пароль, дополняемый сканированием сетчатки глаза, или же смарт-карта и распознавание голоса и т. д.

Перейдем теперь к рассмотрению основных принципов работы PAM. Взгляните на листинг 1.6. Это взятый из OpenLinux 2.3 файл конфигурации PAM для службы (такова терминология PAM) login. Кроме него в каталоге /etc/pam.d содержатся файлы конфигурации и для других служб, таких как su, passwd и т.

п., в зависимости от того, какое программное обеспечение установлено в системе. Каждой службе с ограничением доступа (restricted service) соответствует свой файл конфигурации. Если такового нет, то данная служба с ограничением доступа попадает в категорию «other», с файлом конфигурации other.d. (Службой с ограничением доступа называется любая служба или программа, для использования которой требуется пройти авторизацию. Иными словами, если при нормальных условиях служба запрашивает у вас имя пользователя и пароль, она является службой с ограничением доступа.)

Листинг 1.6. Файл конфигурации службы login

```
auth        required        pam_securetty.so
auth        required        pam_pwdb.so
auth        required        pam_nologin.so
#auth       required        pam_dialup.so
auth        optional        pam_mail.so
account     required        pam_pwdb.so
session     required        pam_pwdb.so
session     optional        pam_lastlog.so
password    required        pam_pwdb.so
```

Как видно из листинга, файл конфигурации состоит из трех столбцов. На самом деле всего их четыре, но последний столбец является необязательным и в данном случае не используется. А вообще он предназначен для передачи в модуль дополнительных параметров, но об этом в следующем примере. Строки, начинающиеся с символа решетки (#), игнорируются. Стало быть, модуль pam_dialup (четвертая строка листинга 1.6.) будет пропущен. Смысл остальных строк файла будет легче понять, если изучить диаграмму на рис. 1.1. Обратите внимание, в файле есть строки с одинаковым третьим полем — pam_pwd.so, и первым — auth. Использование нескольких строк с одинаковым первым полем называется накоплением (stacking) модулей и позволяет получать многошаговую авторизацию (стек модулей), включающую несколько различных процедур авторизации.

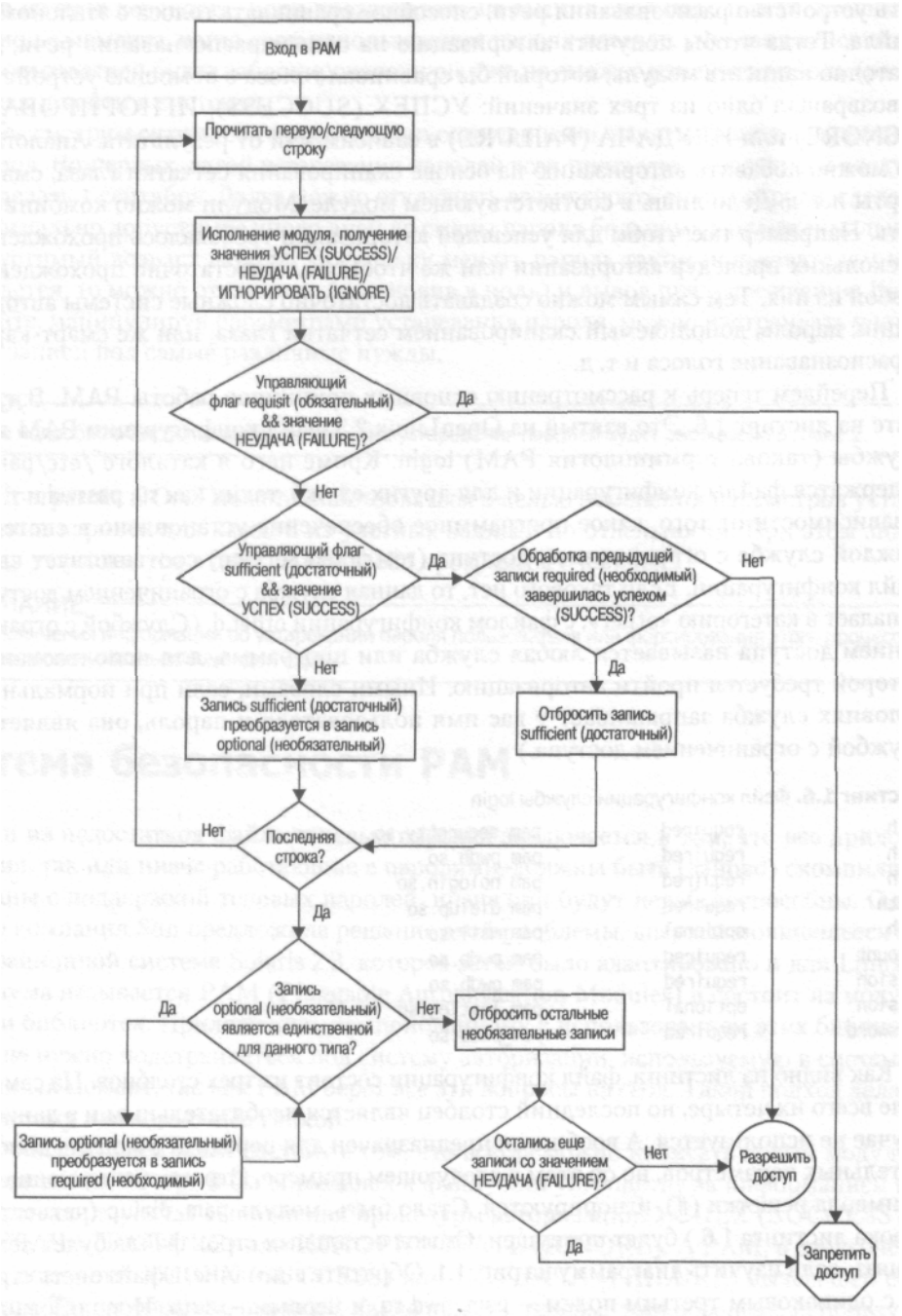

Рис. 1.1. Диаграмма использования модулей PAM

Типы модулей PAM

Первый столбец является столбцом типа. Тип определяется одной из четырех символьных меток: auth, account, session и password. Заметьте, что содержимое всех столбцов рассматривается без учета регистра.

- Тип auth (authentication — аутентификация) используется для выяснения, является ли пользователь тем, за кого он себя выдает. Как правило, это достигается сравнением введенного и хранимого паролей, но возможны и другие варианты. Например, при помощи смарт-карты или каким-либо иным способом, для которого имеется соответствующий модуль.

- Тип account (учетная запись) проверяет дозволено ли использовать службу данному пользователю, на каких условиях, не устарел ли пароль и т. д.

- Тип password (пароль) используется для обновления маркеров авторизации. Например, если согласно полю устаревания пароля из файла /etc/shadow необходима принудительная смена пароля, модуль passwd запустит chauthok (изменение маркера авторизации) и, если смена пароля пройдет успешно, вернет значение УСПЕХ (SUCCESS).

- Тип session (сеанс) выполняет определенные действия при входе пользователя в систему и при выходе пользователя из системы. Таковыми могут быть, например, монтирование и демонтирование каталогов, подготовка пользовательского окружения и т. д.

Управляющие флаги

Второй столбец является полем управляющего флага, определяющим, что делать после возврата из модуля, то есть реакцию PAM на значения УСПЕХ (SUCCESS), ИГНОРИРОВАТЬ (IGNORE) и НЕУДАЧА (FAILURE). Разрешенные значения: requisite, required, sufficient и optional. От значения в этом поле зависит, будут ли обработаны остальные строки файла. Например, если модуль, помеченный как sufficient (достаточный), вернет значение УСПЕХ (SUCCESS), то работа PAM на нем успешно закончится и пользователь будет допущен в систему. Аналогичным образом возврат НЕУДАЧА (FAILURE) модулем, помеченным как requisite (обязательный), означает прекращение авторизации и возврат значения НЕУДАЧА (FAILURE). И в том и в другом случае остальные строки обрабатываться не будут.

- Флаг requisite (обязательный) задает наиболее жесткое поведение. Обработка любой строки с флагом requisite, модуль которой вернул значение НЕУДАЧА (FAILURE), будет прекращена и вызвавшей ее службе будет возвращен статус НЕУДАЧА (FAILURE). Никакие другие строки рассматриваться не будут. Вообще говоря, этот флаг используется достаточно редко. Дело в том, что если помеченный им модуль выполняется самым первым, то следующие за ним модули могут и не выполниться, в том числе и отвечающие за протоколирование, поэтому вместо него обычно применяется флаг required (необходимый).

- Флаг required (необходимый) не прерывает выполнение модулей. Каков бы ни был результат выполнения помеченного им модуля: УСПЕХ (SUCCESS), ИГНОРИРОВАТЬ (IGNORE) или НЕУДАЧА (FAILURE), PAM всегда переходит к обработке следующего модуля. Это наиболее часто используемый флаг, так как результат выполнения модуля не возвращается до тех пор, пока не отработают все остальные модули, а значит, модули, отвечающие за протоколирование, обязательно выполнятся. Более подробно об этом рассказывается чуть далее.

- Флаг sufficient (достаточный) приводит к немедленному завершению обработки строки и возврату значения УСПЕХ (SUCCESS) при условии, что помеченный им модуль вернул значение УСПЕХ (SUCCESS) и ранее не встречалось модуля с флагом required, вернувшего статус НЕУДАЧА (FAILURE). Если такой модуль встречался, то флаг sufficient игнорируется. Если помеченный этим флагом модуль возвратил значение ИГНОРИРОВАТЬ (IGNORE) или НЕУДАЧА (FAILURE), то флаг sufficient рассматривается аналогично флагу optional (см. следующий параграф).

- Результат выполнения модуля с флагом optional (необязательный) принимается во внимание лишь тогда, когда он является единственным модулем в стеке, вернувшим значение УСПЕХ (SUCCESS). В противном случае результат его выполнения игнорируется. Таким образом, неуспешное выполнение помеченного им модуля не влечет за собой неуспех всего процесса авторизации.

Чтобы пользователь смог получить доступ к системе, модули, помеченные флагами requisite и required, не должны возвращать значения НЕУДАЧА (FAILURE). Помните, что результат выполнения модуля с флагом optional принимается в рассмотрение, лишь если он является единственным модулем в стеке, вернувшим УСПЕХ (SUCCESS).

Модули PAM

Третий столбец содержит полное имя файла модуля, связанного с данной строкой. В принципе, модули могут располагаться где угодно, однако если они размещены в предопределенном каталоге для модулей, то можно указывать одно лишь имя, в противном случае нужен еще и путь. В OpenLinux предопределенным каталогом является /lib/security.

Четвертый столбец предназначен для передачи в модуль дополнительных параметров. Не у всех модулей есть параметры, а если есть, то они могут и не использоваться. Передача параметра модулю позволяет изменить его поведение тем или иным образом.

Листинг 1.7 содержит список модулей PAM, входящих в состав OpenLinux.

Листинг 1.7. Список модулей PAM, входящих в состав OpenLinux

```
pam_access.so
pam_cracklib.so
pam_deny.so
pam_dialup.so
```

pam_env.so
pam_ftp.so
pam_group.so
pam_lastlog.so
pam_limits.so
pam_listfile.so
pam_mail.so
pam_nologin.so
pam_permit.so
pam_pwdb.so
pam_radius.so
pam_rhosts_auth.so
pam_rootok.so
pam_securetty.so
pam_shells.so
pam_stress.so
pam_tally.so
pam_time.so
pam_unix_acct.so
pam_unix_auth.so
pam_unix_passwd.so
pam_unix_session.so
pam_warn.so
pam_wheel.so

Сначала я коротко расскажу вам о некоторых наиболее общеупотребительных модулях, а затем на примерах их использования мы рассмотрим накопление модулей (module stacking).

О модулях подробнее

Модуль pam_access.so используется для предоставления/запрещения доступа на основании файла /etc/security/access.conf. Строки этого файла имеют следующий формат:

```
права: пользователи: откуда
```

Где:
- права — либо + (разрешить), либо - (запретить)

- пользователи — ALL, имя пользователя или пользователь@узел, где узел соответствует имени локальной машины, иначе запись игнорируется.

- откуда — одно или несколько имен файлов терминалов (без префикса /dev/), имена узлов, доменные имена (начинающиеся с точки), IP адреса (заканчивающиеся точкой, например, 192.168.0. соответствует всей сети 192.168.0.X), ALL или LOCAL.

Типовой файл конфигурации для этого модуля в OpenLinux по умолчанию отсутствует, однако приведенного выше (или же документации по нему) должно хватить для самостоятельного его написания. Параметров у этого модуля нет.

Модуль pam_cracklib.so проверяет пароли по словарю. Он предназначен для проверки нового пароля и позволяет предотвратить использование в системе легко взламываемых паролей, каковыми считаются общеупотребительные слова, пароли, содержащие повторяющиеся символы, и слишком короткие пароли. Есть необязательные параметры: debug, type= и retry=. Параметр debug включает занесение отладочной информации в файл журнала. Параметр type, за которым следует строка, меняет в выводимом по умолчанию приглашении New Unix password: слово Unix на указанную строку. Параметр retry задает число попыток, предоставляемых пользователю для ввода пароля, по исчерпании которых возвращается ошибка (по умолчанию дается одна попытка).

Теперь посмотрим на листинг 1.8. В нем показано содержимое файла /etc/ pam.d/other. В этом файле содержится конфигурация, используемая механизмом PAM в отношении служб, не имеющих своих собственных конфигурационных файлов в каталоге /etc/pam.d. Иными словами, этот файл применяется в отношении всех служб, неизвестных системе PAM. Как видите, в нем представлены все четыре типа авторизации, auth, account, password и session, каждая из которых вызывает помеченный флагом required модуль pam_deny.so. Таким образом, выполнение неизвестной службы запрещается. Поэтому если вы не

хотите, чтобы кто-либо использовал какую-либо службу, например FTP, достаточно переименовать файл /etc/pam.d/ftp в /etc/pam.d/ftp.orig или присвоить ему другое имя.

Листинг 1.8. Файл /etc/pam.d/other

```
auth        required        pam_deny.so
auth        required        pam_warn.so
account     required        pam_deny.so
password    required        pam_deny.so
password    required        pam_warn.so
session     required        pam_deny.so
```

Модуль pam_dialup.so проверяет, нужно ли указывать пароль для доступа к удаленному терминалу или терминалам, для чего используется файл /etc/security/ ttys.dialup. Заметьте, что модуль применим не только к ttyS, а вообще к любому tty-терминалу. Когда пароль нужен, он сверяется с прописанным в файле /etc/ security/passwd.dialup. Изменения файла passwd.dialup осуществляются программой dpasswd (не следует пытаться изменять его вручную).

Модуль pam_group.so занимается проверками в соответствии с содержимым файла /etc/security/group.conf. В этом файле указываются группы, членом которых может стать указанный в файле пользователь при выполнении определенных условий. Набор условий указывается в этом файле напротив имени пользователя, этот набор определяет службу, терминал и время. Если пользователь работает с указанной службой с указанного терминала в указанное время, он может стать членом указанных групп. Это могут быть дополнительные группы, членом которых в момент входа в систему пользователь не является. Для OpenLinux в каталоге /etc/security/ содержится типовой файл group.conf, содержащий инструкции по его использованию. Параметров у этого модуля нет.

Модуль pam_lastlog.so заносит в файл lastlog сведения о том, когда и откуда пользователь вошел в систему. Обычно этот модуль помечается типом session и флагом optional.

Модуль pam_limits.so позволяет накладывать различные ограничения на пользователей, вошедших в систему. Эти ограничения не распространяются на пользователя root (или любого другого пользователя с нулевым идентификатором). Для желающих использовать этот модуль в OpenLinux имеется типовой файл /etc/security/limits.conf. Ограничения устанавливаются на уровне входа в систему и не являются глобальными или постоянными, действуя только в пределах одного входа. Модуль pam_lastfile.so принимает некоторую запись (item), сравнивает ее со списком в файле и на основании результатов сравнения возвращает УСПЕХ (SUCCESS) или НЕУДАЧА (FAILURE). Параметры этого модуля следующие:

- item=[терминал пользователь | удаленный_узел | удаленный_пользователь | группа| оболочка]

- sense=[allow|deny] (статус для возврата; когда запись найдена в списке, в противном случае возвращается статус, противоположный указанному)

file=/полный/путь/и/имя_файла - onerr=[succeed|fail] (какой статус возвращать в случае возникновения ошибки)

- app1y=[пользователь|@группа] (задает пользователя или группу, в отношении которой применяются ограничения. Имеет смысл только для записей вида item=[терминал | удаленный_узел | оболочка], для записей вида item=[пользователь | удаленный_пользователь | группа] игнорируется)

Модуль pam_nologin.so используется при авторизации типа auth с флагом required. Этот модуль проверяет, существует ли файл /etc/nologin, и если нет, то возвращает значение УСПЕХ (SUCCESS), иначе содержимое файла показывается пользователю и возвращается значение НЕУДАЧА (FAILURE). Этот модуль обычно используется в тех случаях, когда система еще не до конца введена в строй или временно закрыта для обслуживания, но не отсоединена от сети.

Модуль pam_permit.so является дополнительным к модулю pam_deny.so. Он всегда возвращает значение УСПЕХ (SUCCESS). Любые переданные параметры модулем игнорируются.

Модуль pam_pwdb.so предоставляет интерфейс к файлам passwd и shadow. Возможны следующие параметры:

- debug — запись отладочной информации в файл журнала;

- audit — дополнительная отладочная информация для тех, кому недостаточно обычной отладочной информации;

- use_first_pass — никогда не запрашивать пароль у пользователя, а брать его у предыдущих модулей стека;

- try_first_pass — попробовать получить пароль у предыдущих модулей, в случае неудачи запросить у пользователя;

- use_authtok — возвратить значение НЕУДАЧА (FAILURE) в случае, если pam_authtok не был

установлен, не запрашивать пароль у пользователя, а брать его у предыдущих модулей стека (только для стека модулей типа password);

- not_set_pass — не устанавливать пароль из этого модуля в качестве пароля для последующих модулей;
- shadow — поддерживать систему теневых паролей;
- unix — помещать пароли в файл /etc/passwd;
- md5 — при следующей смене паролей использовать пароли md5;
- bigcrypt — при следующей смене паролей использовать пароли DEC C2;
- nodelay — отключить односекундную задержку при неудачной авторизации.

На момент написания этих строк OpenLinux не поддерживает md5 или bygcrypt пароли, поэтому связанные с ними параметры использовать не следует. Кроме того, если в вашей системе вместо обычных используются теневые пароли, то не следует также использовать и параметр u nix.

Модуль pam_rhosts_auth.so разрешает/запрещает использование файлов .rhosts или hosts.equiv. Кроме того, он также разрешает/запрещает использование «опасных» записей в этих файлах. Параметры этого модуля следующие:

- no_hosts_equiv — игнорировать файл /etc/hosts.equiv;
- no_rhosts — игнорировать файл /etc/rhosts или ~/.rhosts;
- debug — протоколировать отладочную информацию;
- nowarn — не выводить предупреждения;
- suppress — не выводить никаких сообщений;
- promiscuous — разрешить использование подстановочного символа «+» в любом поле (очень плохая идея).

Модуль pam_rootok.so возвращает значение УСПЕХ (SUCCESS) для любого пользователя с нулевым идентификатором. Будучи помечен флагом sufficient, данный модуль позволяет получать доступ к службе без указания пароля. Будьте с ним осторожны. Если вставить его вызов с флагом sufficient в строку типа auth для службы login, то тогда суперпользователь сможет входить в систему без указания пароля, что, скорей всего, не входит в ваши планы. Параметр у модуля всего один: debug.

Модуль pam_securetty.so можно использовать только в отношении суперпользователей. Этот модуль работает с файлом /etc/securetty, позволяя суперпользователю входить в систему только через перечисленные в этом файле терминалы. Если требуется разрешить вход суперпользователя в систему посредством telnet (псевдотерминал ttyp), то следует либо добавить в этот файл строки для ttyp0-255, либо закомментировать вызов pam_securetty.so в файле для службы login. Помните, однако, что отключать этот модуль не рекомендуется.

Модуль pam_shells.so возвращает значение УСПЕХ (SUCCESS), если оболочка пользователя, указанная в файле /etc/passwd, присутствует в списке оболочек из файла /etc/shells. Если файл /etc/passwd не назначает пользователю никакой оболочки, то запускается /bin/sh. Если в файле /etc/passwd для пользователя указана оболочка, отсутствующая в списке /etc/shells, модуль возвращает значение НЕУДАЧА (FAILURE). Правом на запись в файл /etc/shells должен обладать только суперпользователь.

Модуль pam_stress.so используется для управления паролями. У него достаточно много параметров, в том числе и неизменный debug, но в общем случае из всех параметров интерес представляют только два:

- rootok — разрешить суперпользователю менять пароли пользователей без ввода старого пароля;
- expired — с этим параметром модуль выполняется, как если бы срок действия пароля пользователя уже истек.

Другие параметры модуля позволяют отключить любой из этих двух режимов, использовать пароль от другого модуля или передать пароль другому модулю и т. п. Здесь я не буду рассматривать все параметры модуля, поэтому если у вас возникнет потребность в использовании специальных возможностей этого модуля, прочтите их описание в документации модуля.

В OpenLinux модуль pam_tally.so в файлах из /etc/pam.d по умолчанию не используется, но рассказать о нем все-таки имеет смысл. Этот модуль производит подсчет попыток прохождения авторизации. При успешном прохождении авторизации счетчик числа попыток можно обнулять. Если количество неудачных попыток подключения превысило некоторое пороговое значение, доступ можно запретить. По умолчанию сведения о попытках помещаются в файл /var/log/ faillog (правом записи в этот файл должны обладать только ответственные лица). Глобальные параметры таковы:

- onerr=[succeed|fail] — что делать, если возникла ошибка, например не удалось открыть файл;
- file=/полный/путь/и/имя_файла — если отсутствует, то используется файл по умолчанию. Следующий параметр имеет смысл только для типа auth:
- no_magic_root — включает подсчет числа попыток для суперпользователя (по умолчанию не

ведется). Полезно, если разрешен вход суперпользователя в систему через telnet. Следующие параметры имеют смысл только для типа account:

- deny=n — отказать в доступе после n попыток. При использовании этого параметра поведение модуля reset/no_reset (см. далее) по умолчанию изменяется с no_reset на reset. Это происходит для всех пользователей, за исключением пользователя root (UID 0), если только не используется параметр no_magic_root;

- no_magic_root — не игнорировать параметр deny для попыток доступа, осуществляемых пользователем root. При использовании совместно с параметром deny= (см. ранее) для пользователя root по умолчанию устанавливается поведение reset, как и для всех остальных пользователей;

- even_deny_root_account — разрешает блокировку учетной записи суперпользователя при наличии параметра no_magic_root. При этом выдается предупреждение. Если параметр no_magic_root не используется, то независимо от числа неудачных попыток учетная запись суперпользователя, в отличие от записей обычных пользователей, никогда не будет заблокирована;

- reset — обнулять счетчик числа попыток при успешном входе;

- no_reset — не обнулять счетчик числа попыток при успешном входе; используется по умолчанию, если только не указан параметр deny=.

Модуль pam_time.so позволяет ограничить доступ к службе в зависимости от времени. Все инструкции по его настройке можно найти в файле /etc/security/ time.conf. Параметров у него нет: все задается в файле конфигурации.

Модуль pam_unix занимается вопросами обычной авторизации Unix (обычно вместо этого модуля используется pam_pwdb.so). Физически данный модуль состоит из четырех модулей, каждый из которых соответствует одному из типов PAM: pam_unix_auth.so, pam_unix_session.so, pam_unix_acct.so и pam_unix_passwd.so. Модули для типов account и auth параметров не имеют. У модуля для типа passwd параметр всего один: strict=false. При его наличии модуль не проверяет пароли на стойкость к взлому, позволяя использовать произвольные, в том числе и небезопасные (легко угадываемые или подбираемые) пароли. Модуль для типа session понимает два параметра: debug и trace. Отладочная информация параметра debug помещается в файл журнала отладочной информации, как указано в syslog.conf, а информация параметра trace из-за ее чувствительности — в журнал authpriv.

Модуль pam_warn.so заносит сообщение о своем вызове в syslog. Параметров не имеет.

Модуль pam_wheel.so разрешает становиться суперпользователем только членам группы wheel. Группа wheel — это специальная системная группа, члены которой имеют большие привилегии, чем обычные пользователи, но меньшие, чем суперпользователь. Ее наличие позволяет уменьшить число пользователей системы с привилегиями суперпользователя, сделав их членами группы wheel и тем самым повысив безопасность системы. Если суперпользователь может входить в систему только при помощи терминала, то данный модуль можно использовать для того, чтобы сделать недоступной для пользователей работу через telnet с привилегиями суперпользователя, отказав им в доступе, если они не принадлежат к группе wheel. Данный модуль весьма гибок и может использоваться для целей, отличных от заявленных выше. Модуль использует следующие параметры:

- debug — протоколирование отладочной информации;

- use_uid — определение принадлежности на основании текущего идентификатора пользователя, а не того, что был назначен ему при входе в систему;

- trust — в случае принадлежности пользователя к группе wheel возвращать значение УСПЕХ (SUCCESS), а не ИГНОРИРОВАТЬ (IGNORE);

- group=xxx — использовать для авторизации GID xxx, а не GID группы wheel;

- deny — меняет смысл процедуры на противоположный (возврат НЕУСПЕШНО). В комбинации с group= позволяет отказывать в доступе членам данной группы.

ПРИМЕЧАНИЕ -

Как вы, наверное, уже успели заметить, каталог /etc/security имеет непосредственное отношение к каталогу /etc/pam.d, поскольку содержит файлы конфигурации различных модулей PAM, вызываемых в файлах из /etc/pam.d.

Некоторые примеры

Теперь, когда вы имеете некоторое представление о возможностях модулей PAM, давайте вновь обратимся к листингу 1.6. В первой его строке, имеющей тип auth, стоит вызов модуля securetty с флагом required. Стало быть, войти в систему как суперпользователь можно лишь с терминалов, перечисленных в

файле /etc/securetty. Второй строкой проверяется пароль пользователя. Результат выполнения третьей строки определяется наличием файла /etc/nologin. Если такой файл не существует, возвращается значение УСПЕХ (SUCCESS). Четвертая строка закомментирована и потому игнорируется, но если удалить символ комментария, то будет включен механизм проверки удаленного доступа с использованием авторизации и паролей, управляемый файлами tty.dialup и passwd.dialup из каталога /etc/security. Последняя строка типа auth помечена флагом optional, однако она может понадобиться, если кто-то использует систему только для электронной почты.

Строка типа account проверяет, не устарел ли пароль пользователя.

Две строки типа session весьма стандартны. В первой строке модуль pwdb проверяет, может ли пользователь воспользоваться службой. Модуль второй строки, заносящий запись в файл lastlog, помечен флагом optional, поскольку пользователь должен получить возможность войти в систему даже в том случае, если файл lastlog не доступен на запись. Таким образом, в результате выполнения модуля lastlog вне зависимости от того, какое из трех значений — УСПЕХ (SUCCESS), НЕУДАЧА (FAILURE) или ИГНОРИРОВАТЬ (IGNORE) - получено, пользователь все равно сможет войти в систему.

Строка типа password помечена как required и предназначена для обновления маркеров авторизации в случае, если необходима смена пароля.

Некоторые файлы из /etc/pam.d весьма просты. В файле chfn, например, используется всего один модуль — pam_pwdb.so, помеченный как required для типов auth, account и passwd. Точно так же выглядит и файл для chsh. Файлы для служб электронной почты imap и pop чуть сложнее: в них модуль pwdb используется и для типа session, и еще вызывается модуль nologin для типа auth. Возможно, это не самая лучшая идея, но запись nullok разрешает использование «пустых» паролей.

Файлы остальных служб уже должны быть в принципе понятны, но пару-тройку из них, тем не менее, стоит прокомментировать. Во-первых, взглянем на файл службы ftp. В первой его строке используется модуль listfile. Смысл этой записи в том, чтобы посмотреть, есть ли имя данного пользователя в файле /etc/ftpusers, и если есть, отказать ему в доступе к ftp (в данном случае гораздо легче сказать, кто не может пользоваться ftp, чем перечислять всех, кто может). Если файл не существует или запись в него разрешается для всех пользователей, считается, что пользователь найден не был.

Теперь перейдем к файлу для rlogin. Первая строка проверяет, является ли безопасным терминал, с которого суперпользователь входит в систему (мы не хотим передавать пароль суперпользователя открытым текстом по сети, которой мы не доверяем). Поэтому проверьте, что у вас прописано в файле /etc/securetty. Согласно следующей строке наличия файлов /etc/hosts.equiv или ~/rhosts достаточно для предоставления доступа. Если же эти файлы не существуют, то доступ будет предоставлен лишь в случае успешного выполнения всех оставшихся модулей. Заметьте, строка, содержащая вызов модуля cracklib.so, закомментирована, поэтому, возможно, имеет смысл активизировать ее, чтобы пользователи не могли использовать небезопасные пароли.

Последний файл, на который стоит обратить внимание, это файл для службы su. Вот его первая строка:

```
auth      sufficient      pam_rootok.so
```

Эта строка позволяет суперпользователю становиться любым пользователем при помощи команды su без ввода пароля — свойство, которое вы уже, наверное, заметили, работая с системой.

Как видите, использование модулей и наличие возможности накопления их позволяет задавать авторизацию практически любой сложности. Если вы решите отредактировать какие-либо файлы из каталога /etc/pam.d в соответствии с собственными предпочтениями, то мне остается лишь посоветовать вам не лениться и всегда тестировать результаты изменений на нескольких сценариях, позволяющих проверить, что все работает точно так, как и было задумано, и никому лишнему доступ не предоставляется. Тестируйте как случай, когда авторизация должна проходить успешно, так и случай, когда подключающийся пользователь должен получить отказ. И не забудьте проверить суперпользователя, как истинного, так и лишь выдающего себя за такового.

Записи PAM в файлах журналов

Результаты авторизации служб с ограничением доступа протоколируются демоном syslogd, который помещает все сообщения от PAM в файл /var/log/secure (листинг 1.9).

Листинг 1.9. Содержимое /var/log/secure

```
Jan 11 16:45:14 chiriqui PAM_pwdb[30022]:   (su) session opened for user root
by david(uid=0)
Jan 11 16:45:25 chiriqui PAM_pwdb[30022]: (su) session closed for user root
Jan 11 17:18:06 chiriqui login[13217]: FAILED LOGIN 1 FROM (null) FOR david,
Authentication failure
```

```
Jan 11 17:18:13 chiriqui login[13217]: FAILED LOGIN 2 FROM (null) FOR david.
Authentication failure
Jan 11 17:18:17 chiriqui PAM_pwdb[13217]: (login) session opened for user david
by (uid=0)
Jan 11 17:18:06 chiriqui login[13217]: FAILED LOGIN 1 FROM (null) FOR david.
Authentication failure
Jan 11 17:18:13 chiriqui login[13217]: FAILED LOGIN 2 FROM (null) FOR david,
Authentication failure
Jan 11 17:18:17 chiriqui PAM_pwdb[13217]: (login) session opened for user david
by (uid=0)
Jan 11 17:18:17 chiriqui -- david[13217]: LOGIN ON ttyl BY david
Jan 11 17:18:20 chiriqui PAM_pwdb[13217]: (login) session closed for user david
```

Каждая запись начинается с даты, времени и имени узла. После чего следует имя модуля PAM и идентификатор процесса, заключенный в квадратные скобки. Затем, в круглых скобках, идет имя службы с ограничением доступа. Для листинга 1.9 это либо su, либо login. После имени службы следует либо «session opened» (сеанс открыт), либо «session closed» (сеанс закрыт).

Запись, следующая непосредственно за записью с «session opened», является сообщением о входе в систему, из которого можно узнать, кто и откуда вошел в систему. Если после нескольких попыток вы так и смогли войти в систему, имеет смысл обратиться к файлу /var/log/secure и посмотреть, отчего и на каком этапе вход в систему заканчивается неудачей.

Заключение

В этой главе рассказывалось о пользователях и процедуре входа в систему. Из данной главы вы узнали о том, каким образом система обрабатывает файл /etc/passwd, как этот файл использовался в прошлом и как он используется сейчас. Также вы узнали о назначении файла /etc/shadow, который является более защищенным аналогом файла /etc/passwd. Кроме того, речь шла и о файле /etc/group.

Затем мы рассмотрели файл /etc/login.defs и его связь со значениями по умолчанию, используемыми программой useradd при добавлении нового пользователя в систему.

Завершилась же глава рассказом о системе безопасности PAM. Мы рассмотрели принципы ее работы, используемые ею файлы, такие как /etc/securetty, /etc/ shells, и файлы из /etc/security/, а также какие модули PAM имеются в системе и как их можно использовать для управления доступом к службам.

В следующей главе тема групп будет рассмотрена более подробно, с позиций взаимодействия пользователей, групп и файлов. Из данной главы вы узнали, что в операционной системе Linux существуют группы, а вот для чего они нужны, объясняется в следующей главе.

2 Безопасность уровня пользователей и групп

В данной главе рассматриваются следующие вопросы:

- что такое пользовательская группа по умолчанию и частные пользовательские группы;
- изменение пользователя/группы;
- как изменение пользователя/группы влияет на графический интерфейс;
- безопасность и пользователи;
- безопасность и пароли;
- защита паролей;
- выбор хорошего пароля;
- взлом паролей.

Управлять пользователями можно лишь одним единственным способом: каждому из них назначается уникальное имя. А вот механизм групп позволяет вам выбрать одну из двух различных схем. К выбору схемы следует подходить обдуманно, поскольку желательно единожды определиться с используемой схемой и более не менять ее (особенно если вы администрируете систему с большим количеством пользователей). Схемы эти таковы: группа по умолчанию и частные группы пользователей. Как можно предположить, у каждой из них имеются свои достоинства и недостатки.

Группа по умолчанию

Исторически, во всех разновидностях Unix, равно как и в других операционных системах, напоминающих Unix (включая Linux), использовалась схема, в рамках которой любые вновь создаваемые новые пользователи системы по умолчанию становились членами одной группы. При создании учетной *записи* администратор мог специально изменить членство в группе, однако так редко кто поступал. Причина была *в* том, что в старых системах пользователь не мог принадлежать к нескольким группам одновременно. Любой пользователь мог быть членом только одной группы. Поэтому все созданные пользователи попадали в группу по умолчанию, достаточно безопасную для того, чтобы ее членом мог быть кто угодно. Пользователям, которым надо было стать членом другой группы (например, если несколько пользователей работают над одним проектом и должны обладать доступом к одним и тем же данным), приходилось прибегать к команде newgrp. Эта команда позволяет пользователю покинуть свою старую группу и присоединиться к новой (разумеется при условии, что у этого пользователя есть право на присоединение к новой группе).

В настоящее время ограничения на одновременную принадлежность пользователя лишь к одной группе более не существует. Любой пользователь может принадлежать одновременно к нескольким группам. Вы можете сделать пользователя членом любого удобного для вас количества групп. Очевидно, теперь операционные системы устроены более гибко. В OpenLinux, независимо от того, используете ли вы группу по умолчанию или частные группы пользователей (о которых рассказывается далее), пользователи могут быть членами произвольного числа групп. По команде newgrp пользователь становится членом указанной в команде группы, при этом данная группа становится для данного пользователя *группой входа в систему* (login group). Имейте в виду, что при этом пользователь продолжает оставаться членом тех групп, в которые он входил до выполнения команды newgrp. Группа входа в систему является группой, которая (если атрибуты каталога не предписывают ничего другого) становится групповым владельцем файлов, создаваемых пользователем.

ССЫЛКА
Про атрибуты каталога подробнее рассказывается в *главе 4,*

Различие между группой по умолчанию и частными группами пользователей заключается в степени открытости этих двух схем. В случае схемы с группой по умолчанию любой пользователь может читать (а

часто и изменять) файлы другого пользователя. С частными же группами чтение или запись файла, созданного другим пользователем, возможны лишь если его владелец явно предоставил права на эти операции остальным пользователям.

Если требуется, чтобы пользователи могли присоединяться и покидать группу без вмешательства системного администратора, то этой группе можно назначить пароль. Как говорилось ранее, пользователь может пользоваться привилегиями определенной группы только в случае, если он принадлежит к ней. Здесь есть два варианта: либо он принадлежит к группе с момента входа в систему, либо он становится членом группы впоследствии, уже после того, как он начал работу с системой. Чтобы пользователь мог присоединиться к группе, к которой он не принадлежит, этой группе должен быть назначен пароль (желательно теневой).

По умолчанию в OpenLinux групповые пароли не используются, поэтому файла gshadow в каталоге /etc нет. О том, как активизировать этот механизм, рассказывается в главе 1.

В случае с OpenLinux тип используемой схемы зависит от программы, которую вы используете для создания новых пользователей. Программы useradd и LISA используют схему с группой по умолчанию. Средство администрирования COAS работает с частными группами пользователей. Безусловно, поведение по умолчанию любой из этих программ можно изменить. Более того, если вы не сделаете этого и в дальнейшем для создания новых пользователей будете использовать все три программы, в вашей системе будет использоваться помесь из двух схем, а это плохая практика. Чтобы избежать этого, следует либо изменить файл /etc/ login.defs, либо перенастроить COAS. Если вы не хотите чего-либо перенастраивать, то для создания новых пользователей можно пользоваться исключительно одной из трех программ. Когда требуется изменить поведение этой программы по умолчанию, это можно сделать из командной строки, кроме того, вы можете сначала создать нового пользователя, а затем привести его параметры в соответствие с той или иной схемой. Однако чем меньше времени уходит у вас на тривиальные административные задачи, тем больше его остается на вопросы безопасности.

СОВЕТ -

Если для выполнения рутинных задач администрирования пользователей вы постоянно используете только одну из программ — useradd, LISA или COAS, — файлы настроек пользователей получаются более согласованными и более легкими в сопровождении.

Преимущество схемы с группой по умолчанию заключается в том, что она облегчает совместное использование файлов, так как при ее использовании не нужно заботиться о правах доступа к ним. Эта схема подразумевает открытый подход к системе по принципу «разрешено все, что не запрещено».

ССЫЛКА

Главы 3 и 4 содержат более подробную информацию о владельцах файлов и их связи с правами на чтение и изменение файлов.

Для пользователей понятней, а значит, в чем-то удобней, схема с группой по умолчанию, однако для больших систем, где нельзя одинаково доверять всем пользователям, предпочтительней использовать частные группы пользователей. Открытый подход упрощает не только совместное использование файлов, но и их порчу или удаление. Поэтому если пользователей много, имеет смысл оградить их от неосторожных действий друг друга. В OpenLinux группа по умолчанию называется users.

СОВЕТ

Настройка параметров пользователей по умолчанию — это высокоприоритетная задача, которую следует выполнить сразу же после того, как вы установите систему.

Частные группы пользователей

Частные группы пользователей обладают именами, совпадающими с именами пользователей. Частная группа делается группой входа в систему, поэтому по умолчанию, то есть если атрибуты каталога не предписывают ничего другого, она назначается в качестве группы-владельца всех файлов данного пользователя. Таким образом, пользователю david ставится в соответствие группа david, которая назначается группой-владельцем всех файлов этого пользователя.

Преимущество частных групп пользователя состоит в том, что пользователям не нужно думать об ограничении доступа к своим файлам: по умолчанию доступ к пользовательским файлам с самого момента их создания будет ограничен. В OpenLinux, при использовании частных групп пользователь может читать

или изменять только принадлежащие ему файлы. Кроме того, создавать файлы он может только в своем домашнем каталоге. Данное поведение по умолчанию может быть изменено системным администратором или пользователем, причем как на уровне отдельного файла, так и на уровне каталога.

Заметим здесь, что по умолчанию переменная umask, которая определяет режим доступа к создаваемому файлу, настроена таким образом, что чтение-запись файла разрешается как владельцу этого файла, так и группе, которой принадлежит файл.

ССЫЛКА

Подробное обсуждение umask можно найти в главе 3.

Если вы используете схему частных групп пользователей, это никак не влияет на поведение базового механизма членства в группах. Если пользователь является членом одновременно нескольких групп, значит, ему доступны файлы, принадлежащие к любой из этих групп. Как это работает, станет ясно из следующих двух глав.

СОВЕТ

Если в вашем ведении находится несколько систем, имеет смысл сделать одну из них основной и скопировать с нее /etc/passwd, /etc/shadow, /etc/groups, /etc/gshadow на остальные системы.

Изменение пользователя/группы

Есть несколько команд, при помощи которых пользователь может управлять своим именем и/или группой, к которой он принадлежит, или именем или группой, от лица которых выполняется программа. Одна из таких программ уже упоминалась ранее. Это newgrp.

Как уже говорилось, команда newgrp может быть выполнена любым пользователем. Она позволяет ему присоединиться к группе, к которой он доселе не принадлежал, но только если этой группе назначен пароль. Данная команда не позволит вам присоединиться к группе без пароля, если вы не являетесь членом этой группы.

Однако команду newgrp можно использовать в отношении группы, членом которой вы уже являетесь. В этом случае newgrp делает указанную группу группой входа в систему. Группы пользователя подразделяются на два типа: группа входа в систему и все остальные группы, к которым принадлежит этот пользователь. Пользователь может принадлежать к нескольким группам, однако группой-владельцем файлов, создаваемых пользователем, всегда будет назначаться группа входа в систему этого пользователя.

ССЫЛКА

В главе 4 рассказывается о том, как можно изменить такое поведение для какого-либо каталога.

Помимо newgrp для управления принадлежностью файла тому или иному пользователю или группе можно использовать также команды chown и chgrp. Предпочтительней, тем не менее, использовать именно newgrp, так как при использовании команд chown и chgrp легче допустить ошибку. Предположим, например, что пользователь silvia намеревается создать несколько файлов типа GIF. Группой входа в систему для нее является группа silvia, однако созданные ею файлы должны быть доступны не только ей, но и другим членам группы gifs (но больше никому). Если пользователь silvia окажется забывчивой и не выполнит команду newgrp, ей придется явно менять владельца созданных ею файлов командами chown или chgrp. Выполнение их для каждого файла по отдельности чревато пропуском одного или нескольких файлов. Проблему можно решить, использовав шаблон имени файла (*.gif), однако помимо созданных только что в каталоге могут находиться и другие файлы GIF, которые silvia ранее преднамеренно сделала недоступными для остальных пользователей. Для запуска программы, создающей файлы GIF, silvia может воспользоваться командой sg. Команда sg на самом деле представляет собой символическую ссылку, указывающую на newgrp. Будучи вызвана как sg, она делает пользователя членом новой для него группы не перманентно, а лишь на время выполнения указанной программы. Свое название команда получила от словосочетания *«substitute group»* (подстановка группы). С эквивалентом этой команды, применяемым в отношении имени пользователя, мы познакомимся в следующем разделе.

Если вашей группой входа в систему является группа readers, а помимо этой группы вы также принадлежите к группе users, то по умолчанию выполнение программ и сохранение файлов будет происходить в контексте группы readers. Чтобы файлам, создаваемым программой xv, в качестве владельца назначалась группа users, можно запустить с использованием команды sg. Например, чтобы файлам, созданным в программе xv, в качестве владельца назначалась группа users, запустите эту программу

следующим образом:

```
sg - users -c xv
```

При наличии у запускаемой программы параметров команда запуска этой программы заключается в кавычки:

```
sg - users -c "xv my.gif"
```

Заметьте также, что при работе в среде X Window нужно разрешить этой группе использование дисплея с помощью команды xhost.

Область действия команды newgrp в среде X Window ограничивается программой xterm, в которой она была выполнена: в контексте новой группы будут выполняться лишь программы, запущенные через этот терминал, а значит, пользователь не может изменять с ее помощью группу входа в систему для программ, запущенных посредством диспетчера окон. Программу, которую всегда нужно выполнять в контексте вторичной группы, можно запускать через сценарий, устанавливающий для нее требуемую группу входа в систему.

Далее про X

Система X Window всегда привносит в жизнь пользователей дополнительные трудности. В данном случае трудности эти не связаны напрямую с X, а следуют из логики работы /etc/groups и /etc/gshadow. Тем, кто не использует теневые пароли для групп, беспокоиться особенно не о чем. В случае с X установить группу, защищенную паролем, из простого сценария не получится, однако для вторичных групп пользователя, не требующих ввода пароля, смена группы осуществляется предельно просто. Достаточно следующего сценария:

```
#!/bin/bash
sg - gifs -c /usr/X11R6/bin/xv &
```

В результате запуска этого сценария будет запущена программа xv, первичной группой которой будет сделана группа gifs. Что и требовалось получить.

Труднее тем, кто использует теневые групповые пароли, поскольку в этом случае при выполнении данного сценария на экране появится сообщение об ошибке. Когда в файле /etc/groups перечислены входящие в группу пользователи, любой из них автоматически считается ее членом непосредственно после входа в систему. Однако в случае теневых паролей список пользователей группы перемещен в файл /etc/gshadow, так что вошедший в систему пользователь не зачисляется автоматом в ее члены, но может присоединяться к ней посредством команды newgrp или же выполнять от ее имени какую-либо программу при помощи команды sg. Проблема в том, что с точки зрения X данный пользователь (который не обязательно является пользователем, инициировавшим рабочий сеанс X) не имеет права на установку соединения. Поэтому для незащищенных паролем групп приведенный ранее сценарий изменяется следующим образом:

```
#!/bin/bash
xhosts +localhost
sg - gifs -c /usr/X11R6/bin/xv &
```

Добавленная строка позволяет получать доступ к экрану новой группе (gifs). Для большинства рабочих станций это не должно привести к каким-либо существенным проблемам с безопасностью, поскольку данная строка всего лишь разрешает доступ к экрану пользователям локального узла (для получения дополнительной информации об X и xhost обратитесь к хорошему руководству системного администратора Linux).

ПРИМЕЧАНИЕ

Использование X-сервера (в особенности совместо с xdm или kdm) влечет за собой целый ряд своих тонкостей, еще более усугубляемых графическими приложениями, поскольку их можно запускать не только через командную строку, но и при помощи значка на графическом рабочем столе.

Изменение пользователя

Одна из прописных истин системного администрирования гласит: никогда не входите в систему как суперпользователь без абсолютной на то необходимости. Почему? Потому что большую часть задач можно решить на уровне привилегий обычного пользователя. Если же вы намерены выполнить некоторое действие, которое можно выполнить только на уровне привилегий root, вы можете превратиться в пользователя root лишь на время. Выполнив ту или иную привилегированную процедуру, вы можете снова

стать обычным пользователем.

ПРИМЕЧАНИЕ

Обычный пользователь не в силах нанести системе столько вреда, сколько может сделать неосторожный суперпользователь. Последствия вашей опечатки как суперпользователя могут быть весьма фатальными, вплоть до того, что всем вашим системным файлам (а то и вообще всем файлам, хранящимся в системе) можно будет сказать «прощай». В некоторых компаниях после этого могут сказать «прощай» и вам.

Превращением одного пользователя в другого занимается команда su. Свое название команда получила от *«substitute user»* (подстановка пользователя), но так как чаще всего она используется, чтобы стать суперпользователем, некоторые считают, будто ее название расшифровывается как *«super user»* (суперпользователь). Для эффективного использования этой команды вам следует кое-что знать об особенностях ее работы.

Команда su, вызванная без аргументов, запросит у вас пароль, после чего (получив в ответ правильный пароль, разумеется) сделает вас пользователем root. Данная команда является службой с ограничением доступа, поэтому все аспекты ее безопасности можно настроить через файл /etc/pam.d/su. При желании вы можете сделать так, что некоторые избранные пользователи смогут стать пользователем root, не зная при этом пароля root, однако это очень рискованная идея.

ССЫЛКА

Про внутреннее устройство файла /etc/pam.d/su рассказывается в галве 1.

Вызов команды su без аргументов равносилен ее вызову с аргументом root Вообще говоря, чтобы стать каким-либо пользователем с помощью команды su, следует указать его имя в качестве аргумента (причем если вы суперпользователь, чтобы стать другим пользователем, вы не обязаны будете указывать пароль этого пользователя). Это чрезвычайно удобный способ решения проблем, связанных с подключением пользователей. В этом случае, скорее всего, следует поставить дефис (-) между su и именем пользователя. Наличие дефиса говорит команде, что смена пользователя должна осуществляться по полной программе, как если бы вы входили в систему под именем этого пользователя, то есть со сменой пользовательского окружения. Без указания дефиса в результате выполнения su вы станете другим пользователем, но окружение останется вашим изначальным. В частности, значение переменной PATH останется прежним и не будет заменено на значение PATH пользователя, которым вы стали. Думается, такое поведение не слишком способствует разрешению проблем с учетными записями пользователей.

ПРИМЕЧАНИЕ -

Обращение su без указания имени пользователя (с дефисом или без) трактуется как указание сделать вас пользователем root.

Но несмотря на все удобство su, пароль суперпользователя должны знать не более шести человек, иначе контроль за ним можно считать утерянным. Но что делать, если вы хотите наделить пользователя полномочиями root и при этом не намерены раскрывать ему пароль суперпользователя? Есть несколько способов. Наиболее правильным считается использование команды sudo. Данная команда позволяет избранным пользователям выполнять некоторые программы на правах суперпользователя, причем у обратившегося к этой команде пользователя запрашивается не пароль суперпользователя, а его собственный пароль. Используется sudo подобно команде sg. Пользователь вводит sudo команда_для_выполнения, затем свой пароль, и если ему это разрешено, указанная команда выполняется в контексте привилегий суперпользователя.

Упомянутое ранее количество пользователей, знающих пароль root, может вызвать недоумение. Почему именно шесть? Однако попробуйте вспомнить всех тех, кому вы раскрыли пароль суперпользователя за последние шесть месяцев. Скорей всего, на ум придет около шести человек. Кроме того, пароль суперпользователя куда надежней защищен, когда эти пользователи знают, что их всего шесть, чем когда каждому из них известно, что кроме него пароль знают еще 15 человек. И кстати, сколь часто вы готовы менять пароль суперпользователя лишь потому, что кто-то из них вышел из вашего поля зрения? Так что чем меньше, тем лучше.

Конфигурационный файл находится в каталоге /etc и в силу своей очевидности не требует каких-либо пояснений. Способ с sudo хотя и лучший, но не единственный. В качестве альтернативы можно задать выполнение программы в контексте суперпользователя через изменение атрибутов файла. Более подробно об этом рассказывается в главе 4.

Безопасность и пользователи

Не следует питать каких-либо иллюзий относительно взгляда пользователей на вопросы безопасности: им до них нет никакого дела. О безопасности пользователь вспоминает лишь тогда, когда у него возникают проблемы с нею, а до тех пор он пребывает в уверенности, что его файлы надежно защищены и ничего *с* ними случиться не может. Поэтому администратор, исходящий из предпосылки, что злейший враг файлов пользователей — это сами пользователи, не так уж и далек от истины.

Пользователи обычно интересуются только тем, как войти в систему и запустить нужные им программы. Интерес к безопасности появляется у них лишь после утери важных файлов. Но длится он недолго. Узнав, что меры приняты, пользователи быстро забывают о всякой предосторожности.

Вообще говоря, не их это забота — безопасность. Системный администратор должен продумать, реализовать и поддерживать политику безопасности, которая позволила бы пользователям делать свою работу, не отвлекаясь на посторонние для них вопросы защиты.

Основная опасность для системы, как правило, исходит изнутри, а не снаружи. Ее источником (особенно в крупных системах) может стать, например, рассерженный пользователь. Следует, однако, избегать излишней подозрительности, когда вред, нанесенный по незнанию, принимается за злой умысел. О том, как оградить пользователей от непреднамеренного повреждения своих и чужих файлов, рассказывается в первой части книги. Как показывает практика, среднестатистический пользователь не в состоянии повредить систему. Беспокоится нужно лишь о тех пользователях, которые в состоянии найти лазейку в механизмах защиты и действительно способны причинить целенаправленный вред системе. Но таких пользователей обычно мало и со временем они становятся известны, особенно если знать, на что обращать внимание. К группе риска относятся пользователи, которые в силу своего положения или благодаря своим связям могут получить доступ на уровне привилегий root. По мере того как вы будете овладевать материалом данной книги, вы будете узнавать, что именно следует расценивать как признаки надвигающейся беды.

По умолчанию пользователи получают полный контроль над своими домашними каталогами. Если вы используете группу по умолчанию, все пользователи системы принадлежат к одной группе. Любой пользователь обладает правом доступа к домашним каталогам других пользователей и расположенным в них файлам. При использовании схемы с частными группами пользователей любой из пользователей системы обладает доступом только к своему собственному домашнему каталогу, а домашние каталоги других пользователей ему не доступны.

Если для всех пользователей системы требуется обеспечить общий доступ к некоторому набору общих файлов, рекомендуется создать где-нибудь общий каталог специально для этих целей, завести группу, членами которой были бы все пользователи (это может быть группа users или любая другая созданная вами группа), и предоставить этой группе соответствующие права доступа к данному общему каталогу. Если пользователь желает сделать некоторые из своих файлов доступными для других пользователей, он может просто скопировать их в этот каталог и обеспечить, чтобы эти файлы принадлежали к той самой группе, членами которой являются все пользователи.

ССЫЛКА

0 том, как группа пользователей может использовать общий каталог для работы над общим проектом, равно как и о других подобных конфигурациях, рассказывается в главе 4.

Некоторые пользователи нуждаются в использовании или просто не могут обойтись без программ, не входящих в комплект OpenLinux. Большинство пользователей со временем обзаводится множеством собственных файлов: документы, конфигурационные файлы, сценарии и т. п. Система OpenLinux не предоставляет пользователям особой помощи в деле организации своих файлов, оставляя эту задачу системному администратору. О системном администрировании Linux написано множество замечательных книг, поэтому здесь я не буду вдаваться в подробности, а ограничусь лишь общими соображениями.

Структура каталогов, создаваемых в домашнем каталоге каждого нового пользователя, определяется содержимым каталога /etc/skel. В типичном /etc/skel обычно присутствуют следующие каталоги:

/bin

/src

/docs

/misc

Эти каталоги используются для хранения (соответственно) бинарных файлов, исходных файлов, файлов документов и других разнообразных файлов. Многие программы по умолчанию предлагают сохранять файлы тех или иных типов в одном из этих подкаталогов. Получив разъяснение о назначении имеющихся в их распоряжении каталогов, пользователи обычно охотно начинают пользоваться ими,

поскольку это избавляет их от необходимости придумывать что-то свое. Не забудьте только сделать каталог ~/bin одним из последних каталогов, перечисляемых в переменной PATH пользователей. Более подробно об этом рассказывается в книгах, посвященных системному администрированию Linux.

Безопасность и пароли

Говорят, что где тонко, там и рвется, — это высказывание часто вспоминают, когда речь заходит о значимости паролей в системе безопасности. Вообще говоря, надежность системы безопасности определяется множеством факторов, в частности тем, какие службы Linux-система делает доступными для внешних пользователей (используется ли она в качестве web-сервера, можно ли войти в нее при помощи telnet и т. д.). Другим определяющим фактором являются пароли пользователей, что подводит нас к еще одному фактору — соблюдение пользователями политик безопасности. Но как говорилось ранее, простой пользователь знать ничего не желает о безопасности. Если мы уважаем пользователя и не хотим менять его отношение к безопасности принудительными методами, нам следует сделать систему безопасности удобной и понятной для него. Труднее всего обеспечить удобство. Все безопасное обычно не слишком удобно (поскольку за удобством стоят не сочетающиеся с безопасностью предсказуемость и элементарность) и потому входит в конфликт с обычным поведением людей, предпочитающих всем возможным способам самый удобный. В конце концов, пользователи работают с системой для того, чтобы выполнить возложенную на них работу, а не добавить себе новой. Дабы пользователи сознательно не шли по пути наименьшего сопротивления при работе с паролями, я обычно стараюсь объяснить им, для чего вообще нужны пароли и почему так важно поддерживать их безопасность. Важно не с общих позиций типа «систему с низкой безопасностью могут взломать и украсть или повредить важные файлы», а с позиций личных интересов пользователя.

СОВЕТ

Наилучший способ заручиться поддержкой пользователей в деле поддержания безопасности паролей на должном уровне — объяснить им, почему соблюдение норм безопасности при работе с паролями находится в их же собственных интересах.

Большинство пользователей понимают всю важность электронной почты для своей работы. Тем не менее они не осознают, что любой вошедший в систему под их именем получает возможность использовать их электронную почту от их имени против них. Спросите у пользователя, использует ли он электронную почту в личных целях. Скорее всего, он ответит, что да. Затем спросите его, приходилось ли ему решать по электронной почте важные деловые вопросы. Таких, которые ответят «нет», с каждым днем становится все меньше и меньше. Но даже в случае отрицательного ответа некоторые из деловых партнеров вполне могут считать сделку по электронной почте столь же обязывающей, как сделка по телефону.

После чего объясните пользователю, что его электронные письма подчас обладают такой же важностью, как и его личная подпись. И хотя заголовок электронного послания можно подменить, в большинстве случаев подобная подмена так же противозаконна, как и подделка подписи. Но если некто, тем или иным способом узнав пароль другого пользователя, войдет в систему под его именем, то тем самым он, образно говоря, получит возможность подписываться подписью другого человека. Любая почта, отправленная им, будет технически неотличима от почты, отправленной самим пользователем. Практика предоставления кому-либо возможности входа в систему под другим именем является нежелательной и ее следует избегать (исключением являются системные администраторы, которые используют эту возможность для тестирования сценариев входа в систему и параметров пользователя, но для этого им нет необходимости знать пароль этого пользователя). К нежелательным явлениям следует отнести и вход в систему под чужим именем (даже с разрешения другого пользователя). Насколько это нежелательно? Ответ на этот вопрос определяется строгостью политики безопасности предприятия.

Однако пользователи должны понимать, что есть и другие не менее опасные способы получить несанкционированный доступ к их учетной записи. Наиболее распространен случай, когда пользователь, опасаясь забыть пароль, делает его простым для запоминания, а значит, и угадывания, или же записывает пароль на бумажку, которая зачастую просто прикрепляется к монитору. Система парольной безопасности основывается на двух вещах: постоянное имя пользователя и периодически меняющийся пароль. Большинство людей никому не скажут PIN-код для доступа к своему банковскому счету, однако свой пароль пользователя они оберегают далеко не столь ревностно. Хотя в отличии от ситуации с банковским счетом, где постоянная часть, то есть кредитная карта, является физическим объектом, доступ к которому еще нужно получить, постоянная часть системы парольной безопасности, то есть имя пользователя, известна всем (по крайней мере, всем в пределах компании и тем, с кем данный пользователь вел

переписку по электронной почте). Поэтому если переменная часть где-то записана или легко угадывается или подбирается программой, перебирающей слова из словаря, то такую учетную запись нельзя считать хорошо защищенной.

Наконец, пользователи должны знать о существовании такого метода получения пароля, как «социальная инженерия» (social engineering). Большинство из нас встречалось в своей жизни хотя бы с одним человеком, о котором можно сказать «скользкий как уж». Такие люди обладают способностью убеждать других людей, прибегая к логично построенной аргументации, предоставить нужную им информацию. Но это не единственно возможный способ узнать чужой пароль. Иногда достаточно просто подсмотреть. Например, вы вводите имя пользователя и пароль не глядя на экран и, оторвав свой взгляд от клавиатуры, обнаруживаете, что где-то ошиблись (случайно нажали клавишу Enter перед тем, как ввести имя пользователя, или произошло что-нибудь вроде того), в результате вместо имени пользователя вы ввели свой пароль, и теперь ваш пароль виден любому, кому виден экран вашего монитора. Вполне вероятно, что ваш пароль только что был «украден» кем-то, кто просто проходил мимо.

Средством противодействия подобным казусам является регулярная смена пароля. Как часто это должно происходить, решать вам. Можно, конечно, менять пароль раз в десять лет, но лучше не делать промежутки между сменами слишком длинными, равно как лучше не делать их и слишком короткими, например, раз в час. Не менять пароль слишком долго означает подвергать себя риску взлома. Если вы меняете его слишком часто, значит, вы рискуете запутаться в паролях, забыть пароль или поддаться искушению записать его на бумагу.

ПРИМЕЧАНИЕ-

Проникновение постороннего в систему под видом обычного пользователя может иметь печальные последствия не только для файлов этого пользователя, но и для всей системы в целом, поскольку чем больше этот посторонний будет знать о вашей системе, тем легче ему будет найти прорехи в ее защите.

Даже если не впадать в крайности, все равно период смены пароля может изменяться в достаточно широком диапазоне. Произвести за вас оценку риска эта книга не сможет, здесь я приведу несколько вопросов, ответы на которые помогут вам определить наиболее подходящую для вас частоту смены пароля (в RFC 2196 объясняются основы оценки риска и даются ссылки на другие источники, содержащие информацию по этой важной теме).

- Свободно ли ваше рабочее пространство от любопытных глаз? Иначе говоря, если ваш пароль появится на несколько секунд на экране монитора, какова вероятность того, что кто-то его заметит?

- Насколько безопасна ваша сеть? Сеть, состоящая всего из двух машин, одной из которых пользуетесь только вы, а другой — члены вашей семьи, без сомнений относится к разряду безопасных (если, конечно, ваши дети не являются малолетними хакерами).

- Предусматривается ли подключение к вашей локальной системе других пользователей в то время, как вы уже подключились и работаете с ней?

- Используете ли вы тот же самый пароль для удаленного доступа к другим компьютерам вашей сети с использованием не защищенных (то есть не шифруемых) протоколов? Если да, существует ли возможность «прослушивания» сетевых каналов с других сетевых узлов (карты Ethernet в режиме прослушивания сети)?

Выбрав для себя подходящий интервал смены пароля, будь это неделя, месяц, три месяца, шесть месяцев или год, вы можете легко обеспечить его соблюдение при помощи механизма устаревания пароля, о чем рассказывалось в предыдущей главе. Но кроме того, сам пароль должен быть одновременно устойчив ко взлому, легко запоминаем (чтобы не было нужды записывать его) и трудно угадываем для тех, кто может вас знать.

Так какой же пароль является хорошим? Чем больше пароль напоминает набор случайных символов, тем лучше. Длина хорошего пароля находится в пределах от шести до восьми символов, хотя при некоторых обстоятельствах пяти символов может оказаться вполне достаточно. Пароли длиннее восьми символов будут усечены, при их использовании могут возникнуть проблемы с некоторыми программами, поэтому их лучше не использовать.

Придумать хороший пароль хотя и сложно, но вполне возможно. Однако я бы рекомендовал следующий путь.

1. Установите в системе программу makepasswd (ее можно найти по адресу http:// tech.ilp.physik.uni-essen.de/www.debian.org/Packages/stable/admin/makepasswd.html).

ПРИМЕЧАНИЕ

Программу makepasswd можно найти на компакт-диске, прилагаемом к данной книге.

2. Сгенерируйте несколько паролей с помощью этой программы и выберите сре ди них тот, который вам легче всего будет запомнить.

3. Измените свой пароль.

4. После этого с десяток раз выйдите и вновь войдите в систему. Через несколько часов повторите это упражнение еще раз. Оно поможет вам запомнить ваш новый пароль. Двадцати повторений обычно бывает достаточно, чтобы паль цы запомнили последовательность нажатия клавиш.

Но если выдумывать пароль самому, без помощи makepasswd, как определить, хороший он или плохой? Для этого можно использовать признаки, перечисляемые далее. Комбинация символов является хорошим паролем, если она:

- содержит специальные символы (!@#$%^&*, а также цифры) в двух или более знакоместах;
- содержит как заглавные, так и строчные буквы; - имеет длину от шести до восьми символов;
- представляет собой нечто непонятное или же комбинацию слов, разделенных специальными символами.

Комбинация символов является плохим паролем, если она:

- является словарным словом или же его модификацией (включая иностранные слова): party, fiesta, party5, fi3sta и т. п.;
- совпадает с именем кого-нибудь из ваших домашних: жены, ребенка, кота и т. п.
- представляет собой дату: 610930, 300961 и т. п.;
- взята из ваших личных данных: место рождения, номер паспорта и т. п.;
- имеет отношение к вашим увлечениям и т. п. (если это слово одно из тех, что все время вертится у вас в голове, поскольку связано с вашей личностью или с компанией, на которую вы работаете, то знающему вас будет нетрудно угадать его).

Помните, однако, что даже самый хороший пароль становится плохим, стоит только записать его на бумагу.

Когда мне нужно придумать пароли, одновременно легкие для запоминания и устойчивые ко взлому, я обычно выбираю какую-нибудь тему и использую предлагаемые ей слова в качестве основы для своих паролей. Возьмем, например, тему «растения». Используя названия растений (rose, oak, ivy и т. п.) и комбинируя их с чем-нибудь другим, скажем, цветами (red, blu, blk) и специальными символами, я с легкостью могу выдать целый ряд паролей, которые легко запомнить, но трудно взломать:

```
blk*ros3
.. blu!ivy
red#oak4
```

Чтобы проверить, насколько устойчивы ко взлому порождаемые вашей схемой пароли, имеет смысл натравить на них программу взлома паролей. Но об этом чуть далее.

Пока же вернемся к программе makepasswd. Будучи вызвана без аргументов, она возвратит шести- или восьмисимвольный пароль, который невозможно взломать простой атакой по словарю. Для взлома такого пароля придется прибегнуть к перебору всех возможных паролей, что связано с огромными вычислительными затратами. К несчастью, это пароли из разряда тех, которые пользователь обязательно запишет на бумагу и при виде которых сразу становится ясно, что это пароль. Поэтому задача донесения до пользователей необходимости сохранения паролей в секрете приобретает здесь первостепенную важность.

Программа makepasswd представляет собой простой сценарий на языке Perl, который выводит случайный пароль и, при необходимости, этот же пароль в зашифрованном виде. Например:

```
# makepasswd --char 8 --count 8 --crypt
ecuraCdKaFP4Fy.p/K9bY
dLeiVWVd    Flqcui.9L3xQI
7FSBJEFH    MkHjkpOIdSmLc
ORA2vLsv    !QYuK3Fw5Ih8U
DuSbFxDj    bB.thDEpz7Zi.
wCPOIX6v    Xe3ntRWjABCnM
SowKUgvg    Z485y6UQyMEdE
xPViT6AU    X9gm2NtZc.hK6
```

Удобство makepasswd в том, что эта программа позволяет автоматизировать процесс создания большого числа учетных записей. Например, можно написать сценарий, который бы последовательно перебирал бы всех пользователей из определенного списка (скажем, от studentl до student100), генерировал пароль, создавал учетную запись для каждого имени и помещал имена учетных записей и пароли в файл, из

которого их потом можно было бы раздавать студентам.

ПРИМЕЧАНИЕ

На компакт-диске, прилагаемом к книге, имеется примитивный сценарий, берущий из файла список имен пользователей, которых нужно добавить в систему, и создающий для них хорошие (с учетом ранее изложенных признаков) восьмисимвольные пароли. Более точно: для каждого имени пользователя из файла сценарий создает группу и пользователя (с одним и тем же номером UID/GID), генерирует и шифрует пароль и помещает его в файл /etc/shadow.

Обратите внимание, что перед началом своей работы сценарий осуществляет некоторые проверки: выполняется ли он на уровне привилегий root, не занят ли начальный UID и т. д. Тем не менее нельзя сказать, что он проверяет все[1].

Например, в нем предполагается, что перечисленные в файле имена пользователей отсутствуют в системе, иными словами, сценарий позволяет только добавлять пользователей, но не обновлять их пароли.

Некоторые вещи жестко заданы в исходном тексте — например, использование восьмисимвольных паролей. Если вы решите изменить длину пароля, не забудьте поменять значение параметра самой последней команды cut. Вообще говоря, рассматривайте этот сценарий не как нечто законченное и совершенное, но как пример, демонстрирующий один из возможных подходов к проблеме создания множества учетных записей. Так что если вам что-нибудь в нем не нравится, смело изменяйте его. Для этого он и создан. Более правильно было бы, как мне кажется, написать сценарий на expect, но я не особо силен в области expect-сценари-ев. Для тех, кто не знает, expect — это язык для написания сценариев управления интерактивными программами, позволяющий имитировать сеанс работы с программой, предполагающей интерактивное вмешательство в ее работу (например, ftp или rogue). Программа ожидает ввода с клавиатуры, однако вместо этого ей передается команда сценария, благодаря этому удается автоматизировать сеанс работы с программой. Я написал данный сценарий по одной причине: существующие программы неудобно использовать для быстрого и эффективного добавления в систему множества пользователей. Пришлось самому решать эту проблему в меру своих умений. Далее приводится текст сценария (комментарии и текстовый вывод переведены на русский язык, если ваша система не поддерживает русский, вы должны модифицировать сценарий соответствующим образом).

```
1  #!/bin/bash
2  #  Разработан  Дэвидом  Банделом  (D.Bandel)  -  3  сентября  99;  распространяется  в  соответ
ствии с GPL
# Сценарий принимает список имен пользователей и создает набор учетных записей
4  #  Зашифрованные  пароли  записываются  в  файл  /etc/shadow
5
6  prog=/usr/bin/makepasswd
7  names=/root/newusernames
8  1ogins=/root/newlogins
9  tmpshadow=/root/shadow.tmp
10 startid=1000
11
12 if [ $UID != 0 ] ; then
echo "Чтобы запустить программу $0, вы должны быть пользователем root"
14 exit 1
15 fi
16
17 if [ ! -e /etc/shadow ] ; then
18       echo "Вы не используете систему shadow"
19       exit 2

20 fi
21 if [ ! -x $prog ] ; then

22       echo "В вашей системе не установлена программа $prog"
23       exit 3

24 fi
25 if [ 'grep ":${startid}:" /etc/passwd' ] ; then
26 echo  "Пожалуйста,  измените  startid  (начальный  UID/GID)  и  запустите  сценарий
заново"
27 exit 4
28 fi
29 echo : echo "идет работа ..."
```

[1] Более того, он работает только с makepasswd версии 1.07, поскольку в более поздних версиях (1.10 на момент перевода этой книги) отсутствует параметр --clear. — *Примеч. перев.*

```
30  for i in 'cat $names'
31  do
32  j='$prog --char 8'
33  echo "${i}:${j}" >>$logins ; echo " . "
34  done
35

36  k=$startid;cp /etc/shadow /etc/shadow.orig
37  echo ; echo "работа продолжается ..."
38  for i in 'cat $logins'
39  do
40  j='echo $i | cut -d : -f 1 -'
41  l='echo $i | cut -d : -f 2 -'
42  groupadd -g $k $j
43  useradd -m -u $k -g $j $j
44  k=$[$k+l]
45  m='$prog --clear=$l --crypt | cut -b 12 --'
46  sed "s|$j:\*not set\*|$j:$m|" /etc/shadow > $tmpshadow
47  mv $tmpshadow /etc/shadow ; echo " . "
48  done
49
50  exit 0
```

Сам сценарий предельно прост. Нужно лишь создать файл /etc/newusernames, содержащий имена пользователей, которых нужно добавить в систему. После чего можно выполнять сценарий.

- Строки 6-10 инициализируют некоторые переменные, являющиеся, по сути дела, параметрами сценария. В силу характера информации, помещаемой во временные файлы, местом хранения временных файлов выбран подкаталог /root. Предполагается, что права на чтение /root имеет только суперпользователь. Возможно, вам понадобится изменить имя входного файла, содержащего имена пользователей, и начальное значение идентификатора группы и пользователя (UID/GUID).

- Строки 12-28 осуществляют некоторые простейшие проверки. Проверяется не все, но только самое важное.

- Строки 30-34 перебирают все имена пользователей и создают для них пароли, сохраняемые во временном файле для дальнейшего использования.

- Строки 38-48 создают для каждого имени частную группу пользователя, самого пользователя, шифруют пароль и помещают его в /etc/shadow.

Выполнение сценария может занять некоторое время. В основном оно уходит на генерацию программой makepasswd случайных паролей и на их последующее шифрование.

Взлом паролей

Один из способов проверки безопасности системы подразумевает, что вы ставите себя на место злоумышленника и пытаетесь думать и действовать так, как действовал бы человек, пытающийся нарушить вашу защиту. Это означает, что вы прогуливаетесь среди пользователей, подсматривая, не прикреплен ли к какому-нибудь монитору записанный пароль, не оставил ли кто-нибудь на столе бумажку с записанными на ней идентификационными данными, или же «проходите мимо» как раз в то утреннее время, когда пользователи входят в систему (быть может, удастся заметить, как кто-нибудь из них будет набирать пароль на клавиатуре).

Это также означает, что вы должны обращать внимание на ориентацию монитора пользователя, имеющего доступ к чувствительной информации, дабы выяснить, видна ли она кому-нибудь еще. Далее, когда эти пользователи отлучаются от своего рабочего места, запускают ли они заблокированную паролем программу-заставку (screen saver), а может, выходят из системы или же не делают ничего?

СОВЕТ

Однако наилучший способ проверить на прочность систему парольной безопасности и отношение пользователей к ней — попытаться взломать пароли пользователей. Регулярное выполнение программы взлома паролей способно дать достаточно хорошую оценку крепости вашей системы парольной защиты.

Хороших программ для взлома паролей написано множество, и все время появляются еще более развитые программы. Тем не менее, какой бы хорошей не была программа, взлом паролей всегда связан со значительными вычислительными и временными затратами, поэтому лучше заниматься этим ночью, когда вычислительная нагрузка на систему со стороны пользователей достигает своего минимума.

Заключение

В этой главе было продолжено рассмотрение вопросов безопасности, имеющих отношение к пользователям и группам. Сначала было рассказано о двух схемах членства пользователей в группах: группа по умолчанию и частные группы пользователей. Затем я рассказал о том, как сменить группу входа в систему и как выполнить подстановку пользователя. После чего были рассмотрены вопросы безопасности пользователей, в том числе как лучше всего защитить пользователей от самих себя и от других пользователей, обеспечив при этом требуемые возможности по доступу к файлам. Наконец, была затронута тема паролей и рассказано, какие пароли следует считать хорошими, а какие — плохими. Материал этой главы логически продолжает изложенное в первой главе. Вместе эти две главы дают сведения, необходимые для понимания материала следующих двух глав.

3 Файлы и права доступа

В данной главе рассматриваются следующие вопросы:
- какую роль файлы играют в операционной системе Linux;
- типы файлов;
- права доступа к файлам;
- настройка прав доступа по умолчанию;
- изменение прав доступа.

В этой главе рассказывается о файлах как таковых и о правах доступа к ним в частности. В ней продолжается разговор о пользователях и группах, начатый в главах 1 и 2, и рассматривается их связь с файлами и правами доступа.

Linux: файловая система

Linux — это файловая система. И потому все устройства, имеющиеся в вашей Linux-системе — принтеры, дисководы, жесткие диски и т. д., — представляются в ней в виде файлов. В результате все, что можно сделать с файлом, можно сделать и с устройством. Как и в файл, в устройство можно передавать информацию при помощи программ или каналов и аналогичным же образом получать информацию из него. Получение информации из устройства не имеет особого смысла для устройств типа принтера, который может только принимать файлы для печати. Тем не менее и экрану, на который вы смотрите, и остальным устройствам (модему, принтеру, жесткому диску, разделам диска и т. д.), каждому из них в Linux соответствует свой файл.

Управляет всем этим хозяйством ядро системы. Когда кто-то запрашивает доступ к устройству, ядро определяет, обладает ли запрашивающая сторона правами доступа к соответствующему файлу устройства. При этом происходит анализ

идентификаторов пользователя (UID), запросившего доступ, и идентификаторов всех групп (GID), к которым принадлежит данный пользователь. На основании этого анализа выносится вердикт, разрешить запрошенный доступ или нет.

Типы файлов

В Linux имеется несколько различных типов файлов. О них и пойдет речь далее. После чего мы обсудим этот вопрос с точки зрения безопасности.

Выполнив команду ls -la, можно получить много разнообразной информации о файлах каталога, в том числе и о том, к какому типу принадлежит тот или иной файл. Рассмотрим список файлов некоего каталога, в котором содержатся файлы различных типов:

```
drwxr-xr-x24 root    root        2048 Sep 4 00:01 .
drwxr-xr-x20 root    root        1024 Aug 26 19:09 ..
drwxr-xr-x3 root     root        1024 Jul 22 22:17 .civctp
crw-rw-r--1 root     root       29,  0 Aug 5 09:12 fbO
brw-rw-rw-1 root     root        2,  0 Jul 27 19:14 fdO
-rw-r--r--  1 root   root         694 Sep 2 21:02 foo
srwxrwxrwx1 root     root           0 Sep 3 19:18 mysql.sock
prw------- 1 root    root           0 Sep 3 19:14 initctl
lrwxrwxrwx1 root     root           4 Aug 5 08:49 sh -> bash
```

Обратите внимание на самый левый столбец. В этом столбце содержатся символы d, d, d, c, b, -, s, p, l. Каждый из этих символов соответствует определенному типу файла (см. табл. 3.1). Символ «d» обозначает каталог (от английского слова *«directory»)*. Каталоги тоже являются файлами, однако некоторые файловые

операции для них просто не имеют смысла и потому не приводят к ожидаемым результатам. Самые первые два каталога списка, то есть каталоги с именами «.» и «..», имеют предопределенное значение. Точка всегда соответствует текущему каталогу, две точки — родительскому каталогу. Исключением является корневой каталог (/), для которого имя «..» соответствует ему самому, а не каталогу верхнего уровня. Подкаталог с именем «..» присутствует в каждом каталоге файловой системы Linux, благодаря чему для перемещения вверх по дереву каталогов всегда можно использовать простую команду «cd ..». Два специальных каталога с именами «.» и «..» можно рассматривать как ссылки, более того, они ими и являются. Так как эти два каталога являются ссылками, то эти ссылки можно разорвать. В этом случае команда cd .. будет вести себя так же, как и для корневого каталога, то есть при ее выполнении вы останетесь в том же каталоге, в котором и были. Для осуществления низкоуровневых операций над диском, которые невозможно выполнить обычными средствами, можно воспользоваться программой debugfs. В частности, эту программу можно использовать для разрыва ссылки «..», однако следует помнить, что многие программы и сценарии полагаются на эту ссылку, так что действовать нужно обдуманно.

ВНИМАНИЕ

Разрыв ссылки может иметь далеко идущие последствия. Поэтому не делайте этого, если вы не до конца понимаете, к чему могут привести ваши действия.

Таблица 3.1. Типы файлов

Обозначение типа	Тип	Примеры
	Обычный файл	Файлы данных, текст, программы
d	Каталог	/bin
b	Блочное устройство	/dev/hda (первый жесткий диск на первом IDE-интерфейсе)
c	Символьное устройство	/dev/ttyS1 (аналог com2 в DOS)
s	Сокет	/dev/log
p	Именованный канал (named pipe)	/dev/inictl (именованный эквивалент операции «|»)
I	Символическая ссылка	/dev/modem -> /dev/ttyS1

Но вернемся к списку файлов. В нем присутствует еще один каталог, в начале имени которого стоит символ точки (.). Информация о файлах, имена которых начинаются с точки (по-английски их называют *dot files),* при обычном запуске команды ls не отображается. Чтобы получить информацию о таких файлах, необходимо в командной строке ls указать ключ -a. В остальном это вполне обычные файлы, большинство которых являются файлами конфигурации. Информация об этих файлах скрывается не из соображений безопасности, а для удобства. Как правило, такие файлы являются служебными (их используют многие программы, например командная оболочка bash), поэтому обычные пользователи напрямую с ними не работают. Чтобы эти файлы лишний раз не отвлекали внимание обычных пользователей, операционная система скрывает их, отображая информацию о них только тогда, когда это действительно нужно. Однако не стоит использовать этот механизм для того, чтобы утаить что-либо от пользователей. Если вы хотите блокировать доступ к какому-либо файлу, лучше использовать для этой цели соответствующую конфигурацию разрешений на доступ.

Следующие два файла (помеченные буквами c и b) являются файлами устройств. Первое устройство является символьным, второе — блочным. Традиционно все файлы устройств располагаются в каталоге /dev. При желании, однако, их можно хранить и в другом месте, например, на гибком диске или в оперативной памяти (на виртуальном диске). Когда ядро получает запрос на использование файла устройства, оно сначала узнает из информации о файле, какой драйвер (модуль) управляет этим устройством, после чего данный драйвер используется для работы с устройством в соответствии с правами доступа, назначенными файлу этого устройства. Различие между символьными и блочными устройствами заключается в том, что символьные устройства передают данные последовательно, символ за символом, тогда как блочные устройства передают данные параллельно, обычно 8 байт за раз, и не посимвольно, а целыми блоками.

Тут вы должны были бы насторожиться. Если доступ к устройству осуществляется через файлы, то выходит, что любой пользователь может создать файл устройства, например, в своем домашнем каталоге или на гибком диске и смонтировать его. Таким образом он получит это устройство в полное свое распоряжение. Совершенно верно. Поэтому крайне не рекомендуется разрешать обычным пользователям бесконтрольно создавать и монтировать файлы устройств. На самом деле не все так просто. Файлы устройств считаются специальными, поэтому только пользователь root (иначе говоря, обладатель нулевого идентификатора пользевателя) может выполнять предназначенную для создания файлов устройств программу mknod. Таким образом, от вас требуется обеспечить, чтобы никто не смог смонтировать раздел (на жестком диске или просто гибкий диск), содержащий активные файлы устройств. Хорошей мерой предосторожности является также проверка наличия файлов устройств в подкаталогах каталога /home. Для

этой процедуры можно использовать следующие две строчки:

```
find /home -type b -print
find /home -type c -print
```

ССЫЛКА

Подробней о монтировании файловых систем рассказывается в главе 5.

На самом деле вопрос, нужны ли обычному пользователю файлы устройств в его домашнем каталоге, является спорным. Настолько спорным, что вам обязательно следует обсудить его с пользователями, в домашних каталогах которых обнаружились эти файлы.

Следующий файл, с типом «-», является обычным файлом. К категории обычных файлов относится большинство файлов в системе Linux. В эту категорию входят исполняемые файлы, файлы данных или файлы с текстовым содержимым (обыкновенные текстовые файлы ASCII, файлы сценариев, многие конфигурационные файлы и т. п.).

В следующей строке листинга показан файл типа «s», то есть сокет. В данном случае это сокет программы mysql Среди сведений, выводимых на экран по команде netstat -a, можно обнаружить следующую строку:

```
unix O [ACC] STREAM LISTENING 12210 /tmp/mysql.sock
```

Таким образом вы можете убедиться в том, что это действительно активный сокет, через который программа ожидает поступления данных (сокет относится к категории LISTENING). Подробней о сокетах рассказывается в следующих главах книги.

ССЫЛКА

Команда netstat обсуждается в главе 10.

После сокета размещается информация о файле, являющимся *именованным каналом* (named pipe). Файлы этого типа обозначаются символом «p». Назначение именованного канала точно такое же, как и у любого другого канала: он передает данные, выводимые одной программой, на вход другой программы.

Последний файл листинга является *символической ссылкой* (symbolic link). Символическую ссылку иногда называют *мягкой ссылкой* (soft link). Файл символической ссылки — это указатель, который указывает на другой файл или каталог. Файл, на который указывает ссылка, называется целевым файлом (target file). Содержимое самого целевого файла в файле символической ссылки не хранится, файл символической ссылки содержит в себе только сведения о том, в каком месте файловой системы расположен целевой файл. Поэтому удаление целевого файла оставляет все указывающие на него символические ссылки в подвешенном состоянии, поскольку они более никуда не указывают. Файл символической ссылки всегда создается без каких-либо ограничений на доступ, однако при работе с символической ссылкой используются права на доступ к файлу, на который указывает эта ссылка. Для Macintosh приблизительным аналогом символической связи является псевдоним (alias), а для Windows — ярлык (shortcut).

ПРИМЕЧАНИЕ

На диске информация о файле хранится в блоке, называемом индексным дескриптором или inode (information node). Файл принадлежит каталогу, если номер его индексного дескриптора встречается среди номеров индексных дескрипторов, перечисленных в файле каталога. Индексный дескриптор содержит всю важную информацию о файле, такую как номер индексного дескриптора, тип файла, права доступа, номер версии, идентификаторы группы и пользователя, размер, время последнего доступа (atime), время создания (ctime), время модификации (mtime), номера блоков, содержащих данные файла, и прочее. Содержимое индексного дескриптора можно посмотреть с помощью программы debugfs.

Кроме символической ссылки в Linux существует еще один похожий тип файлов, который называется *«жесткая ссылка»* (hard link) или просто ссылка (link). Понять смысл этой концепции поможет другой список файлов, содержащий информацию, выводимую при указании ключей -i и -I:

```
20512 -rwxr-xr-x  3 root   root     49280 Jul 27 19:37 gunzip
20512 -rwxr-xr-x  3 root   root     49280 Jul 27 19:37 gzip
20512 -rwxr-xr-x  3 root   root     49280 Jul 27 19:37 zcat
```

Сведения, выдаваемые командой ls при указании этих ключей, практически идентичны предыдущим,

только в данном случае слева появился новый столбец, в котором для всех трех записей указано число 20512. В этом столбце выводится номер индексного дескриптора (inode) файла. Индексный дескриптор — это место, в котором хранится вся важная информация о файле. Говоря иначе, в индексном дескрипторе хранится вся информация о файле, выводимая командой ls, а также другая важная информация. Обратите внимание, что три файла с различными именами обладают одним и тем же индексным дескриптором (в самом левом столбце стоит одно и то же число). Это означает, что три различных файловых имени на самом деле соответствуют одному и тому же файлу. Если посмотреть на размер файла, то можно убедиться, что во всех трех случаях он один и тот же, а именно 49 280 байт. Таким образом, команда ls выводит не список файлов каталога, а список принадлежащих каталогу ссылок на файлы. Файл же представляет собой некоторый набор данных, идентифицируемый при помощи номера индексного дескриптора (inode). Получается, что сам по себе файл не привязан к какому-либо каталогу и не обладает именем, однако на него может указывать одна или несколько ссылок, которые могут располагаться в разных местах файловой системы. Файл существует, пока есть хотя бы одна указывающая на него ссылка. Поэтому только удаление самой последней из всех указывающих на файл ссылок является действительно значимым и приводит к потере информации, содержащейся в этом файле. Число, содержащееся в столбце непосредственно слева от первого появления слова root, соответствует количеству ссылок, указывающих на файл. Например, если удалить файл zcat, то с файлами gunzip и gzip ничего не случится, однако вместо числа 3 в третьем столбце будет стоять число 2. На вновь созданный каталог всегда указывают две ссылки. Первая из них содержится в родительском каталоге. Вторая содержится в самом созданном вами каталоге. Эта ссылка обладает именем «.» и указывает на каталог, в котором она содержится. Если счетчик ссылок, указывающих на некоторый каталог, равен единице, значит, для этого каталога ссылка с именем «.» разорвана. Напомню, что помимо ссылки с именем «.» во вновь созданном каталоге автоматически появляется ссылка с именем «..», которая указывает на родительский каталог.

Жесткие ссылки ссылаются на номера индексных дескрипторов (inode), каждый из которых уникален в пределах одной файловой системы, поэтому жесткие ссылки не могут указывать на файл, расположенный в другой файловой системе или на другом разделе диска. Символические ссылки, напротив, могут указывать на файлы, располагающиеся в других разделах и на других дисках. Владельцем символической ссылки назначается создавший ее пользователь, но для определения прав на доступ используется набор разрешений целевого файла. Таким образом, хотя создать символическую ссылку, указывающую на файл, может любой пользователь, при использовании этой ссылки для доступа к целевому файлу система может отказать этому пользователю в доступе. Решение будет приниматься на основании набора разрешений целевого файла. Если у пользователя, обращающегося по ссылке, недостаточно прав для доступа к целевому файлу, ему будет отказано, хотя, повторюсь, сама символическая ссылка, то есть файл этой ссылки, не имеет никаких ограничений на доступ. При обращении по жесткой ссылке информация о правах на доступ к файлу и сведения о владельце файла извлекаются из индексного дескриптора файла.

Базовые разрешения на доступ к файлу

Взгляните на еще один листинг:

```
-rwxr-xr-x    I root    root    3164 Jul 27 20:27 arch
-rwxr-xr-x    1 root    root    317504 Jul 27 22:22 bash
-rwxr-xr-x    1 root    root    464564 Jul 28 02:54 bash2
•rwxr-x---    1 root    root    34236 Jul 28 09:37 box
-rwxr-xr-x    1 root    root    11856 Jul 28 09:37 build_menu
Irwxrwxrwx    1 root    root    5 Aug 5 08:50 bunzip2 -> bzip2
Irwxrwxrwx    1 root    root    5 Aug 5 08:50 bzcat -> bzip2
-rwxr-xr-x    1 root    root    57404 Jul 27 20:11 bzip2
-rwxr-xr-x    1 root    root    7152 Jul 27 20:11 bzip2recover
-rwxr-xr-x    1 root    root    9240 Jul 27 21:12 cat
-rwxr-xr-x    1 root    root    11692 Jul 27 21:38 chgrp
-rwxr-xr-x    1 root    root    11820 Jul 27 21:38 chmod
-rwxr-xr-x    1 root    root    12164 Jul 27 21:38 chown
-rwxr-xr-x    1 root    root    28204 Jul 27 21:38 cp
-rwxr-xr-x    1 root    root    47448 Jul 27 19:32 cpio
Irwxrwxrwx    1 root    root    4 Aug 5 09:02 csh -> tcsh
-rwxr-xr-x    1 root    root    38132 Jul 27 22:59 ctags
```

Как вы уже знаете, самый первый столбец (в этом примере в нем содержатся символы «-» или «l») позволяет определить тип файла. В данном случае мы имеем дело с обычными файлами и с символическими ссылками. Следующие девять столбцов можно разбить на три группы по три столбца каждая. Эти девять столбцов содержат информацию о разрешениях на доступ к файлу. В Linux эти

сведения называются также *режимом* (mode) файла. Первые три столбца определяют режим доступа для владельца файла, следующие далее три столбца определяют режим доступа для группы, наконец, последние три столбца определяют режим доступа для всех остальных пользователей. Символ «r» означает право на чтение (Read), «w» — на запись (Write), «x» — выполнение (eXecute). Таким образом, для большинства файлов из этого листинга назначены следующие наборы разрешений: владельцу файла разрешается чтение, запись и выполнение (rwx), группе, обладающей файлом, разрешается чтение и выполнение (r-x), остальным пользователям также разрешается чтение и выполнение (r-x). На уровне индексного дескриптора файлов (inode) разрешения на доступ обозначаются тремя восьмеричными числами. Для обозначения разрешений на доступ к файлу используется диапазон значений от 0 до 7, или, в двоичной системе исчисления, от 000 до 111. Каждый из битов этого числа соответствует одному из трех разрешений: «rwx». Разрешению на выполнение соответствует первый (самый младший) бит, разрешению на запись соответствует второй бит, на чтение — третий. Если изначально все три бита сброшены, то чтобы разрешить исполнение файла, необходимо добавить к этому значению единицу (1), чтобы разрешить чтение, следует добавить двойку (2), а чтобы разрешить запись — следует добавить четверку (4). Сумма любых комбинаций этих трех значений дает значение из диапазона от 0 до 7. Таким образом, полный набор разрешений на доступ к файлу (режим файла) может быть идентифицирован числом из диапазона от 000 до 777. Первый разряд этого числа определяет права владельца файла, второй — права группы владельца, а третий — права всех остальных пользователей. Рассмотренному нами ранее набору разрешений «rwxr-xr-x» соответствует число 755. Такой режим доступа устанавливается для большинства утилит, доступных для всех пользователей системы. Чтобы назначить файлу этот набор разрешений, необходимо выполнить команду chmod 755 имя_файла. Описание разрешений на доступ содержится в табл. 3.2.

Таблица 3.2. Разрешения на доступ в восьмеричном виде

Разрешения на доступ	Восьмеричный эквивалент	Комментарии
--------	000	Числа соответствуют трем группам разрешений на доступ
- -x- -x—x	111	Каждое такое число получается путем сложения чисел (4, 2 и 1), соответствующих установленным разрешениям
-w - -w - - w -	222	rwxr-x--x = 751
-wx-wx-wx	333	
r--r--r--	444	
r-xr-xr-x	555	
rw-rw-rw-	666	
rwxrwxrwx	777	

Вслед за разрешениями на доступ в листинге располагается столбец числа ссылок, за которым следуют столбцы имен владельца и группы (в оставшихся столбцах выводится размер, дата и время модификации, а также имя файла). Когда ядро операционной системы получает от пользователя, программы или командной оболочки запрос на доступ к файлу, оно первым делом смотрит на UID субъекта, запросившего доступ. Если этот UID совпадает с идентификатором (не именем) владельца файла, ядро определяет набор допустимых операций в отношении файла на основании того, какие из первых трех битов набора разрешений для данного файла установлены. Если UID обращающегося к файлу субъекта не совпадает с UID владельца файла, осуществляется сравнение GID субъекта, обращающегося к файлу с GID группы, к которой принадлежит файл. Если эти два идентификатора совпадают, для определения набора допустимых операций используются разрешения, предоставленные группе, владеющей файлом. Если ни UID, ни GID не совпадают, то субъект, запросивший доступ к файлу, попадает в категорию «остальные». В этом случае используются разрешения, определяемые третьей группой битов. В качестве примера рассмотрим четвертую строку листинга. Разрешения на доступ здесь таковы: владелец — чтение, запись, выполнение; группа — чтение, выполнение; «остальные» — прав нет (для данного примера я немного изменил эту строку). Таким образом, режим доступа к файлу равен 750. Владельцем файла является пользователь root, поэтому если доступ к файлу box запросит пользователь с нулевым значением UID (например, пользователь root), ему будет предоставлено разрешение на чтение, запись и выполнение этого файла. Если UID запросившего доступ не равен нулю, анализу подвергается его GID. Если GID запросившего доступ равен нулю, пользователь сможет читать и выполнять файл, однако не сможет записывать в него данные. Если ни UID, ни GID обратившегося к файлу пользователя не равны нулю, этому пользователю будет отказано в доступе, так как для категории «остальные» никаких разрешений на доступ к этому файлу не предоставляется.

ПРИМЕЧАНИЕ

Сопоставление прав доступа по идентификаторам завершается при первом же совпадении.

При анализе набора разрешений сравнение идентификаторов выполняется до тех пор, пока не будет найдено первое совпадение. Это означает, что если UID пользователя совпадает с UID владельца файла и этот пользователь пытается получить доступ к файлу с режимом 055, ему будет отказано в доступе, даже если этот пользователь принадлежит к группе, владеющей файлом. Режим доступа 055 означает, что владельцу файла не предоставлено никаких разрешений на доступ к файлу, при этом группе, владеющей файлом (впрочем, как и всем остальным пользователям), предоставлены разрешения на чтение и выполнение файла. Определение прав на доступ, предоставленных пользователю, завершится на первом совпадении идентификаторов, поэтому если пользователь является владельцем файла, групповые права для него рассмотрены не будут. Таким образом, возможны ситуации, когда пользователь является владельцем файла, но доступ к нему получить не может.

Однако удалить такой файл все равно можно. При отсутствии у владельца доступа на запись в файл попытка его удаления приведет к выдаче ядром запроса «Delete file overriding mode 0055? (y/n)», что в переводе означает: «удалить файл несмотря на режим доступа 0055 (да/нет)». В результате положительного ответа на этот запрос файл будет удален. Однако в данном случае владелец файла не сможет выполнить в отношении этого файла никаких других операций.

Если вы хотите выполнить сценарий командной оболочки, вы должны обладать в отношении соответствующего файла как правом на выполнение, так и правом на чтение. Однако чтобы выполнить компилированную программу, достаточно обладать правом на ее выполнение. Причина в том, что компилированные программы запускаются ядром операционной системы, пользователь не сможет запустить их как-либо иначе. В то же время для того, чтобы выполнить сценарий оболочки, необходимо обладать возможностью чтения инструкций, содержащихся в файле сценария. Если пользователь не обладает такой возможностью, он не сможет выполнить сценарий. Однако если пользователь обладает правом чтения файла сценария и при этом не обладает правом выполнения этого файла, он все равно сможет выполнить этот файл, задав его имя в качестве параметра интерпретирующей программы:

```
perl   myperlsript.pl
или sh myshellscript.sh
```

ПРИМЕЧАНИЕ

Имя сценария оболочки не обязано заканчиваться на «.sh», равно как и имя сценария на языке Perl — на «.pl». Это просто соглашение, используемое автором, чтобы отличать файлы сценариев оболочки от файлов сценариев на языке Perl.

Определить, чем является данный файл — исполняемым бинарным файлом, сценарием оболочки, текстовым документом или чем-то еще, поможет программа file. Признаки, по которым программа определяет вид файла, хранятся в /usr/ share/misc/magic. Определения новых видов файлов следует помещать в /etc/magic.

Разрешения на доступ для каталогов трактуются несколько иначе, чем для файлов. Как и у любого другого файла, у каталога есть владелец и группа. Однако для каталогов права на запись, чтение и выполнение имеют другое значение, не такое, как для файлов. Наличие права на чтение позволяет тому, для кого оно установлено, то есть владельцу, группе и/или остальным пользователям просматривать содержимое каталога. Иными словами, если вы обладаете правом чтения каталога, значит, вы можете получить список имен файлов, содержащихся в этом каталоге. Но то обстоятельство, что вы можете прочитать имя файла, вовсе не означает, что вы можете прочитать его содержимое. Сами файлы, содержащиеся в этом каталоге, могут оказаться недоступными для чтения.

Разрешение на запись позволяет тому, для кого оно установлено, записывать что-либо в каталог. Этого разрешения может оказаться, однако, недостаточно для изменения уже существующего файла или его удаления. Возможность изменения или удаления файла определяется специальными разрешениями, устанавливаемыми для этого конкретного файла. Но вы можете создавать в каталоге новые файлы.

Наличие права на выполнение позволяет переходить в этот каталог при помощи команды cd. Если вы не обладаете правом выполнения каталога, это не лишает вас возможности читать список файлов каталога, создавать в нем новые файлы или удалять существующие. Однако войти в этот каталог, не имея доступа на выполнение, вы не сможете.

Примером каталога, для которого имеет смысл не устанавливать все разрешения одновременно, является каталог incoming на ftp-сервере. Обычно этому каталогу назначают следующие разрешения на доступ:

```
drwx-wx-wx   2 root      ftp          1024 Jul    8 12:47 incoming
```

При таких правах любой пользователь ftp-сервера сможет поместить файлы в этот каталог, но видны они будут только суперпользователю и только суперпользователь сможет прочесть их. Для любого другого пользователя каталог будет выглядеть пустым. Это позволяет оградить ftp-сервер, поддерживающий закачку файлов, от нежелательного использования этой возможности анонимными пользователями

(например, для обмена файлами через него, когда один пользователь закачивает их, а другой после скачивает).

ПРИМЕЧАНИЕ

Предоставление юридических советов не входит в задачи данной книги. Тем не менее примите к сведению, что есть законы и по этой части, и потому владельцы системы, системные администраторы и/ или пользователи несут юридическую ответственность за незаконные действия, производимые с системой. Это означает, что из-за вашей халатности как системного администратора у вас могут возникнуть серьезные проблемы с законом.

Режим доступа по умолчанию

Каждый файл обязан иметь владельца, группу и режим доступа. Поэтому эти три элемента всегда по тем или иным правилам назначаются каждому создаваемому файлу. В нормальных условиях владельцем созданного файла назначается создавший его пользователь. В качестве группы, владеющей файлом, назначается группа входа в систему этого пользователя. Режим доступа к файлу, назначаемый ему по умолчанию, определяется при помощи значения umask (user mask) пользователя.

Значение umask — это восьмеричное число, которое конъюнктируется либо с числом 0777, либо с числом 0666 (в зависимости от типа файла), в результате получается набор разрешений на доступ, назначаемый создаваемому файлу по умолчанию. Для тех, кто не знаком с булевой арифметикой, приведу другую формулировку: значение umask, рассматриваемое как восьмеричное число, вычитается либо из 0777, либо из 0666, в зависимости от типа файла. Полученное в результате вычитания число и является начальными правами доступа к файлу.

Тип файла влияет на начальный режим доступа следующим образом: если команда file показывает, что созданный файл является бинарным исполняемым файлом, значение umask вычитается из 0777, для всех остальных файлов значение umask вычитается из 0666.

В OpenLinux при входе пользователя в систему выполняется файл из/etc/config.d/ shells (обычно это bashrc), который и устанавливает значение umask пользователя. Для суперпользователя это значение равно 022, для обычного пользователя — 002. Таким образом, бинарным исполняемым файлам, создаваемым суперпользователем, по умолчанию назначается режим 755 (rwxr-x-r-x), а всем остальным файлам назначается режим 644 (rw-r--r--). Бинарным файлам, создаваемым обычными пользователями, назначается режим 775 (rwxrwxr-x), а всем остальным файлам назначается режим 664 (rw-rw-r--). Узнать текущее значение umask можно, выполнив команду umask без аргументов. Для изменения значения umask нужно вызвать эту команду с новым значением в качестве аргумента (например, umask 222).

Изменение разрешений на доступ к файлу

Управлять доступом к файлу позволяют команды chown и chmod. Первая из них меняет владельца файла, а вторая — режим доступа к файлу. Изменять владельца файла и режим доступа к файлу может только пользователь, обладающий соответствующими правами на доступ к файлу или каталогу. Если у вас нет разрешения на запись в файл или каталог, вы не сможете применить к нему эти команды. Команда chown принимает в качестве аргументов имена владельца и/или группы, за которыми следует имя файла или каталога, для которого их надо установить. Имя группы отделяется от имени владельца точкой. Например, допустим, что владельцем файла foo является пользователь silvia и что этот файл принадлежит группе silvia. Если вы намерены сделать владельцами этого файла пользователя root и группу gifs, вы должны выполнить следующую команду:

```
chown root.gifs foo
```

Если требуется изменить только имя владельца или группы, в команде можно указать только root или .gifs.

Команда chmod меняет режим доступа к файлу. Как и при использовании предыдущей команды, если у вас нет доступа на запись в файл или каталог, вы не сможете изменить его режим. Режим файла можно задавать в восьмеричном представлении, о котором говорилось ранее. Например, если значение umask для вас равно 002, то созданный вами файл сценария изначально будет обладать режимом 664. Попытка выполнения такого файла приведет к сообщению об ошибке «permission denied» (разрешение отсутствует). Вполне предсказуемый результат, поскольку файл сценария является не бинарным, а текстовым файлом, и потому разрешение на выполнение этого файла по умолчанию не назначается. Добавление права на выполнение означает изменение режима файла с 664 на 755 (rwxr-xr-x). Чтобы изменить режим указанным образом, следует выполнить команду:

```
chmod 755 filename
```

Иногда восьмеричное представление является не слишком удобным, например, если требуется изменить всего один бит режима. В таком случае можно воспользоваться другим способом настройки режима: для настройки режима в команде chmode указывается буква, оператор и еще одна буква.

Принцип здесь следующий: первая буква, а вернее буквы, задают, для кого следует изменить режим. Для этой цели используются буквы из набора u, g, o, a, где и обозначает владельца, g — группу, o — всех остальных пользователей, a — все три категории. После этого указывается один из операторов: +, = или -. Оператор + устанавливает указанный за ним бит, оператор - сбрасывает указанный за ним бит, а оператор = сбрасывает все биты, кроме указанного. После оператора указывается модифицируемый бит (r, w или x). Вернемся к примеру со сценарием. Если вы решили, что на самом деле вам нужен режим не 755, а 775, то добавить недостающий режим 020 можно командой:

```
chmod g+w filename
```

Этот метод позволяет устанавливать или сбрасывать лишь некоторые биты режима, оставляя остальные без изменений.

Заключение

В этой главе вы ознакомились с основами безопасности файлов в Linux. Вы узнали о типах файлов, разрешениях на доступ к файлам и о том, как их изменить. Вы также узнали, каким образом файлу назначается режим доступа по умолчанию и как этим управлять.

Следующая глава закрывает тему прав доступа к файлам, рассказывая о дополнительных правах, о которых не рассказывалось в данной главе.

4 Атрибуты SUID/SGID для файлов и каталогов

В данной главе рассматриваются следующие вопросы:
- атрибуты SUID и SGID для файлов и каталогов;
- потенциальная опасность применения атрибутов SUID/SGID;
- контроль файлов с атрибутами SUID/SGID;
- изменение разрешений на доступ;
- специальные атрибуты файловой системы ext2;
- использование команды chattr;
- использование команды Isattr.

Помимо базовых атрибутов файлов и каталогов, определяющих конфигурацию разрешений на чтение, запись и выполнение, в файловой системе Linux используются некоторые дополнительные атрибуты, о которых будет рассказано в данной главе. Мы рассмотрим, для чего нужны эти дополнительные атрибуты, как устанавливать их с помощью команды chmod и в чем заключается опасность их использования с точки зрения безопасности.

После этого я расскажу вам об атрибутах, поддерживаемых только в файловой системе ext2, их назначении и модификации.

Атрибуты SUID/SGID

В главе 1 я рассказал вам о файле /etc/shadow, в котором содержатся хэширован-ные значения всех паролей системы. Чтение и модификация этого файла разрешаются только суперпользователю. Тем самым этот файл ограждается от посягательств злоумышленников, которые могут воспользоваться содержащимися в нем сведениями для получения реальных паролей системы. Кроме хэшированных паролей в этом файле хранится число дней, по истечении которых пользователю будет предложено сменить свой пароль. Однако если изменять этот файл дозволено только суперпользователю, то каким же образом обычный пользователь сможет изменить свой пароль? Очень просто. Атрибуты команды passwd таковы, что она всегда выполняется от имени суперпользователя и потому имеет полный доступ к файлу паролей.

При нормальных условиях исполняемый файл выполняется от имени вызвавшего его пользователя, то есть имеет те же самые привилегии или ограничения, что и пользователь. Однако если установить у исполняемого файла специальный атрибут, называемый SUID (Set User ID), то независимо от того, кто запускает программу, эта программа всегда будет выполняться от имени своего владельца.

Когда речь заходит о программах с атрибутом SUID, то, как правило, подразумеваются программы, выполняемые от имени суперпользователя, поскольку таково наиболее распространенное использование этого атрибута. Как правило, этот атрибут устанавливается для того, чтобы обычный пользователь мог запускать программы, которым для выполнения их функций необходимы привилегии суперпользователя.

Однако этот атрибут может понадобиться и обычному пользователю, как способ сделать доступной для остальных программу, которой иначе может пользоваться только он. Такими программами могут быть, например, средства доступа к персональным базам данных пользователя, которые этот пользователь желает сделать доступными для других пользователей.

Необходимо заметить: установка атрибута SUID в отношении файла сценария ничего не меняет. Сценарий, как и прежде, будет выполняться от имени вызвавшего его пользователя, поскольку установка этого атрибута действует только для бинарных исполняемых файлов. Чтобы добиться SUID-эффекта для сценария, можно установить атрибут SUID на все вызываемые им программы или же вызывать их при помощи специальной программы-оболочки (которая легко пишется на языке С), обладающей атрибутом SUID.

Помимо атрибута SUID есть схожий с ним атрибут SGID. Файл, для которого установлен этот атрибут, всегда выполняется от имени группы, которая обладает этим файлом. Для пользователя, который хочет сделать доступной для остальных свою личную программу, применение атрибута SGID зачастую оказывается лучшим решением, чем использование атрибута SUID.

Для каталогов атрибут SGID имеет иное значение. Обычно он устанавливается для каталога, содержащего файлы, используемые совместно несколькими пользователями. Если у каталога установлен атрибут SGID, то любому файлу, созданному в нем, в качестве группы-владельца назначается группа этого каталога, а не группа создавшего файл пользователя. Поэтому если установить этот атрибут у каталога с gif-файлами и сделать владельцем этого каталога группу gifs, то всем файлам, создаваемым в нем, в качестве владельца будет назначаться группа gifs. Если блокировать доступ к этому каталогу для всех остальных пользователей, за исключением владельца и группы, то размещенные в каталоге файлы будут доступны только пользователям, принадлежащим к группе gifs.

Последним атрибутом, который необходимо упомянуть, является так называемый «sticky bit» (липкий бит). Устанавливается он аналогично атрибутам SUID и SGID, о чем будет рассказано далее. Для исполняемых файлов этот атрибут имеет скорее историческое, чем практическое значение: в прошлом он предписывал ядру выгружать завершившуюся программу из памяти не сразу же, а лишь спустя некоторое время, что позволяло избежать постоянной загрузки с диска наиболее часто вызываемых программ. Теперь же завершившаяся программа выгружается из памяти лишь тогда, когда возникает нехватка памяти, поэтому для Linux никакой смысловой нагрузки этот атрибут не несет (в некоторых других Unix-системах он используется, но для иных целей). Для каталога этот атрибут означает, что только владельцам дозволено изменять содержащиеся в нем файлы. Если пользователь не является владельцем файла, то изменить его он не сможет. Далее мы еще рассмотрим, где и как используется этот атрибут.

Опасность применения атрибутов SUID/SGID

Неразумное количество SUID-программ, разбросанных по всей системе, может привести к проблемам. Как говорилось ранее, для некоторых программ этот атрибут является действительно необходимым, поскольку без него обычные пользователи не смогут воспользоваться этими программами. Однако список этих программ не должен содержать ничего лишнего.

В частности, в нем не должно быть программ, позволяющих пользователю выполнять произвольные команды или обращаться к командной оболочке. Как и у всякого правила, здесь могут быть свои исключения, но общий подход в данном случае следующий: если нет действительно весомых причин назначать некоторому файлу атрибут SUID, значит, делать этого не следует. Кроме того, вряд ли имеет смысл разрешать пользователям иметь SUID-программы в своих домашних каталогах.

Перечень SUID-файлов системы можно получить при помощи команды:

```
find / -perm +u+s -exec ls -l {} \:
```

В моей системе присутствуют следующие файлы SUID:

```
-r-sr-xr-x    I root    root    2105164 Aug 30 07:25 /usr/local/bin/vmware
-PWS--X--X    1 root    root    606031 Jul 22 20:07 /usr/local/bin/ssh1
---S--X--X    1 root    bin     1647592 Jul 25 13:34 /usr/local/bin/xlock
-rwsr-xr-x    1 root    root    285151 Aug 21 17:23 /usr/local/sbin/mtr
-rwsr-xr-x    1 root    root    17134 Aug 27 10:55 /usr/XllR6/bin/cardinfo
-PWS--X--X    1 root    root    7150 Jul 27 20:25 /usr/X11R6/bin/Xwrapper
-rwsr-xr-x    1 root    root    177280 Jul 28 00:32 /usr/bin/lpq
-rwsr-xr-x    1 root    root    235672 Jul 28 00:32 /usr/bin/lpr
-rwsr-xr-x    1 root    root    171060 Jul 28 00:32 /usr/bin/lprm
-rwsr-xr-x    1 root    root    14576 Jul 27 19:17 /usr/bin/rcp
-rwsr-xr-x    1 root    root    10512 Jul 27 19:17 /usr/bin/rlogin
-rwsr-xr-x    1 root    root    7840 Jul 27 19:17 /usr/bin/rsh
-r-sr-xr-x    1 root    bin     8703 Jul 27 19:16 /usr/bin/passwd
-rws--x--x    2 root    root    556924 Jul 28 03:05 /usr/bin/suidperl
-rws--x--x    2 root    root    556924 Jul 28 03:05 /usr/bin/sperl5.00502
-rwsr-xr-x    1 root    root    38752 Jul 27 20:15 /usr/bin/chage
-rwsr-xr-x    1 root    root    28244 Jul 27 20:15 /usr/bin/expiry
-rwsr-xr-x    1 root    root    32268 Jul 27 20:15 /usr/bin/gpasswd
-rwsr-xr-x    1 root    root    28320 Jul 27 20:15 /usr/bin/newgrp
-rwsr-xr-x    1 root    root    13872 Jul 27 20:27 /usr/bin/chfn
-rwsr-xr-x    1 root    root    13712 Jul 27 20:27 /usr/bin/chsh
-r-sr-sr-x    1 uucp    uucp    125852 Jul 27 20:53 /usr/bin/cu
-r-sr-xr-x    1 uucp    uucp    91568 Jul 27 20:53 /usr/bin/uucp
-r-sr-sr-x    1 uucp    uucp    38492 Jul 27 20:53 /usr/bin/uuname
-r-sr-xr-x    1 uucp    uucp    99656 Jul 27 20:53 /usr/bin/uustat
-r-sr-xr-x    1 uucp    uucp    92720 Jul 27 20:53 /usr/bin/uux
---S--X--X    1 root    root    23420 Jul 27 19:18 /usr/bin/crontab
-rwsr-xr-x    1 root    root    173920 Jul 28 00:32 /usr/sbin/lpc
-rwsr-xr-x    1 root    root    12544 Jul 27 20:48 /usr/sbin/ppplogin
-rwsr-xr-x    1 root    root     17212 Jul 27 19:18 /usr/sbin/sliplogin
```

```
-r-sr-xr-x    1 root     root        21146 Jul 27 19:22 /usr/sbin/traceroute
-r-sr-xr-x    1 uucp     uucp       221348 Jul 27 20:53 /usr/sbin/uucico
-r-sr-xr-x    1 uucp     uucp       101852 Jul 27 20:53 /usr/sbin/uuxqt
-r-sr-x---    1 news     uucp       101368 Jul 27 23:22 /usr/libexec/inn/bin/rnews
-r-sr-x---    1 root     news        51732 Jul 27 23:22 /usr/libexec/inn/bin/startinnfeed
-rwsr-xr-x    1 root     root        16700 Jul 27 21:30 /usr/libexec/sendmail/mail.local
-r-sr-xr-x    1 root     mail       317628 Jul 27 21:30 /usr/1ibexec/sendmail/sendmail
-rwsr-xr-x    1 root     majordom    8268 Jul 27 19:55 /usr/1ib/majordomo/wrapper
-rwsr-sr-x    1 root     root       91140 Jan 10 1995 /usr/1ib/svga/demos/3d
-rwsr-xr-x    1 root     root       14576 Jul 27 19:19 /bin/ping
-rwsr-xr-x    1 root     root       14000 Jul 27 20:32 /bin/su
-rwsr-xr-x    1 root     root       54112 Jul 27 20:27 /bin/mount
-rwsr-xr-x    1 root     root       27804 Jul 27 20:27 /bin/umount
-rwsr-sr-x    1 root     root        6644 Jul 27 23:57 /opt/kde/bin/kcheckpass
-rwsr-xr-x    1 root     root      371796 Jul 28 00:59 /opt/kde/bin/kppp
-r-sr-sr-x    1 root     tty        38044 Jul 27 19:24 /sbin/dump
-r-sr-sr-x    1 root     tty .     397068 Jul 27 19:24 /sbin/restore
-rwsr-xr-x    1 root     root       13959 Aug 27 10:55 /sbin/cardctl
```

Глядя на список, в принципе можно понять, почему для этих файлов пришлось установить атрибут SUID. Некоторым службам с ограничением доступа он нужен для того, чтобы их могли выполнять и непривилегированные пользователи. К таковым относится, например, команда изменения пароля. Другим программам, например mount или vmware, этот атрибут необходим, поскольку они обращаются к устройствам, и т. п.

Обратите внимание на группу программ, имена которых начинаются с символов uu. Владельцем этих программ является не root, а uucp. Эти программы обязательно нужно запускать от имени пользователя uucp, поскольку иначе они не будут работать. В наше время подавляющее число систем отказалось от использования UUCP в пользу более совершенной связи через Интернет, потому технология UUCP редко где используется, однако если в вашей системе до сих пор используется UUCP, вы должны принять во внимание дополнительные соображения, связанные с безопасностью. В частности, вам надо уделить внимание организации защиты при подключении через телефонную линию. Тем не менее общий подход к безопасности этой части системы такой же, как и для остальных частей, поэтому общие соображения, не относящиеся напрямую к UUCP, вполне применимы и для этой технологии.

В качестве меры предосторожности можно посоветовать время от времени выполнять приведенную ранее команду и сохранять выданный ею список SUID-файлов на гибком диске. Тогда, если вы заподозрите что-то неладное, можно будет сравнить изначальный список, хранящийся на гибком диске, с текущим перечнем файлов SUID (для этого удобно использовать утилиту diff). Благодаря этому вы сможете выяснить, появились ли в системе без вашего ведома новые SUID-фай-лы. Процедуру проверки SUID-файлов можно автоматизировать. Для этого создайте специальный сценарий и добавьте этот сценарий в набор ежедневных задач сгоп. Результаты проверки можно высылать на ваш электронный адрес по электронной почте. Разумеется существуют более удобные средства проверки, например программное средство tripwire, однако описанная процедура является достаточно простым и быстрым способом контроля SUID-файлов. Как и в других задачах, связанных с безопасностью, полезность результатов определяется тем, как вы их интерпретируете. Еще один способ проверки файлов подразумевает использование команды rpm -V, при условии, конечно, что база данных пакетов и программа rpm не были модифицированы злоумышленником.

Контроль над SUID/SGID файлами

Как вы уже, наверное, поняли, состояние файлов SUID/SGID вашей системы необходимо постоянно и тщательно контролировать. Безусловно, появление в системе новых файлов с этими атрибутами не должно оставаться незамеченным вами. То же самое можно сказать и про замену существующих файлов. Частота осуществления этих проверок зависит от степени риска, допустимой для вашей системы.

ПРИМЕЧАНИЕ

Помните, что нет ничего плохо в том, если пользователь даст атрибуты SUID/SGID своим файлам или каталогам. Беспокоиться нужно лишь тогда, когда в домашнем каталоге пользователя обнаружится SUID-файл, владельцем которого является root.

Однако это всего лишь две из трех необходимых мер безопасности, связанных с SUID/SGID. Не менее важно следить за обнаружением в используемых вами программах ошибок и уязвимых мест. Если вы

узнаете, что в одной из используемых вами программ обнаружена ошибка или уязвимое место, вы должны как можно быстрее заменить эту программу новой, исправленной версией или принять меры, позволяющие тем или иным образом обезопасить вашу систему от вторжения. Как правило, исправленные версии программ становятся доступными для использования уже через несколько часов после того, как становится известно о существовании ошибки или уязвимого места. Однако до появления исправленной версии бинарного файла компания-поставщик вашей операционной системы или такие агентства, как CERT (Computer Emergency Response Team), публикуют рекомендации о том, как обезопасить систему в отсутствие исправленного бинарного файла.

СОВЕТ

Компания Caldera, как и всякий другой поставщик комплекта Linux, поддерживает список рассылки, сообщающий об уязвимых местах и ошибках, обнаруженных в ее продуктах. Подписаться на него можно по адресу http://www.calderasystems.com.

Настройка атрибутов

Процедура установки атрибутов SUID/SGID и атрибута «sticky bit» (липкий бит) ничем не отличается от процедуры установки обычных атрибутов (о настройке обычных атрибутов рассказывалось в главе 3). Для этой цели используется все та же команда chmod. Соответствие между восьмеричными числами и атрибутами файла в данном случае следующее: число 4 соответствует атрибуту SUID, число 2 соответствует атрибуту SGID, наконец, число 1 соответствует атрибуту «sticky bit». При вызове chmod восьмеричное число, задающее желаемую комбинацию этих атрибутов, добавляется в качестве четвертого, самого старшего разряда восьмеричного числа, обозначающего набор разрешений на доступ к файлу (как рассказывалось в предыдущей главе, три других разряда обозначают комбинацию базовых разрешений на доступ к файлу со стороны владельца, группы и других пользователей).

Например, для того чтобы назначить файлу стандартную комбинацию разрешений на доступ, следует выполнить команду chmod 755 имя_файла. На самом деле эта команда является аналогом команды chmod 0755 имя_файла. Ноль в самом старшем восьмеричном разряде означает, что ни один из атрибутов SUID, SGID или «sticky bit» для файла не установлен. Любое число от 1 до 7 на месте этого нуля задает некую комбинацию только что перечисленных атрибутов. Так, для установки атрибута «sticky bit» для каталога /tmp, дабы изменять и удалять файлы в нем могли только их владельцы, необходимо выполнить:

```
chmod 1777 /tmp
```

Если вы хотите дать каталогу атрибут SGID (означающий, что в качестве владельца всех файлов, создаваемых в этом каталоге, будет назначаться группа, владеющая этим каталогом), необходимо выполнить команду chmod 2775 имя_ каталога. Аналогичным образом команда chmod 4755 имя_файла назначает файлу атрибут SUID. Чтобы назначить файлу оба атрибута, SUID и SGID (как сделано для программы си, относящейся к пакету UUCP), достаточно выполнить:

```
chmod 6755 cu
```

Здесь значение для SUID, равное 4, складывается со значением для SGID, равным 2, в результате получается значение 6, которое подставляется на четвертое знакоместо.

Проверить внесенные изменения можно с помощью команды ls -l Обратите внимание на комбинацию разрешений на доступ. Как мы уже знаем, набор базовых разрешений на доступ (чтение, запись, выполнение) обозначается латинскими буквами «rwx». Однако если для файла установлен атрибут SUID или SGID, вместо буквы «x» в этой тройке будет указана буква «s». При назначении атрибута SUID буква «s» займет место символа «x» в тройке прав владельца файла, а для атрибута SGID — в тройке прав группы файла. Атрибут «sticky bit» отображается несколько иначе: ему соответствует буква «t» на месте буквы «x» в тройке прав доступа для остальных пользователей.

Зная, как обозначаются эти атрибуты, несложно догадаться, как устанавливать их с использованием символьной маски в качестве аргумента команды chmod. Итак, чтобы назначить файлу атрибут SUID, оставив остальные атрибуты без изменений, нужно выполнить команду chmod u+s имя_файла. Для назначения атрибута SGID маску следует заменить на u+g. В обоих случаях минус вместо плюса сбрасывает указанный атрибут. Сброс или установка атрибута «sticky bit» (липкий бит) достигается посредством маски a-t или a+t, соответственно. Таблица 4.1 представляет собой расширенную версию таблицы 3.2 и содержит описание битов для дополнительных атрибутов файла.

Таблица 4.1. Права доступа в восьмеричном виде

Разрешения на доступ	Восьмеричный эквивалент	Дополнительные разрешения	Восьмеричный эквивалент	Комментарии
--------	000	- --X- -X- -X	0111	Число «О» в самом старшем разряде указывать необязательно, обязательными являются лишь три последующих восьмеричных числа
--X--X-X	111	--- --- --t	1001	«t» означает «sticky bit» (липкий бит)
-w--w--w-	222	--- --s ---	2010	«s» означает атрибуты SUID/SGID
-wx-wx-wx	333	--- --s --t	3011	Столбец «Дополнительные разрешения» показывает минимальный допустимый набор разрешений на доступ. Единицы в других разрядах стоят потому, что для назначения дополнительного атрибута необходимо, чтобы значение соответствующего разряда было ненулевым.
r--r--r--	444	--s --- ---	4100	
r-xr-x-r-x	555	--s --- --t	5101	
rw-rw-rw	666	--s --s --s	6110	
rwxrwxrwx	777	--s --s --t	7111	

Атрибуты файловой системы ext2

Теперь перейдем к рассмотрению атрибутов, которые поддерживаются в файловой системе ext2, в настоящее время используемой в качестве основной стандартной файловой системы Linux. В настоящее время эта файловая система активно используется в большинстве систем Linux, однако у нее есть существенный по нынешним временам недостаток — отсутствие поддержки журналирования (journaling). Возможно, что в ядре 2.4 (а скорей всего, в более поздних ядрах) вместо файловой системы ext2 будет использоваться файловая система SGI, однако в настоящее время система ext2 является распространенным стандартом.

Файловая система ext2 поддерживает еще восемь дополнительных атрибутов. По умолчанию при создании нового файла эти атрибуты не устанавливаются, поэтому если в этом есть необходимость, активизировать их нужно самостоятельно. Эти атрибуты хранятся в блоке информационного дескриптора (inode) под заголовком flag (флаг) в виде шестнадцатеричного числа.

- Атрибут «A», будучи установлен, отключает обновление поля времени последнего доступа к файлу (поле atime), что позволяет снизить нагрузку на жесткий диск. К сожалению, в настоящее время поддержка этого атрибута не реализована.

- Атрибут «S» предписывает операционной системе асинхронно сохранять на диске все модификации файла. Такое поведение совпадает с поведением MSDOS и эквивалентно выполнению команды sync после каждой модификации файла.

- Атрибут «a» запрещает выполнять над файлом какие-либо операции, кроме добавления данных. Это один из тех атрибутов, установить которые может только суперпользователь.

- Атрибут «c» включает сжатие файла. При записи файла на диск такой файл автоматически сжимается, а при чтении — разжимается.

- Атрибут «d» используется программой dump, которая игнорирует файлы с этим атрибутом и не осуществляет их резервное копирование.

- Атрибут «i» блокирует любую возможность изменения файла. Файлы, обладающие таким атрибутом, являются неизменяемыми (immutable). Файл, обладающий таким атрибутом, нельзя удалить, изменить или переименовать. Кроме того, на него нельзя сослаться при помощи ссылки. Все эти действия блокируются не только для обычных пользователей, но даже и для пользователя root. Только суперпользователь может установить этот атрибут.

- Атрибут «s» предписывает ядру сопровождать удаление файла записью нулей в дисковые блоки,

принадлежащие этому файлу. Такая мера безопасности делает чрезвычайно сложным, а в действительности фактически невозможным восстановление данных файла после удаления этого файла.

- Атрибут «U» предназначен для восстановления файла после того, как файл удален обычным образом. Однако не следует слишком полагаться на этот атрибут, так как зачастую восстановить удается лишь первый блок файла (обычно это 4096 байт). Если вы хотите обеспечить возможность восстановления удаляемых файлов, более правильным решением будет не удалять файл сразу, а перемещать его в специальный каталог, где он будет находиться некоторое время, по истечении которого cron удалит его по-настоящему.

Использование команды chattr

Для настройки перечисленных ранее атрибутов используется команда chattr. Как и для команды chmod, требуемая операция указывается при помощи оператора, за которым следуют символьные обозначения атрибутов. Оператор «+» предписывает команде установить указанные атрибуты, оператор «-» — сбросить их, а оператор «=» — установить указанные атрибуты и сбросить все остальные.

Кроме того, можно использовать параметр -R, означающий, что команда будет применена также рекурсивно в отношении всех подкаталогов, и параметр -V, означающий вывод версии программы chattr, а также дополнительных сообщений во время ее работы.

Поддерживается также параметр -v. При использовании этого параметра после него следует указать число, которое будет установлено в качестве номера версии индексного дескриптора. Никакого смысла, кроме того, который вы сами ему дадите, в номере версии нет — это просто число, которое можно записать в индексный дескриптор. Это число никак не связано с самим файлом. При создании файла ему выделяется индексный дескриптор из числа свободных, номер версии этого индексного дескриптора устанавливается равным единице. Если выполнить команду mv foo foo2, то файл foo2 унаследует свой индексный дескриптор от файла foo. Это означает, что если файл foo2 до этого существовал, его индексный дескриптор (а значит, и номер версии) будет заменен. Однако если при существующем foo2 выполнить cp foo foo2, индексный дескриптор foo2 останется прежним (если файл foo2 до этого не существовал, ему будет выделен свободный индексный дескриптор с номером версии, равным 1). Caldera использует эти номера при установке системы, давая всем устанавливаем файлам уникальный номер версии. Делается это для того, чтобы файлы, созданные после первоначальной установки, можно было отличить от файлов, созданных во время установки. Например, если обновить пакет (rpm -U), то номер версии новых файлов будет равен не заданному во время установки уникальному номеру, а простой единице. Далее мы еще вернемся к этому вопросу.

Использование команды lsattr

Команда lsattr выводит список файлов и их ext2-атрибутов. Подобно команде chattr, команда lsattr поддерживает параметр -R, при наличии которого она рекурсивно обрабатывает все подкаталоги, и параметр -V, при указании которого в первой строке вывода отображается информация о программе. Параметр -a означает отображение информации обо всех файлах, включая файлы, имена которых начинаются с точки. При использовании параметра -d выводятся сведения только о каталогах, но не о файлах. Параметр -l позволяет получить сведения о файлах в расширенном формате, где каждый атрибут отображается не с помощью буквы, а с помощью слов. Наконец, параметр -v включает вывод версии файлов. Будучи запущена без параметров, команда lsattr выдает примерно следующее:

```
------ ./conf.messages
-ucS-a-A ./uid
s---i-d- ./userbatch
```

Отсюда видно, что файл conf.messages является самым обычным файлом, то есть ни один из ext2-атрибутов у него не установлен. В то же время у файлов uid и userbatch, напротив, установлено сразу несколько атрибутов. Состояние одного атрибута представляется одним битом, поэтому для представления состояния всех ext2-атри-бутов достаточно восьмибитного числа, иначе говоря, любая комбинация ext2-атрибутов соответствует числу от 0 до 255, хранящемуся в индексном дескрипторе.

Теперь рассмотрим вывод команды lsattr при использовании параметра -v:

```
1883578780  --------  ./nisdomainname
1883578780  --------  ./domainname
1883580320  --------  ./ps
1883596959  --------  ./ex
1883596959  --------  ./rview
```

```
1883596959    --------        ./rvim
1883596959    --------        ./vi
1883596959    --------        ./view
1883596959    --------        ./vim
1883602714    --------        ./zsh
1883587488    --------        ./csh
1883587488    --------        ./tcsh
      1    --------        ./ksh
      1    --------        ./pdksh
```

Список тех же самых файлов, полученный при помощи команды Is -И:

```
20538    -r-xr-xr-x      2 root      root         6764    Jul 28 01:03 nisdomainname
20538    -r-xr-xr-x      2 root      root         6764     Jul 28 01:03 domainname
20539    -r-xr-xr-x      1 root      root        77428    Jul 27 19:31 ps
20564    rwxrwxrwx       1 root      root            3    Aug 5 08:59 ex -> vim
20565    lrwxrwxrwx      1 root      root            3    Aug 5 08:59 rview -> vim
20566    lrwxrwxrwx      1 root      root            3    Aug 5 08:59 rvim -> vim
20567    lrwxrwxrwx      1 root      root            3    Aug 5 08:59 vi -> vim
20568    lrwxrwxrwx      1 root      root            3    Aug 5 08:59 view -> vim
20569    -rwxr-xr-x      1 root      root       470696    Jul 27 22:59 vim
20571    -rwxr-xr-x      1 root      root       366344    Jul 27 21:03 zsh
20572    lrwxrwxrwx      1 root      root            4    Aug 509:02 csh -> tcsh
20554     -r-xr-xr-x     1 root      root       265516    Jul 27 20:21 tcsh
20573    -rwxr-xr-x      1 root      root       170768    Apr 30 12:14 ksh
20574    lrwxrwxrwx      1 root      root            3    Aug 5 23:52 pdksh -> ksh
```

В этом листинге видно, почему файлы nisdomainname и domainname имеют одинаковые номера версий: они ссылаются на один и тот же индексный дескриптор. Также можно заметить прямую связь между символическими ссылками и целевыми файлами, на которые эти ссылки указывают.

Файл pdksh и указывающая на него символическая ссылка ksh были добавлены уже после установки системы, поэтому номер их версии не является уникальным. Все файлы, добавленные в систему после ее установки, имеют номер версии, равный единице (если, конечно, вы впоследствии не изменили его).

ВНИМАНИЕ

Не следует излишне полагаться на уникальность номеров версий, поскольку при подмене файла злоумышленник может дать ему такой же номер, как и у оригинала. Однако если вдруг выяснилось, что файл /bin/login имеет номер версии, равный 1, то факт его подмены является несомненным.

Весьма полезным также является атрибут «i», который блокирует любые изменения файла как со стороны обычных пользователей, так и со стороны пользователя root. Рекомендуется устанавливать этот атрибут для частей файловой системы, которые либо вообще не меняются, либо меняются чрезвычайно редко. Благодаря этому суперпользователь не сможет по ошибке удалить их. Наиболее подходящими кандидатами на установку этого атрибута являются некоторые каталоги из /usr, каталоги /bin, /sbin, /lib и, быть может, некоторые другие.

Не следует устанавливать этот атрибут в отношении каталогов /var, /etc, /tmp, /home и прочих каталогов, содержимое которых постоянно изменяется.

Заключение

В этой главе был рассмотрено много тем начиная с атрибутов SUID/SGID и их значения для бинарных исполняемых файлов и каталогов и заканчивая некоторыми малоизвестными атрибутами файловой системы ext2. Кроме того, мы рассмотрели атрибут «sticky bit» (липкий бит) и его значение для каталогов. Мы узнали, как управлять этими битами и когда их следует или не следует устанавливать.

После этого мы перешли к рассмотрению команд chattr и lsattr, коснулись темы номеров версий файлов и закончили советом, как с помощью одного из ext2-атрибутов, а именно атрибута «i», уберечь файлы системы от случайного их удаления суперпользователем.

5 Структура файловой системы

В данной главе рассматриваются следующие вопросы:
- точки монтирования и файловые системы;
- параметры монтирования, поддерживаемые файловой системой ext2;
- параметры монтирования различных файловых систем.

В этой главе рассматривается структура файловой системы с точки зрения безопасности. Организуя структуру файловой системы, следует принимать во внимание множество соображений, таких как объем дискового пространства и прочее, однако помимо этого не следует забывать также и о безопасности. В этой главе мы рассмотрим некоторые аспекты защиты файловой системы.

Точки монтирования

Точкой монтирования (mount point) называется каталог, расположенный в некотором разделе, через который делается доступным содержимое другого раздела. Тем самым разрозненные физические разделы объединяются в единое логическое дерево каталогов. Иначе говоря, при монтировании раздела его содержимое добавляется к этому дереву в виде новой ветви, берущей начало в точке монтирования. Тип и физическое расположение монтируемого раздела могут быть самыми разными: монтируемый раздел может обладать форматом ext2 и располагаться на локальном жестком диске, однако это может быть также раздел NFS (Network File System), расположенный на удаленном компьютере, доступ к которому осуществляется через сеть. Чтобы выполнить монтирование, необходимо, чтобы монтируемый раздел был доступен (если связь с удаленным узлом отсутствует, раздел NFS смонтировать не удастся) и чтобы его формат поддерживался операционной системой Linux.

На момент написания этой книги наиболее часто используемой файловой системой в OpenLinux является система ext2. Это основной стандартный тип файловой системы Linux. Об особенностях этой файловой системы рассказывалась в главах 3 и 4, где, в частности, говорилось, что, в отличие от DOS, в ext2 используются индексные дескрипторы, в которых хранится вся информация о файле. На самом деле в разных операционных системах используются разные типы файловых систем, а в разных файловых системах используются разные подходы к хранению файлов. В настоящее время существует огромное количество разнообразных файловых систем, их даже больше, чем операционных систем. Система Linux позволяет работать с очень многими файловыми системами.

Точка монтирования обязательно должна быть каталогом, однако не требуется, чтобы этот каталог был пустым. На самом деле получить полностью пустой каталог обычными средствами вам не удастся, поскольку любой каталог всегда содержит файлы «.» и «..», но речь здесь не об этом. Монтирование раздела приводит к тому, что содержимое каталога, являющегося точкой монтирования, заменяется содержимым раздела. В результате изначальное содержимое каталога становится недоступным до тех пор, пока монтированный раздел не будет размонтирован. Таким образом, если у кого-то получится смонтировать гибкий диск поверх каталогов /bin или /sbin, то вместо оригинальных файлов будут использоваться файлы с гибкого диска со всеми вытекающими из этого последствиями.

Однако всякий, кто пытался монтировать раздел, не будучи суперпользователем, знает, что любые операции монтирования строго контролируются операционной системой. Информация о том, какие разделы и в какие каталоги дозволено монтировать обычным пользователям, содержится в файле /etc/fstab. Вот как может выглядеть содержимое этого файла:

/dev/hdal	/	ext2	defaults	1	1
/dev/hda	/home	ext2	defaults	1	2
/dev/hda4	swap	swap	defaults	0	0
devpts	/dev/pts	devpts	gid=5,mode=620	0	0
/proc	/proc	proc	defaults	0	0
/dev/fdO	/mnt/floppy	msdos	defaults,users,noauto	0	0
/dev/hdc	/mnt/cdrom	iso9660	ro.user,noauto	0	0

Файл fstab состоит из шести столбцов, разделенных символами пробелов. В первом столбце указывается устройство, обычно дисковое, то есть sd*, fd* или hd*. Одно из возможных исключений показано выше: это устройство devpts, соответствующее псевдотерминалу (pty). Ядро, устанавливаемое при установке системы,

скомпилировано с поддержкой 256 таких терминалов, за что отвечают параметры конфигурации ядра CONFIG_UNIX98_PTYS и CONFIG_UNIX98_PTY_COUNT. Во втором столбце указываются точки монтирования, куда будут монтироваться устройства из первого столбца. Если понаблюдать за содержимым каталога /dev/ pts, то можно заметить, что оно меняется в зависимости от запущенных программ. Файлы из этого каталога соответствуют задействованным на данный момент псевдотерминалам. Открывают их, например, такие программы, как telnet или xterm. Устройства pty не используются виртуальными консолями, вместо этого виртуальные консоли используют tty.

Для псевдотерминалов приняты специальные меры безопасности: обратите внимание на режим 620 в четвертом столбце, означающий права на чтение-запись для владельца, на запись для группы и отсутствие прав для остальных пользователей. Теперь посмотрим на содержимое /dev/pts:

```
drwxr-xr-x    2 root    root 0        Sep 13 07:06  .
drwxr-xr-x    4 root    root 11264    Sep 13 08:52  ..
crw--w----    1 david     tty 136,  0 Sep 13 09:18  0
CPW--W----    1 david     tty 136,  1 Sep 13 09:11  1
crw-------    1 david     tty 136.  2 Sep 13 09:13  2
```

Как видите, владельцем любого псевдотерминала является открывший его пользователь, кроме того, все они принадлежат группе tty. Владелец имеет право на чтение-запись во все открытые им псевдотерминалы, группа же имеет право только на запись и только в отношении устройств pty0 и pty1. Устройство pty2 доступно только для чтения. Теперь посмотрим на режимы команд write и wall:

```
-rwxr-sr-x  I root   tty      8356 Jul 27 20:27 /usr/bin/write
-rwxr-sr-x  1 root   tty      6748 Jul 27 19:22 /usr/bin/wall
```

Для этих команд установлен атрибут SGID и потому они выполняются от имени группы tty — той самой группы, которая обладает всеми псевдотерминалами. Таким образом, при помощи этих команд можно будет вывести сообщение на первые два псевдотерминала, но не на последний. Важно, чтобы только владелец имел право на чтение псевдотерминалов, иначе появляется возможность «прослушивания» их с целью перехвата пароля.

Помните команду mesg? Если выполнить ее на всех псевдотерминалах, открытых пользователем david, то для двух из них будут выведено y, а для одного — n. Это означает, что на первые два псевдотерминала сообщения выводиться будут, а на последний — нет. Для включения возможности вывода сообщений на данный псевдотерминал нужно выполнить на нем команду mesg y. При этом группе tty будет предоставлено разрешение на запись в этот псевдотерминал. И наоборот, выполнение команды mesg n отнимает это право, тем самым отключая возможность вывода сообщений на данный псевдотерминал.

Продолжим рассмотрение файлов из /dev/pts. За именем группы, которую имеет файл терминала, следуют старший (major) и младший (minor) номера устройства. В данном случае старший номер для всех файлов одинаков и равен 136, а младший принимает значения 0, 1 и 2. Дата последнего доступа к файлу, следующая за ними, в данном контексте означает дату, когда пользователь последний раз записывал данные в псевдотерминал. Соответственно, после каждого вывода команды через псевдотерминал эта дата меняется.

ПРИМЕЧАНИЕ

Содержимое /dev/pts периодически следует проверять. Команда ps aux | grep pts позволит вам узнать, какие процессы связаны с открытыми на данный момент псевдотерминалами, а команда w или who — кем были открыты эти псевдотерминалы. Подозрительными можно считать псевдотерминалы со старой датой последнего доступа, псевдотерминалы, с которыми не связан ни один активный процесс, и псевдотерминалы, используемые процессами, о которых вы не имеете ни малейшего представления. Возникновение любой из этих ситуаций может означать, что защита вашей системы была нарушена.

В третьем столбце файла /etc/fstab указывается тип файловой системы, монтируемой по умолчанию. Это может быть любой тип из числа тех, что известны ядру, например, nfs, vfat или одно из значений swap, devpts или ignore.

Четвертый столбец содержит параметры. Имеются в виду параметры файла fstab, а не параметры командой строки. Параметров командой строки чуть больше и они несколько отличаются от параметров, указываемых после ключа -o. Например, параметр «только для чтения» в fstab обозначается как ro, в командной же строке это свойство назначается при помощи ключа -o ro или просто -r. В этом тексте рассматриваются только те параметры командной строки, для которых существует эквивалент в файле fstab.

Любой пользователь с нулевым идентификатором UID может монтировать любую файловую систему в любой каталог. Как упоминалось ранее, монтируемая файловая система замещает прежнее содержимое каталога, что в некоторых случаях может оказаться полезным. Например, представьте, что при установке пакета обслуживающая этот пакет библиотека была помещена в каталог /usr/lib. Однако в этом случае данный пакет можно использовать только тогда, когда раздел /usr монтирован в каталоге /usr. Однако в процессе начальной загрузки монтирование раздела /usr происходит не сразу — система вначале выполняет

множество инициализационных процедур и лишь после этого осуществляет монтирование дополнительных разделов. Таким образом, во время начальной загрузки ОС до того момента, пока не будет смонтирован раздел /usr, ни одна программа не сможет воспользоваться рассматриваемым пакетом. Проблему можно решить двумя способами. Во-первых, можно переместить библиотеку в другой каталог. Во-вторых, можно создать в корневом разделе каталог /usr/lib и поместить в него копию необходимой библиотеки. Оба решения вполне приемлемы. Однако для выполнения стандартной процедуры обновления пакета с использованием RPM будет лучше, если все содержащиеся в этом пакете файлы останутся на предназначенных им местах, поэтому правильней будет пойти по второму пути. К сожалению, дисковое пространство, занимаемое библиотекой на корневом разделе, после монтирования раздела /usr будет пропадать зря, однако файлы библиотек, как правило, не занимают много места (несколько десятков килобайтов по сегодняшним меркам совсем немного), поэтому с этим неудобством вполне можно смириться.

С точки зрения безопасности, монтировать файловые системы, не упомянутые в файле /etc/fstab, может только суперпользователь. Обычному же пользователю дозволено монтировать только их, да и то не все, а лишь те системы, для которых указан параметр user или users.

Здесь будут обсуждаться лишь те параметры монтирования, которые имеют отношение к безопасности. Итак, суперпользователь может монтировать все что угодно. На то он и суперпользователь. Однако остальные пользователи являются посторонними для системы, и потому весьма опасно давать им такую возможность. Стало быть, чтобы грамотно ограничивать возможности пользователей по монтированию файловых систем, нужно знать, как именно работает механизм монтирования.

ВНИМАНИЕ

Если вы предоставляете пользователем возможность монтирования файловых систем, вы подвергаете вашу систему определенному риску. Следует ясно представлять себе, с чем связан этот риск, и исходя из этого использовать соответствующие параметры в файле /etc/fstab.

В рассматриваемом нами случае пользователям предоставлена возможность монтирования компакт-диска и гибкого диска. Вернитесь еще раз к примеру файла /etc/fstab, который был приведен чуть ранее в этой главе, и посмотрите на строчки, соответствующие этим двум устройствам. Обратите внимание на то, что в одном случае использован параметр user, а в другом — users. Монтирование файловых систем осуществляется командой mount. Будучи вызвана без параметров, эта команда читает файл mtab и выводит список смонтированных в системе разделов. Вот как может выглядеть ее вывод:

```
/dev/hdal on / type ext2 (rw)
/dev/hda3 on /home type ext2 (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/proc on /proc type proc (rw)
/dev/hdc on /mnt/cdrom type iso9660 (ro,noexec,nosuid,nodev,user=david)
/dev/fdO on /mnt/floppy type msdos (rw.noexec.nosuid.nodev)
```

А вот что находилось в файле /etc/mtab на момент этого вывода:

```
/dev/hdal / ext2 rw 0 0
/dev/hda3 /home ext2 rw 0 0
devpts /dev/pts devpts rw,g1d=5,mode=620 0 0
/proc /proc proc rw 0 0
/dev/hdc /mnt/cdrom iso9660 ro,noexec,nosuid,nodev,user=david 0 0
/dev/fdO /mnt/floppy msdos rw.noexec.nosuid.nodev 0 0
```

ПРИМЕЧАНИЕ

Смонтированные файловые системы перечислены также и в файле /proc/mounts, содержимое которого в точности совпадает с содержимым /etc/mtab. Однако было бы неправильно делать /etc/mtab ссылкой на /proc/mounts, поскольку это может привести к проблемам с устройствами loopback.

Эти два листинга несколько различаются. В файле mtab перечисляются все смонтированные файловые системы (на самом деле те, при монтировании которых не был указан ключ -n), включая соответствующие столбцы из /etc/fstab, а кроме того, в нем присутствуют два столбца, которых нет в выводе команды mount Помимо этого, в выводе команды mount параметры берутся в скобки, поскольку так их легче заметить.

ПРИМЕЧАНИЕ

Если смонтировать файловую систему с ключом -n, означающим «не помещать запись о монтируемой системе в файл mtab», то ни команда df, ни команда mount не покажут эту файловую систему. Тем не менее, если используется /proc, то сведения об этой файловой системе будут присутствовать в файле /proc/mounts. Ключ -n используется в случае, если файловая система монтируется только для чтения и не предназначена для обычного использования.

В обоих листингах есть записи, соответствующие гибкому диску и компакт-диску. В записи для компакт-диска есть строчка user=david. В записи для гибкого диска такой строчки нет. Различие вызвано тем, что для гибкого диска в файле / etc/fstab используется параметр users, а для компакт-диска — параметр user. Параметр user означает, что монтировать данную файловую систему в данную точку монтирования может любой пользователь, а размонтировать — только тот, кто ранее смонтировал ее. Параметр users означает, что монтировать и размонтировать файловую систему может любой пользователь. В этом примере я намеренно отошел от рекомендуемого использования параметров user и users и переставил их местами. Как видно из листинга, компакт-диск смонтирован в режиме только для чтения (ro), тогда как гибкий диск — в режиме чтения-записи (rw). При таком режиме другой пользователь может подменить чужую дискету своею, дождаться окончания записи на нее, вставить обратно оригинальную дискету и спокойно удалиться с чужими файлами на своей дискете. Поэтому подумайте, стоит ли в вашем случае разрешать пользователям монтировать гибкие диски на запись. Для компакт-диска, смонтированного в режиме только для чтения, такой проблемы не существует, поскольку файлы на него никто записывать не будет. Что касается записываемых компакт-дисков (CD-R), то в большинстве устройств чтения и тем более записи компакт-дисков лоток для диска не выйдет до тех пор, пока диск не будет размонтирован, поэтому с компакт-дисками все гораздо проще.

В файле fstab используются следующие параметры (порядок перечисления значения не имеет):

- async/sync — обмен данными с данной файловой системой осуществляется асинхронно/синхронно;

- atime/noatime — обновлять/не обновлять дату последнего доступа (atime);

- auto/noauto — при использовании параметра -a совместно с командой mount данная файловая система будет/не будет смонтирована;

- dev/nodev — интерпретировать/не интерпретировать символьные или специальные блочные устройства;

- exec/noexec — разрешить/запретить выполнение бинарных файлов, расположенных в данной файловой системе; •

- suid/nosuid — принимать во внимание/игнорировать атрибуты SUID/SGID для данной файловой системы;

- nouser/user(s) — разрешить/запретить пользователям монтировать данную систему;

- rw/ro — монтировать систему для чтения-записи/только для чтения;

- remount — если возможно, данную файловую систему следует монтировать заново (обычно используется для изменения режима доступа к корневой файловой системе с го (только для чтения) на rw (чтение-запись) и наоборот);

- defaults — использовать параметры по умолчанию, что означает rw, suid, dev, exec, auto, nouser и async

В отношении параметра defaults могут возникнуть некоторые вопросы. Если использован параметр user(s), а значения остальных параметров равны значениям по умолчанию, то есть suid, dev, exec, то что мешает обычному пользователю смонтировать гибкий диск, содержащий исполняемые файлы с атрибутом SUID (это могут быть программные средства для взлома системы)? Все очень просто. При наличии параметра user(s) значения параметров по умолчанию меняются на noexec, nosuid и nodev. Если по некоторым причинам требуется избавить пользователей от необходимости копировать файл с дискеты для его выполнения, можно явно указать параметр exec. Только нужно, чтобы он следовал после параметра defaults. Команда mount обрабатывает параметры слева направо, замещая ранее встреченные значения параметров значениями, указанными далее. Поэтому указание user перед defaults равносильно указанию nouser вслед за user: в итоге используется значение параметра, указанное последним.

Пятый столбец файла /etc/fstab является управляющим столбцом команды dump. Единица в этой позиции предписывает осуществлять резервное копирование данной файловой системы, ноль — не осуществлять. Шестой столбец управляет порядком проверки файловых систем командой fsck. Для корневой файловой системы следует указать 1, для всех остальных — 2, а для тех систем, которые не нужно проверять, — 0 (некоторые файловые системы просто бессмысленное проверять командой fsck, к этой категории относятся фактически все файловые системы, не являющиеся ext2). Пустое значение в этих двух столбцах интерпретируется как ноль, однако заметим, что если шестой столбец не пуст, то и пятый столбец обязан содержать непустое значение.

Дополнительные параметры различных файловых систем

Рассмотренные ранее параметры применимы ко всем файловым системам. Но кроме них у некоторых

файловых систем есть свой собственный набор параметров. Это связано с тем, что в разных файловых системах с файлами ассоциируется служебная информация разного характера. Иногда эта информация достаточно емкая, а иногда она весьма скудна.

Количество файловых систем, поддерживаемых в Linux, достаточно велико и с каждой новой версией ядра оно увеличивается. Здесь будут рассмотрены некоторые наиболее распространенные файловые системы, для остальных же вам придется заглянуть в соответствующую документацию. Больше всего параметров поддерживается в отношении файловых систем, используемых в однопользовательских операционных системах, таких как DOS и Macintosh. Из всех этих параметров будут рассмотрены лишь те, которые имеют непосредственное отношение к безопасности. Некоторые файловые системы обсуждаться здесь не будут, но не потому, что они не слишком распространены или же у них нет специфичных для них параметров, а потому, что среди этих параметров нет интересных с точки зрения безопасности. Это замечание относится к следующим файловым системам: coherent, ext (более не используется), minix, ncp, nfs, romfs, smbfs, sysv, ufs,

xenix и xiafs (использование не рекомендуется).

ССЫЛКА

Файловая система nfs рассматривается в главе 12.

Amiga affs

Среди параметров данной файловой системы есть несколько параметров, связанных с безопасностью. Как и обычные параметры, параметры безопасности могут быть указаны в командной строке (при этом пользователь, выполняющий монтирование, должен обладать идентификатором UID=0) или же в файле /etc/fstab. Начнем с параметров uid= и gid=. В соответствии со своим названием, эти параметры устанавливают указанные числа в качестве значений идентификаторов пользователя и группы для каталога, являющегося точкой монтирования файловой системы. Если эти параметры не указаны и отсутствуют в файле fstab, то для affs по умолчанию устанавливаются нулевые значения (uid=gid=0). Однако если в файле fstab присутствуют ключевые слова uid= и/или gid= без указания каких-либо числовых значений, то в качестве соответствующего идентификатора для корня файловой системы используется идентификатор UID/GID процесса (пользователя), выполняющего монтирование. Обычно это именно то, что нужно, но вы должны убедиться в том, что данный режим монтирования вам подходит.

Кроме параметров uid= и gid= есть еще параметры setuid= и setgid=, которые устанавливают владельца и/или группу для всех файлов файловой системы. Различие между параметрами setuid= и setgid= и параметрами uid= и gid= состоит в том, что первые определяют разрешения на доступ к файлам в рамках файловой системы, в то время как вторые определяют владельца всей файловой системы.

Следующие два параметра, mode= и protect, являются взаимодополняющими, хотя их можно использовать и по отдельности. Параметр mode= устанавливает режим доступа для всех файлов и каталогов файловой системы, независимо от того, какие разрешения были назначены в отношении этих файлов первоначально. Параметр protect делает невозможным изменение пользователями, в том числе и владельцами, прав доступа к файлам системы. Данные параметры можно использовать в случае, если для монтируемой системы вы хотите запретить запуск исполняемых файлов и при этом намерены блокировать возможность управления атрибутом, разрешающим выполнение программ. В этом отношении обратите внимание также на параметр noexec, о котором упоминалось ранее.

Последним параметром, заслуживающим упоминания в данном тексте (остальные параметры относятся скорее не к безопасности, а к администрированию), является параметр usemp. Этот параметр можно использовать и в отношении других файловых систем, хотя он и является несколько странным. Этот параметр предписывает при монтировании использовать значения UID и GID точки монтирования, а при размонтировании сбрасывать этот параметр.

Linux ext2

Первыми рассмотрим параметр grpid (также известный как bsdgroups) и его антоним nogrpid (также известный как sysvgroups). По умолчанию используется nogrpid. Использование параметра grpid дает тот же эффект, что и установка бита SGID на каталог, но в масштабе всей файловой системы: при создании файла ему назначается группа каталога, а не группа инициировавшего создание пользователя. При наличии параметра nogrpid все происходит наоборот: при создании файла ему назначается группа пользователя, как это принято в версиях Unix ветви System V.

ПРИМЕЧАНИЕ -

BSD и SysV по-разному трактуют идентификаторы группы для каталогов. Что касается Linux, то в этой ОС сделана попытка объединения всего лучшего из обеих ветвей, и в данном случае выиграл подход SysV. В других

случаях, таких как утилиты командной строки, предпочтение отдано подходу BSD, поэтому все утверждения о принадлежности Linux к той или иной ветви верны лишь отчасти.

Далее перейдем к параметрам resgid= и resuid=. В ext2 некоторый процент свободного дискового пространства файловой системы является зарезервированным. Размер этого пространства указывается при создании файловой системы (по умолчанию резервируется 5 % от размера раздела), в дальнейшем же это число можно изменить с помощью программы tune2fs. Параметры resgid= и resuid= задают идентификаторы группы и пользователя, которым разрешается использовать это зарезервированное пространство.

ВНИМАНИЕ

Как правило, пространство на файловой системе резервируется для суперпользователя. Система, на которой полностью окончилось свободное пространство, недоступна для использования всеми пользователями до тех пор, пока не будут приняты специальные меры по ее восстановлению (удалению файлов). Зачастую для этого требуется применять такие средства, как debugfs, однако доводить систему до такого положения дел не рекомендуется.

FAT, MSDOS, UMSDOS, VFAT

Сама по себе FAT не является файловой системой — это общая часть таких файловых систем, как MSDOS, UMSDOS и VFAT. Но поскольку это не просто общая, но неотъемлемая часть этих файловых систем, они будут рассмотрены все вместе, с указанием особенностей каждой отдельной системы.

Параметры uid= и gid= определяют значения идентификаторов UID/GID для всех файлов файловой системы. По умолчанию используются идентификаторы процесса, инициировавшего монтирование. Все файловые системы семейства FAT получают идентификаторы смонтировавшего их пользователя. Если это не то, что вам нужно, вы должны явно указать необходимые вам идентификаторы в файле /etc/fstab.

Следующий параметр это umask=. По умолчанию используется значение umask для текущего процесса (обычно это оболочка пользователя).

Теперь о параметре check=. Он может принимать три значения: r (relaxed), что означает ослабленная проверка, n (normal) — нормальная, и s (strict) — строгая. По умолчанию осуществляется нормальная проверка. С точки зрения безопасности здесь важно то, что нормальная проверка не отклоняет имена файлов, содержащие символы, допустимые в Linux, но не допустимые в MSDOS, такие как +, =, пробелы и т. п. Чтобы блокировать использование таких символов, следует использовать строгую проверку. Поэтому если предполагается, что между системой Microsoft и системой linux будет осуществляться обмен файлами, то , возможно, имеет смысл использовать строгую проверку.

Как известно, между текстовыми файлами DOS и Unix есть одно существенное различие: в DOS строка заканчивается символами возврата каретки и перевода строки, тогда как в Unix — просто символом перевода строки. Для управления преобразованием файлов служит параметр conv=. По умолчанию никакого преобразования не производится, а вообще этот параметр может принимать три значения: b (binary), t (text) и a (auto). Значение t предписывает осуществлять трансляцию между символами конца строки для всех файлов. Значение a — для всех файлов, за исключением тех, которые система считает бинарными. В данном случае система считает бинарными файлы, обладающие расширениями, перечисленными в файле /usr/src/linux/fs/fat/misc.c. Значение b предписывает системе вообще не осуществлять трансляцию.

ВНИМАНИЕ

Автоматическое преобразование может повредить файлы настолько, что их уже невозможно будет восстановить. Поэтому настоятельно рекомендуется полностью отключить преобразование и для решения проблемы пользоваться такими программами, как unix2dos/dos2unix или fromdos/todos (или же возможностями по преобразованию, встроенными в редактор vi).

Далее перейдем к параметру fat=, устанавливающему тип файловой системы. Возможные значения: 12 и 16. По умолчанию тип системы определяется автоматически, но с помощью параметра fat= его можно переопределить. Однако к использованию этого параметра следует подходить осторожно, поскольку при неправильном значении этого параметра любая запись в файловую систему может повредить содержащиеся в ней данные.

Кроме того, у семейства FAT есть такие параметры, которые лучше не использовать. Все они представляют собой грубые попытки внедрения в FAT соглашений Unix или DOS. К таким параметрам относятся: sys_immutable, showexec, dots, nodots, doysOK=[yes|no].

У VFAT есть дополнительный параметр — posix. Он позволяет использовать имена файлов, различающиеся только регистром символов.

OS/2 HPFS

Имеющие отношение к безопасности параметры HPFS представляют собой подмножество соответствующих параметров для FAT. А именно: uid=, gid=, umask=, conv=. Смысл их точно такой же как и для FAT. Отходить от значения по умолчанию b для параметра conv= крайне не рекомендуется.

CD-ROM ISO9660

В файловой системе обычного компакт-диска (без расширений, ISO9660) применяется стандартное соглашение об именах DOS в формате 8.3, и все имена используют верхний регистр символов. Никакой информации о владельцах, правах доступа и т. п. с файлами не ассоциируется. Такие компакт-диски сегодня редко где встретишь. В настоящее время в большинстве компакт-дисков используются расширения ISO9660: Rock Ridge или Joliet. При этом только расширение Rock Ridge применимо к ISO9660. Идея этого расширения заключается в использовании специальных файлов для отображения коротких имен в длинные и хранения информации о владельцах и правах доступа к файлам. Поддержка Rock Ridge по умолчанию включена, однако есть параметр, позволяющий ее выключить. Так как компакт-диски всегда монтируются только для чтения, большинство параметров не столь важны для безопасности, как в случае с файловыми системами, монтируемыми и для записи.

По умолчанию файловая система ISO9660 монтируется с использованием значений uid=0 и gid=0. В этом она отличается от таких файловых систем, как AFFS, FAT и HPFS, которые по умолчанию монтируются с использованием идентификаторов UID/GID, позаимствованных у процесса, выполняющего монтирование. Если такое положение дел вас не устраивает, его можно переопределить при помощи параметров uid= и gid=.

Кроме того, можно также использовать параметры mode= и conv=, обсуждавшиеся ранее.

Параметр unhide разрешает отображение скрытых и ассоциированных файлов.

PROC

Файловая система /proc на самом деле вовсе не является настоящей файловой системой. Тема эта настолько велика, что она будет отдельно рассмотрена в следующей главе. В отношении файловой системы proc можно использовать только два параметра: uid= и gid=, однако на момент написания данной книги эти параметры никак не влияли на стандартное поведение системы proc (по крайней мере, на компьютере автора с использованием ядра версии 2.2.12). Даже если в вашей системе эти параметры действуют, не существует никакой разумной причины, по которой ими следует воспользоваться.

СОВЕТ

Хотя значения по умолчанию в файле /etc/fstab можно и не указывать, это все-таки полезно сделать, поскольку благодаря этому вы сможете избежать проблем с безопасностью в случае, если в будущем значения по умолчанию по той или иной причине будут изменены.

Заключение

В этой главе был рассмотрен процесс монтирования файловых систем. Было рассказано про точки монтирования и параметры монтирования, имеющие отношение к безопасности. Я рассказал вам о том, какому риску вы подвергаете свою систему, разрешая обычным пользователям выполнять монтирование файловых систем. Все же, несмотря на определенный риск, без возможности монтирования обычным пользователям подчас не обойтись, однако прежде чем предоставлять им такую возможность, вы должны четко представлять себе возможные последствия.

Только суперпользователь может монтировать все, что захочет, туда, куда захочет. Остальным пользователям разрешается монтировать только файловые системы, указанные в файле /etc/fstab, при этом монтирование будет выполнено с учетом указанных в этом файле параметров. Таким образом, чтобы обеспечить безопасность монтирования, необходимо грамотно настроить записи этого файла.

6
Файловая система /proc

В данной главе рассматриваются следующие вопросы: - файловая система /proc;
- каталог /proc/sys;
- файловая система /dev/pts;
- соображения безопасности для /proc.

Файловая система /proc не похожа на все остальные файловые системы. За исключением /dev/pts (псевдотерминальные устройства или попросту псевдотерминалы, о которых рассказывается далее), все остальные файловые системы представляют собой ровно то, что предполагает сам термин «файловая система»: некоторым образом структурированный набор (система) файлов на некотором носителе. Все эти файлы существуют физически, даже псевдотерминалы, однако файловая система /proc существует лишь во время работы операционной системы. И хотя псевдотерминалы, как и /proc, прекращают свое существование при нормальном завершении работы, однако при грубом выключении питания соответствующие им файлы все-таки остаются на жестком диске (конечно же, в процессе следующего запуска системы эти файлы будут удалены с диска). В отличие от псевдотерминалов файловая система /proc существует в оперативной памяти и больше нигде. Иначе говоря, файловая система /proc — это иллюзия файловой системы, которая формируется и поддерживается операционной системой.

Файловая система /proc

Файловую систему /proc можно сравнить с окном, при помощи которого процессы и пользователи могут наблюдать за состоянием операционной системы. По большей части, именно в этом и состоит ее назначение — предоставлять информацию о различных частях системы. И хотя данная глава не слишком большая, но концепции, затрагиваемые в ней, весьма важны, поскольку имеют непосредственное отношение к основам Linux.

При желании можно скомпоновать ядро, в котором не будет поддержки /proc, но в этом нет особого смысла. Очень многие программы и утилиты в процессе своего функционирования обращаются к файловой системе/proc для получения необходимой информации о состоянии системы. Если механизм /proc отсутствует в системе, все использующие его программы и утилиты либо не смогут работать, либо будут вынуждены получать необходимую информацию из других источников. Однако другие источники информации зачастую менее надежны и предлагают информацию в меньшем объеме. Чтобы убедиться в том, что поддержка /proc добавлена в ядро, в процессе конфигурирования ядра следует перейти в меню File Systems и выбрать поддержку файловой системы /proc (по умолчанию поддержка файловой системы /proc включена). В файле .config поддержка /proc включается благодаря наличию строки:

```
CONFIG_PROC_FS=y
```

Использовать поддержку /proc в виде модуля нельзя.

Если поддержка /proc в ядре уже имеется, то система готова использовать ее, однако, как и для других механизмов Linux, эту файловую систему можно включать и отключать. Активация и деактивация файловой системы /proc осуществляется при помощи файла /etc/fstab. Поскольку /proc является хотя и фиктивной, но файловой системой, то прежде чем ее можно будет использовать, она, как и любая другая файловая система, должна быть смонтирована. Для успешного монтирования любой файловой системы необходимо, чтобы существовал каталог, в который она монтируется, и /proc здесь не исключение. Если удалить каталог /proc, то монтировать ее будет некуда, и потому вместо /proc вы получите сообщение об ошибке. Соответствующая запись файла /etc/fstab выглядит следующим образом: /proc /proc proc defaults O

О

На месте пути к файлу устройства здесь стоит /proc, поскольку эта файловая система существует внутри ядра, а не на устройстве. Монтируется она в каталог / proc, с типом proc. В ее отношении никогда не выполняется каких-либо проверок, и она не попадает под действие команды dump. Действительно, большинство файлов в /proc отображают сиюминутное состояние операционной системы и, как следствие, доступны только для чтения, поэтому их бессмысленно проверять или сохранять для дальнейшего восстановления. Исключение составляют файлы из /proc/sys, позволяющие изменять состояние системы и потому доступные и для записи, но восстановление этих файлов из резервной копии почти наверняка приведет систему в нестабильное состояние, так что резервное копирование им не нужно.

Файловая система /proc очень динамична. Некоторые из ее частей меняются фактически постоянно, интенсивность изменений зависит от нагрузки на систему. Поэтому любые два снимка ее содержимого, скорей всего, будут различаться. На листинге 6.1. показан один из снимков содержимого /proc.

Листинг 6.1. Содержимое */proc*

```
dr-xr-xr-x    58 root     root     0 Nov 15 19:01 .
drwxr-xr-x    21 root     root     1024 Nov 15 12:02 ..
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1076
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1079
dr-xr-xr-x    3 dns       dns            0 Nov 16 07:46 1088
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1092
dr-xr-xr-x    3 bin       root     0 Nov 16 07:46 1094
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1176
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1181
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1190
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1202
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1263
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1270
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1271
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1284
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1286
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1287
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1288
dr-xr-xr-x    3 nobody    nobody   0 Nov 16 07:46 1317
dr-xr-xr-x    3 nobody    nobody   0 Nov 16 07:46 1318
dr-xr-xr-x    3 nobody    nobody   0 Nov 16 07:46 1319
dr-xr-xr-x    3 nobody    nobody   0 Nov 16 07:46 1320
dr-xr-xr-x    3 nobody    nobody   0 Nov 16 07:46 1321
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1363
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1371
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1372
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1373
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1374
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1375
dr-xr-xr-x    3 nobody    nobody   0 Nov 16 07:46 1618
dr-xr-xr-x    3 nobody    nobody   0 Nov 16 07:46 1619
dr-xr-xr-x    3 nobody    nobody   0 Nov 16 07:46 1620
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1741
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 1800
dr-xr-xr-x    3 david     david    0 Nov 16 07:46 1892
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 2
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 3
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 4
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 4856
dr-xr-xr-x    3 david     david    0 Nov 16 07:46 4858
dr-xr-xr-x    3 david     david    0 Nov 16 07:46 4914
dr-xr-xr-x    3 david     david    0 Nov 16 07:46 4915
dr-xr-xr-x    3 david     david    0 Nov 16 07:46 4917
dr-xr-xr-x    3 david     david    0 Nov 16 07:46 4963
dr-xr-xr-x    3 david     david    0 Nov 16 07:46 4964
dr-xr-xr-x    3 david     david    0 Nov 16 07:46 4965
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 4967
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 4968
dr-xr-xr-x    3 root      root     0 Nov 16 07:46 5
```

```
dr-xr-xr-x    3 root     root        0 Nov 16 07:46 5038
-r--r--r--    1 root     root        0 Nov 16 07:46 apm
dr-xr-xr-x    4 root     root        0 Nov 16 07:46 bus
-r--r--r--    1 root     root        0 Nov 16 07:46 cmdline
-r--r--r--    1 root     root        0 Nov 16 07:46 cpuinfo
-r--r--r--    1 root     root        0 Nov 16 07:46 devices
-r--r--r--    1 root     root        0 Nov 16 07:46 dma
-r--r--r--    1 root     root        0 Nov 16 07:46 fb
-r--r--r--    1 root     root        0 Nov 16 07:46 filesystems
dr-xr-xr-x    2 root     root        0 Nov 16 07:46 fs
dr-xr-xr-x    4 root     root        0 Nov 16 07:46 ide
-r--r--r--    1 root     root        0 Nov 16 07:46 interrupts
-r--r--r--    1 root     root        0 Nov 16 07:46 ioports
-r--------    1 root     root   134221824 Nov 16 07:46 kcore
-r--------    1 root     root        0 Nov 15 19:01 kmsg
-r--r--r--    1 root     root        0 Nov 16 07:46 ksyms
-r--r--r--    1 root     root        0 Nov 16 07:46 loadavg
-r--r--r--    1 root     root        0 Nov 16 07:46 locks
-r--r--r--    1 root     root        0 Nov 16 07:46 meminfo
-r--r--r--    1 root     root        0 Nov 16 07:46 misc
-r--r--r--    1 root     root        0 Nov 16 07:46 modules
-r--r--r--    1 root     root        0 Nov 16 07:46 mounts
dr-xr-xr-x    3 root     root        0 Nov 16 07:46 net
-r--r--r--    1 root     root        0 Nov 16 07:46 partitions
-r--r--r--    1 root     root        0 Nov 16 07:46 pci
-r--r--r--    1 root     root        0 Nov 16 07:46 rtc
dr-xr-xr-x    2 root     root        0 Nov 16 07:46 scsi
lrwxrwxrwx    1 root     root       64 Nov 16 07:46 self -> 5038
-r--r--r--    1 root     root        0 Nov 16 07:46 slabinfo
-r--r--r--    1 root     root        0 Nov 16 07:46 stat
-r--r--r--    1 root     root        0 Nov 16 07:46 swaps
dr-xr-xr-x    9 root     root        0 Nov 16 07:46 sys
dr-xr-xr-x    4 root     root        0 Nov 16 07:46 tty
-r--r--r--    1 root     root        0 Nov 16 07:46 uptime
•r--r--r--    1 root     root        0 Nov 16 07:46 version
```

Первое, что бросается в глаза, это размер файлов. Размер практически всех файлов и каталогов в системе /proc равен нулю. Ненулевой размер имеет файл self, являющийся символической ссылкой, и файл kcore, который соответствует системной памяти.

Обратите внимание, что любой из файлов доступен только для чтения даже для владельца файла (в большинстве случаев владельцем является пользователь root). Именно благодаря этому вы не сможете выполнять в отношении системы /proc какие-либо манипуляции, вы сможете только просматривать ее содержимое.

Более внимательное изучение листинга позволяет разделить его на две части: файлы и каталоги с числовыми именами и файлы и каталоги с именами символическими. Владельцем большинства числовых имен является root, но встречаются и имена с другими владельцами. На самом деле каждое такое число — это идентификатор процесса (Process ID, PID). Процессы с такими идентификаторами выполнялись в системе в момент, когда был получен этот листинг. Таким образом, для каждого работающего в системе процесса в каталоге /proc существует подкаталог, имя которого совпадает с идентификатором этого процесса. В этом подкаталоге содержится информация об этом процессе. Выполнив команду ps a их, можно обнаружить, что разрешения в /proc для каждого из процессов соответствуют владельцу и группе соответствующего процесса в списке процессов. На листинге 6.2. показано содержимое одного из таких подкаталогов:

Листинг 6.2. Содержимое некоторого каталога /proc/PЮ

```
-r--r--r--    1  root     root        0 Nov   16 08:23 status
-r--r--r--    1  root     root        0 Nov   16 08:23 statm
-r--r--r--    1  root     root        0 Nov   16 08:23 stat
lrwx------    1  root     root        0 Nov   16 08:23 root -> /
-rw-------    1  root     root        0 Nov   16 08:23 mem
pr--r--r--    1  root     root        0 Nov   16 08:23 maps
dr-x------    2  root     root        0 Nov   16 08:23 fd
lrwx-----     1  root     root        0 Nov 16 08:23 exe -> /usr/sbin/klogd
-r--------    1  root     root        0 Nov   16 08:23 environ
lrwx-----     1  root     root        0 Nov   16 08:23 cwd -> /
-r--r--r--    1  root     root        0 Nov   16 08:23 cmdline
dr-xr-xr-x   58 root     root        0 Nov   15 19:01 ..
```

```
dr-xr-xr-x    3  root     root     0 Nov  16 08:23 .
```

В любом из каталогов содержатся одни и те же файлы (табл. 6.1). Некоторые из них являются символическими ссылками, указывающими в другие места системы, другие файлы содержат в себе некоторую информацию о процессе. Именно ее, только в обработанном и отформатированном виде, выводят такие утилиты, как ps, top и т. п.

Таблица 6.1. Файлы каталога, описывающего состояние процесса

Имя	Содержимое
cmdline	Аргументы командной строки
cwd	Ссылка на рабочий каталог (каталог, из которого была выполнена породившая процесс команда)
environ	Переменные окружения
exe	Ссылка на исполняемый файл процесса
fd	Каталог, содержащий дескрипторы файлов процесса, ссылки на другие файлы и т. п. 1r-x-----1 root root 64 Dec 13 08:56 0 -> /dev/null l-wx----- 1 root root 64 Dec 13 08:56 1 -> /var/log/xdm-errors l-wx----- 1 root root 64 Dec 13 08:56 2 -> /var/log/xdm-errors lrwx----- 1 root root 64 Dec 13 08:56 3-> socket:[1556] lrwx----- 1 root root 64 Dec 13 08:56 5 -> socket:[573]
maps	Исполняемые файлы и библиотеки, отображенные в память процесса
mem	Память процесса
root	Ссылка на корневой каталог процесса (обычно /)
stat	Состояние процесса
statm	Состояние памяти процесса
status	Состояние процесса в понятном для человека виде (формат файлов stat и statm не понятен для человека)

Что касается второй части листинга — файлов с символьными именами, — то большинство перечисленных в нем имен должны быть вам понятны. О характере содержимого этих файлов можно догадаться, взглянув на имя. Файл interrupts (прерывания), например, содержит ровно то, что и предполагает его название: список всех прерываний и устройств, использующих их. Определенная информация содержится также в подкаталогах. Структура каталога /proc на всех системах практически одинакова. Присутствие некоторого подкаталога для определенной конфигурации может оказаться бессмысленным, однако из соображений стандартизации этот подкаталог все равно будет присутствовать в /proc, просто он будет абсолютно пустым. Например, даже если в вашей системе нет никаких устройств SCSI, все равно вы сможете обнаружить в каталоге /proc подкаталог scsi, который будет пустым. В системе, базирующейся на SCSI, контроллер IDE может полностью отсутствовать, однако даже в этом случае в каталоге /proc будет присутствовать пустой подкаталог ide[2].

Тем, кто интересуется работой с /proc на уровне системного программирования, следует заглянуть в /usr/src/linux/include/linux/proc_fs.h. Данный файл содержит описания структур данных и функций, позволяющих представлять /proc в виде дерева в памяти и динамически добавлять в эту систему новые элементы. Еще одним источником информации о /proc является файл документации /usr/src/linux/Documentation/proc.txt. И хотя в этом файле собрана далеко не вся информация о /proc, это единственный более или менее полноценный документ, посвященный данной файловой системе.

При монтировании /proc создаются все поддерживаемые данной системой каталоги, но не файлы. Файлы создаются только для тех механизмов, которые действительно включены в состав ядра. Например, если поддержка звука была включена в ядро, а значит, присутствует в системе постоянно, то связанный с ней файл sound будет создан непосредственно при монтировании /proc. Если же поддержка звука реализована в виде отдельных модулей, то соответствующий файл появится в /proc лишь после загрузки необходимых для поддержки звука модулей. И до тех пор, пока не будет загружен модуль конкретной звуковой карты, этот файл будет пустым. После же загрузки такого модуля из этого файла можно будет узнать, какой модуль был загружен и с какими параметрами[3].

Это верно и для других подсистем, таких как ppp, slip и т. д. Связанные с этими подсистемами каталоги всегда будут присутствовать в /proc, тогда как соответствующие файлы могут отсутствовать. В качестве другого примера рассмотрим привод SCSI CD-ROM, поддерживаемый при помощи модуля. Пусть это будет модуль aha!54x компании Adaptec. Если этот модуль не загружен и привод CD-ROM является единственным SCSI устройством в системе, то изначально каталог /proc/scsi будет абсолютно пустым — в нем не будет существовать ни одного файла. После загрузки модуля aha!54x в этом каталоге появятся два файла: scsiO и sda (или sd(b|c|d), в зависимости от конфигурации устройства). В этих файлах будут содержаться всевозможные сведения об этом устройстве. Затем, после монтирования компакт-диска, появятся новые записи в /proc/filesystems и т. д.

– [2] Однако, судя по исходным текстам, созданием каталогов в /proc занимаются сами подсистемы, так что если нет поддержки SCSI, то и нет каталога scsi. Поэтому либо автор попросту не прав, либо «соображения стандартизации» относятся не к Linux в целом, а к дистрибутиву Caldera. — *Примеч. перев.*

– [1,2,3] В 2.4 я /proc/sound не нашел. По крайней мере, в 2.2 мне удалось, причем достаточно быстро, найти место в исходниках, где создается этот файл, а вот в 2.4 — нет. — *Примеч. перев.*

Далее, предположим, что по некоторым причинам система была переведена в однопользовательский режим (уровень выполнения 1), а все файловые системы размонтированы. После этого в системе были смонтированы корневая файловая система в режиме только для чтения и файловая система /proc. После этого вы хотите монтировать гибкий диск с файловой системой vfat в режиме только для чтения. Так как корневая система доступна лишь для чтения, команда mount не сможет создать файл /etc/mtab, содержащий записи о монтированных файловых системах (в этом случае можно воспользоваться ключом -n, который предписывает команде mount не вносить запись о монтируемой системе в файл /etc/mtab). В результате если вы попробуете узнать о монтированных файловых системах при помощи команд df или mount, гибкий диск будет отсутствовать в списке монтированных файловых систем. В подобной ситуации достоверную информацию о монтированных файловых системах можно получить при помощи /proc. Заглянув в файл /proc/filesystems, можно увидеть, что в системе включена поддержка vfat, а заглянув в файл /proc/mounts, вы узнаете о том, что именно в настоящий момент монтировано в системе.

Каталог /proc/sys

Среди всех прочих каталогов /proc каталог sys является особенным. Для построения ядра с поддержкой фильтрации пакетов необходимо также включить в ядро поддержку такой вещи, как sysctl (system control). Для этого в конфигурационном меню следует выбрать General Setup > Sysctl support. По умолчанию поддержка sysctl включена, то есть в файле конфигурации присутствует строка:

CONFIG_SYSCTL-y

Если в ядре присутствуют поддержка sysctl и поддержка /proc и если файловая система /proc смонтирована, то в каталоге /proc появляется подкаталог sys, содержащий в себе несколько файлов и подкаталогов. В отличие от остальной файловой системы /proc, которая предназначена только для чтения, каталог /proc/sys содержит в себе файлы, в которые можно записывать информацию.

ПРИМЕЧАНИЕ

В новых ядрах, содержащих поддержку mttr, в корне /proc располагается файл mttr, в который также можно производить запись. Более того, это единственный файл в /proc, если не считать символические ссылки и kcore, обладающий ненулевым размером, поэтому его можно считать исключением из общего правила.

Файлы каталога /proc/sys позволяют изменять состояние (параметры работы) ядра прямо в процессе его функционирования. Данной возможностью пользуются многие программы. Например, после запуска сервера XFree86 вы не сможете инициировать завершение работы системы при помощи комбинации клавиш Ctrl+Alt+Del, не переключившись предварительно на виртуальный терминал. Система ведет себя так потому, что X-сервер записал значение 0 в файл /proc/sys/ kernel/ctrl-alt-del и тем самым запретил перезагрузку при нажатии этой комбинации клавиш. Виртуальный терминал, наоборот, записывает в этот файл значение 1 и, таким образом, разрешает использование комбинации Ctrl+Alt+Del привычным для вас образом.

Кроме того, вы можете модифицировать файлы каталога /proc/sys напрямую. Если вы считаете, что можете настроить ядро лучше, чем это обеспечивается настройками по умолчанию, то с помощью /proc/sys вы сможете изменить настройки множества подсистем, включая виртуальную память, файловые системы и сетевые подсистемы, на которые приходится основная масса изменяемых через /proc/sys параметров. Более того, иногда это попросту необходимо. Например, чтобы включить механизм перенаправления IP (IP forwarding), нужно записать в файл /proc/sys/net/ipv4/ip_forward значение 1 (если перенаправление IP выключено, в этом файле содержится значение 0). По умолчанию в этом файле содержится 0. Очевидно, что без поддержки /proc и sysctl включить перенаправление IP в процессе работы ядра не удастся. Значение этого параметра (как, впрочем, и других подобных параметров), используемое по умолчанию, определено в исходных кодах ядра. Конечно, вы можете отредактировать исходные файлы ядра таким образом, чтобы по умолчанию механизм перенаправления IP был включен, однако это не самая лучшая идея. Кроме очевидных неудобств, связанных с переходом на новую версию ядра, в результате подобных изменений могут возникнуть проблемы и с текущим ядром. Значения параметров по умолчанию являются безопасными, иначе говоря, они специально подобраны так, чтобы скомпилированное с их использованием ядро нормально функционировало на большинстве самых разнообразных компьютеров. Потому если вы займетесь редактированием исходного кода ядра и будете изменять изначальные значения параметров работы какой-нибудь жизненно важной подсистемы, например виртуальной памяти, вы можете получить ядро, которое невозможно будет использовать на многих компьютерах, в том числе и на вашем. Гораздо правильней будет оставить исходные тексты в покое, а для управления параметрами работы ядра использовать другие способы (например, конфигурационные файлы или систему /proc/sys).

ВНИМАНИЕ

В результате изменения значений некоторых файлов из каталога /proc/sys (в особенности в подкаталоге vm) система может перейти в нестабильное состояние, поэтому прежде чем менять параметры работы какой-либо подсистемы, рекомендуется сначала внимательно прочитать документацию /proc, содержащуюся в каталоге /usr/src/linux/Documentation.

В ядрах версий 2.4x файловая система /proc существенно не изменится. Некоторые файлы будут называться по-другому, возможно, изменятся значения по умолчанию для некоторых параметров, возможно, появятся новые каталоги, но произойдет это не потому, что старая структура /proc обладала недостатками или приводила к проблемам с безопасностью, а вследствие структурных изменений самого ядра. Если меняется само ядро, значит, меняется и его отображение через файловую систему /proc.

Файловая система /dev/pts

Файловая система /dev/pts является одним из улучшений, появившихся в ядрах серии 2.1x. Она пришла смену псведотерминалам в стиле BSD, то есть устройствам /dev/pty?, дабы устранить некоторые их недостатки. Совместное существование обоих этих механизмов в одной системе невозможно: либо вы используете /dev/pty?, либо /dev/pts. В OpenLinux предпочтение отдано стилю /dev/pts.

Чтобы активизировать файловую систему /dev/pts, необходимо настроить два конфигурационных параметра ядра и добавить в файл /etc/fstab одну дополнительную запись. Первый параметр ядра устанавливается при помощи пункта Character devices > Unix98 PTY support. Там же можно указать и максимальное число псевдотерминалов, доступных для одновременного использования. По умолчанию можно использовать не более 256 псевдотерминалов, но при необходимости, например для серверных систем, это число можно изменить вплоть до максимального значения, указанного в справке. Для архитектуры Intel максимальное допустимое значение равно 2048.

CONFIG_UNIX98_PTYS=y

Второй параметр находится в меню File systems и называется /dev/pts file system for Unix98 PTYs. Без него система будет продолжать использовать псевдотерминалы в стиле BSD.

CONFIG_DEVPTS_FS=y

После получения ядра с поддержкой файловой системы /dev/pts эту файловую систему необходимо активизировать, для чего в /etc/fstab нужно поместить следующую строку:

devpts /dev/pts devpts gid=5,mode=620 0 O

Если заместить файл /etc/fstab, устанавливаемый Caldera OpenLinux 2.2 или выше, файлом /etc/fstab для ядра 2.0.x, то вы получите неработоспособную систему, поскольку она не сможет создать необходимые устройства. Эти устройства создаются в каталоге /dev/pts прямо в процессе функционирования системы и нумеруются.

Приведенная здесь запись в файле /etc/fstab означает, что файловая система находится на устройстве devpts, точкой ее монтирования является /dev/pts, тип файловой системы — devpts, монтируется она с GID, равным 5, что соответствует системному пользователю tty, а режим доступа rw--w----. Режим 620 позволяет другим пользователям пересылать на данный псевдотерминал сообщения и эквивалентен команде mesg=y. Чтобы запретить посылку сообщений на данный псевдотерминал, нужно выполнить команду mesg=n или изменить его режим на 600.

В результате открытия нового сеанса telnet, xterm и т. п. в файловой системе /dev/pts создается новое устройство. Максимальное количество таких устройств определяется, как уже говорилось, при конфигурации ядра. При закрытии сеанса соответствующее ему устройство удаляется из каталога /dev/pts.

Соображения безопасности для /proc

Файловая система /proc является источником важной информации обо всей системе. Любой, у кого есть полный доступ к /proc, может воспользоваться этим механизмом для получения самых разных сведений о системе, а также для изменения параметров ее работы при помощи файлов /proc/sys.

Однако если внимательно проанализировать конфигурацию на доступ к файлам из /proc, то можно увидеть, что непривилегированные пользователи могут извлечь из /proc очень мало информации помимо той, которую они могут получить при помощи различных утилит. Все действительно важные файлы, содержащие чувствительную информацию, обычному пользователю недоступны. Ну а если рассматривать ситуацию со злоумышленником, получившим привилегии суперпользователя, то /proc относится к тем вещам, о которых следует беспокоиться в последнюю очередь. Злоумышленник и так может делать с

системой все что угодно, поэтому информация из /proc ему, скорей всего, малоинтересна.

Разумеется, непривилегированные пользователи не смогут ничего сделать и с файлами из /proc/sys. Отказ от использования /proc как таковой, равно как и отказ от использования /proc/sys, не приведет к улучшению безопасности, а лишь значительно усложнит администрирование системы. Файловая система /proc была разработана не просто так, а потому, что она делает взаимодействие с системой более удобным как для человека, так и для различных утилит.

ВНИМАНИЕ

Не следует делать резервное копирование содержимого /proc. Причин тому несколько. Одна из них заключается в том, что в силу особенностей этой файловой системы сама по себе операция восстановления /proc с ленты является бессмысленной и ненужной. Поэтому не стоит тратить на резервное копирование /proc время и пространство на ленте. Ленты следует хранить в надежном месте, поскольку важные системные файлы, содержащиеся на них, например /etc/shadow и т. д., могут быть восстановлены на другой системе.

Для файловой системы devpts тоже есть некоторые соображения, связанные с безопасностью. Правом чтения из файлов псевдотерминалов должны обладать только их владельцы (устанавливаемый по умолчанию режим 620 обеспечивает это). Обладая правом чтения такого файла, пользователь получает возможность читать коды клавиш, пересылаемых данному псевдотерминалу из буфера клавиатуры. Таким образом можно перехватывать пароли в момент их ввода пользователем. Подобная опасность угрожает многим файлам устройств, однако по умолчанию все они создаются с использованием корректной конфигурации разрешений на доступ, благодаря чему подобных опасностей удается избежать.

Заключение

В этой главе речь шла о файловой системе /proc, являющейся окном, через которое можно наблюдать за состоянием и работой ядра операционной системы. Кроме того, в ней рассказывалось о каталоге /proc/sys, позволяющем управлять параметрами ядра во время его функционирования.

После этого мы рассмотрели еще одну не совсем обычную файловую систему -devpts, которая реализует поддержку псевдотерминалов стандарта Unix98 PTY. И хотя потенциально файлы псевдотерминалов могут быть использованы для «прослушивания» вводимой через них информации, соответствующий режим в /etc/ fstab делает это невозможным.

7 Процесс загрузки

В данной главе рассматриваются следующие вопросы:
- процесс начальной загрузки ОС;
- программа init
- файл /etc/inittab и инициализация в стиле System V;
- сценарии инициализации.

Чтобы понять, как именно можно нарушить защиту системы во время начальной загрузки, необходимо хорошо представлять себе, как именно загружается операционная система и какие процедуры при этом выполняются. Для квалифицированного взломщика, имеющего физический доступ к системе, загрузка — это время, когда система защищена менее всего. Консольные атаки будут рассмотрены в следующей главе, а в данной главе мы рассмотрим собственно процесс начальной загрузки и инициализации системы.

В системе Caldera Open Linux инициализация осуществляется в стиле System V, альтернативой которому является стиль BSD. Это два основных стиля, используемых для инициализации систем Unix и Linux. Стиль BSD основан на использовании нескольких больших сценариев, обеспечивающих всю инициализацию, тогда как стиль System V базируется на концепции уровней выполнения (runlevels) и наборе небольших сценариев инициализации. Сценарии инициализации используются для запуска и остановки демонов (фоновых процессов) при переходе с одного уровня выполнения (также называемого *состоянием системы* — system state — или просто *состоянием* — state) на другой. Соответственно, для каждого демона или подсистемы имеется свой сценарий, и все такие сценарии хранятся в специально отведенном для этого каталоге, который будет рассмотрен во всех подробностях далее. Всего имеется семь уровней выполнения, от 0 до 6, каждый из которых соответствует некоторому режиму функционирования системы, но зачастую не все из них используются. Новичкам, особенно пришедшим в Linux из мира DOS или Windows, такой подход к инициализации может показаться несколько запутанным, однако это всего лишь плата за гибкость, недоступную для инициализации в стиле BSD.

В BSD, наоборот, все инициализирующие действия запрятаны внутрь нескольких больших сценариев. Этот стиль инициализации является характерной особенностью дистрибутива Slackware. И хотя в настоящее время стиль BSD не так популярен, как стиль System V, он решает ту же задачу, что и стиль System V, только несколько иным способом. В Slackware имеется всего два режима: S (от single user), означающий однопользовательский режим, и M (от multiuser), означающий многопользовательский режим. Но несмотря на это и Slackware, и Caldera пользуются для инициализации одними и теми же исполняемыми файлами. Различие проявляется на более высоком уровне: на уровне файла конфигурации программы init (файла inittab).

Процесс загрузки

Все компьютеры семейства x86 загружаются одинаково. Детали этого процесса для целей данного текста значения не имеют и потому будут опущены. Процесс загрузки представляет собой последовательность событий, начинающуюся с включения питания. Обычно после этого начинает выполняться проверка оборудования, известная как POST (Power On Self Test), во время которой подсистемы компьютера проверяются на наличие проблем, мешающих его нормальному функционированию, таких как плохая память и прочее. Как правило, успешное завершение этого теста сопровождается одиночным гудком из динамика компьютера. Два гудка или более означают наличие проблемы. По количеству гудков можно определить причину проблемы.

После успешного завершения POST запускается небольшая программа-загрузчик, которая, в свою очередь, запускает загрузчик побольше. Этот второй загрузчик последовательно просматривает все места,

в которых может содержаться загрузочный код операционной системы, и запускает на выполнение первый найденный. Обычно порядок просмотра следующий: загрузочный сектор гибкого диска, загрузочный сектор первого жесткого диска, загрузочный сектор компакт-диска в первом приводе CD-ROM. Изменение порядка просмотра осуществляется при помощи параметров BIOS, для доступа к которым следует нажать специальную клавишу (F2, Del или Insert) во время выполнения POST.

Обнаружив загрузочный сектор, в котором содержится загрузочный код ОС, система перемещает его в память и запускает на выполнение. В большинстве Linux систем это будет код LILO (от Linux Loader). LILO — это программа, позволяющая пользователю выбрать, какую из установленных на компьютере операционных систем (или различных ядер Linux) следует загрузить. Загрузчик LILO удобно использовать в случае, если на компьютере установлено несколько разных ОС (например, Linux и Windows), но даже если система всего одна, загрузчик LILO позволяет выбрать один из нескольких вариантов конфигурации этой системы. Будучи написан специально для Linux, загрузчик LILO позволяет передавать дополнительные аргументы ядру или программе init.

ПРИМЕЧАНИЕ

В данном тексте обсуждается архитектура Intel. В других архитектурах, таких как Spare или Alpha, используются похожие загрузчики, но с другими именами, например SILO (Spare) или MILO (Alpha) и т. д.

После того как пользователь сделает свой выбор, LILO загружает ядро ОС. Подробное обсуждение процесса загрузки ядра выходит за рамки этой книги, достаточно сказать, что это многоступенчатый процесс. Как правило, на диске образ ядра хранится в сжатом виде, а при загрузке в память он разжимается. В мире Linux принято использовать букву z как признак того, что данный образ является сжатым, например vmlinuz, zImage или bzImage. Отсюда следует, что, в отличие от DOS, образ ядра считывается с диска лишь во время загрузки. После загрузки никаких обращений к образу ядра на диске не делается, поэтому удаление или модификация соответствующего файла никак не влияет на функционирующее ядро. Для вступления в силу изменений, затронувших образ ядра на диске, система должна быть перезагружена. Кроме того, по размеру образа ядра на диске нельзя сказать, сколько места оно займет в памяти.

init — место, откуда начинается инициализация системы

После того как ядро было загружено в память, систему можно считать работающей. Однако пользы от ее работы не слишком много, поскольку взаимодействием с пользователями ядро не занимается. Получив управление, ядро запускает одну (и только одну) программу: init. Эта программа отвечает за выполнение всех остальных процедур и является родителем всех процессов. Передав управление init, ядро переходит к выполнению своей обычной функции менеджера системы, то есть к управлению пространством ядра и распределению ресурсов между процессами.

ПРИМЕЧАНИЕ

Под пространством ядра (kernel space) понимается память и функции, принадлежащие исключительно ядру. Эта память является защищенной. Дополнительным к пространству ядра является пространство пользователя (user space), в котором выполняются все программы, запущенные пользователями (включая суперпользователя).

Начав работу, программа init обращается к расположенному в каталоге /etc файлу inittab (от *initialization table — таблица инициализации)* для получения значений конфигурационных параметров. Для отсутствующих в нем параметров используются значения по умолчанию, но только если они не были переопределены из командной строки. Листинг 7.1 показывает содержимое файла inittab, используемого в OpenLinux 2.3.

СОВЕТ

Суффикс tab в названии таких файлов, как inittab, fstab, mtab и т. д., означает, что файл представляет собой таблицу, как правило, конфигурационную. В случае с inittab в ней содержится информация об инициализации системы, примерно как в файле config.sys в DOS.

Листинг 7.1. Файл /etc/inittab с добавленными номерами строк[4]

```
1  #
2  # inittab    This file describes how the INIT process should set up
3  #               the system in a certain run-level.
4  #
5  # Author:     Miquel  van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
6  #               Modified for RHS Linux by Marc Ewing and Donnie Barnes
7  #               Modified for COL by Raymund Will
8  #
9
10  # The runlevels used by COL are:
11  #  0 - halt (Do NOT set initdefault to this)
12  #  1 - Single user mode (including initialisation of network interfaces,
13  #    if you do have networking)
14  #  2 - Multiuser, (without NFS-Server und some such)
15  #     (basically the same as 3, if you do not have networking)
16  #  3 - Full multiuser mode
17  #  4 - unused
18  #    (should be equal to 3, for now)
19  #  5 - X11
20  #  6 - reboot (Do NOT set initdefault to this)
21
22  #
23  # Default runlevel.
24  id:5:initdefault:
25
26  # System initialization.
27  sO::sysinit:/bin/bash -c 'C=/sbin/booterd; [ -x $C ] && $C'
28  si::sysinit:/bin/bash -c 'C=/etc/rc.d/rc.modules: [ -x $C ] && $C default'
29  s2::sysinit:/bin/bash -c 'C=/etc/rc.d/rc.serial; [ -x $C ] && $C'
30  bw::bootwait:/etc/rc.d/rc.boot
31
32  # What to do in single-user mode.
33  ~1:S:wait:/etc/rc.d/rc 1
34  ~~:S:wait:/sbin/sulogin
35
36  10:0:wait:/etc/rc.d/rc 0
37  11:l:wrtt:/etc/rc.d/rc 1
38  12:2:wait:/etc/rc.d/rc 2
39  13:3:wait:/etc/rc.d/rc 3
40  14:4:wait:/etc/rc.d/rc 4
41  15:5:wait:/etc/rc.d/rc 5
42  16:6:wait:/etc/rc.d/rc 6
43  # Normally not reached, but fallthrough in case of emergency.
44  z6:6:respawn:/sbin/sulogin
45
46  # Trap CTRL-ALT-DELETE
47  ca:12345:Ctrlaltdel:/sbin/shutdown  -t3  -r now
48
49  # Action on special  keypress (ALT-UpArrow).
50  kb::kbrequest:/bin/echo "Keyboard Request--edit /etc/inittab to let this work."
51
52  # When our UPS tells us power has failed,  assume we have a few minutes
53  # of power left.    Schedule a shutdown for 2 minutes from now.
54  # This does, of course,  assume you have powerd installed and your
55  # UPS connected and working correctly.
56  pf::powerfail:/sbin/shutdown  -h +5 "Power Failure:  System Shutting Down"
57
58  # If battery is fading fast -- we hurry...
59  pi::powerfailnow:/sbin/shutdown -c 2> /dev/null
60  p2::powerfailnow:/sbin/shutdown -h now "Battery Low..."
61
62  # If power was restored before the shutdown kicked in, cancel it.
63  po:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"
64
65
66  # Run gettys in standard runlevels
67  1:12345:respawn:/sbin/getty ttyl VC linux
68  2:2345:respawn:/sbin/getty tty2 VC linux
69  3:2345:respawn:/sbin/getty tty3 VC linux
```

[4] Комментарии в листинге оставлены без перевода, чтобы не нарушить нумерацию строк. Подробное описание работы сценария см. далее в книге. — *Примеч. перев*

```
70 4:2345:respawn:/sbin/getty tty4 VC linux
71 5:2345:respawn:/sbin/getty tty5 VC linux
72 6:2345:respawn:/sbin/getty tty6 VC linux
73
74 # Run kdm in runlevel 5
75 kdm:5:respawn:/opt/kde/bin/kdm -nodaemon > /var/log/kdm 2>&1
```

Файл /etc/inittab, содержимое которого приведено в листинге 7.1., позаимствован из комплекта OpenLinux 2.3 компании Caldera. В другом комплекте Linux содержимое этого файла почти наверняка будет иным. Однако если сравнить между собой комплекты Caldera и Red Hat (равно как и SuSE, Mandrake или Debian), то различий наберется куда меньше, чем при сравнении любого из этих комплектов и комплекта Slackware. В каждом из комплектов Linux используется одна и та же программа инициализации init (в любом из них эта программа получается в результате компиляции одного и того же исходного кода), то вся разница в процессах инициализации задается исключительно файлом inittab, подробным рассмотрением которого мы и займемся.

Структура inittab

Строки файла inittab, начинающиеся с символа #, являются комментариями и потому командой init не обрабатываются. Очевидно, они предназначены для пользователей системы и помогают понять логику работы файла. Остальные строки имеют обычную для конфигурационных таблиц структуру в виде набора полей, разделенных двоеточием. Поля эти, если перечислять их слева направо, следующие:

- id (идентификатор). Первое поле содержит уникальный идентификатор строки, состоящий из алфавитно-цифровых символов. Максимально допустимая длина идентификатора равна четырем, но на практике, как правило, используются двухсимвольные идентификаторы. В старых системах длина этого поля не могла превышать двух, и большинство комплектов Linux (все, которые я знаю), еще не отошли от этого ограничения;

- runlevel (уровень выполнения). Во втором поле содержится уровень выполнения, на котором следует выполнять эту строку. Если конкретный уровень не указан, строка выполняется на всех уровнях. Примером такой строки с пустым полем уровня выполнения является строка 60 в листинге 7.1;

- action (действие). Значением этого поля является одно из предопределенных ключевых слов. Чаще всего это respawn, но также допустимы и boot, bootwait, ctrlaltdel, initdefault, kbrequest, off, once, ondemand, powerfail, powerokwait, powerwait и sysinit;

- process (процесс). Процесс или программа для выполнения.

Каждой строке файла inittab ставится в соответствие уникальный идентификатор. Выбирать его принято так, чтобы он легко ассоциировался с выполняемым действием. Например, для строки, порождающей getty применительно к первому последовательному порту, подойдет идентификатор s1.

По умолчанию уровни выполнения обозначаются цифрами от 0 до 6 и буквами от A до C (или от a до c). Уровни выполнения 0,1 и 6 являются специальными, и без особой надобности их изменять не следует. Уровень 0 соответствует останову системы, уровень 1 — однопользовательскому режиму, уровень 6 — перезагрузке, поэтому изменение любого из этих уровней может иметь далеко идущие последствия. Остальные уровни, то есть уровни 2-5, можно настраивать совершенно свободно.

ПРИМЕЧАНИЕ

Традиционно для передачи команд init используется telinit, однако можно вызывать init и напрямую. В OpenLinux файл telinit является не отдельной программой, а символической ссылкой, указывающей на init. В других комплектах Linux для этой цели может использоваться жесткая ссылка. Так или иначе, в конечном итоге все равно вызывается init.

В некоторых комплектах Linux имеется команда runlevel (которая, как правило, устанавливается в каталоге /sbin), отображающая номера предыдущего и текущего уровней выполнения. Если предыдущий уровень выполнения отсутствует, как это бывает после перезагрузки, то вместо него выводится буква N, например N 3. Если затем перевести систему в состояние 2 и снова выполнить команду runlevel, то команда отобразит комбинацию чисел 3 2. Кроме того, текущий уровень выполнения отображается как аргумент команды init в листинге информации о процессах (ps ax).

Уровни выполнения всегда обозначаются числами, но программе init в качестве уровня исполнения можно передавать и символьные аргументы. Аргумент Q обсуждается в следующем абзаце. Символы A, B и C (или a, b, c, что функционально одно и то же) используются только с действием ondemand, которое, в свою очередь, используется только с уровнями выполнения A-C. Данное действие позволяет суперпользователю (только суперпользователь может вызывать inittab) порождать процессы по своему

усмотрению.

Аргумент Q (можно использовать нижний регистр — q, — это одно и то же) предписывает init заново прочесть конфигурационный файл inittab. Менять inittab можно когда угодно и сколько угодно раз, но заново считывается он лишь в следующих случаях:

- завершение процесса, порожденного init (порождать еще один?);
- получение сигнала о сбое питания (от демона или из командной строки);
- переход с одного уровня выполнения на другой;
- обращение к init с аргументом Q или q.

inittab от начала и до конца

В этом разделе мы пройдемся по строкам листинга 7.1. Ранее мы ознакомились с форматом файла inittab, теперь же рассмотрим, как он работает на практике.

Строки 1-24 являются комментариями, поэтому мы их пропустим и начнем сразу со строки 24. Обычно такая строка выполнялась бы только на уровне выполнения 5. Но в данной записи указано действие initdefault, означающее, что данный уровень выполнения следует сделать уровнем выполнения по умолчанию (этот уровень будет использоваться в случае, если программа init запущена без указания требуемого уровня в командной строке). В OpenLinux (и некоторых других комплектах Linux) уровню 5 ставится в соответствие графический вход в систему, поэтому если вы не желаете, чтобы при загрузке системы подключение к ней осуществлялось в графическом режиме, измените уровень выполнения по умолчанию на 3. Это замечание справедливо также для комплекта Red Hat, однако в Debian уровень выполнения по умолчанию равен 2, а тип входа в систему (графический/обычный) зависит не от уровня выполнения, а от того, запускается или нет сценарий xdm.

ПРИМЕЧАНИЕ

В данном случае рассматривается система Caldera OpenLinux, откуда и взят рассматриваемый здесь файл inittab, поэтому если вы используете другой комплект Linux или другую версию OpenLinux, то ваш файл inittab, скорее всего, будет отличаться от рассматриваемого здесь. Однако основные концепции в любом случае будут одинаковы.

После строки установки уровня выполнения по умолчанию следуют строки (27-30), в которых осуществляется запуск сценариев инициализации системы. Как явствует из их названий, эти сценарии загружают модули, инициализируют последовательные порты и выполняют прочие действия, которые необходимо выполнить в процессе загрузки системы. В строке 30 указано действие bootwait, поэтому на ней программа init приостановится и будет ждать, пока не завершится выполнение указанного в ней сценария. Все четыре строки (27-30) будут выполняться одновременно, однако до завершения процесса из строки 30 init не будет запускать никаких других процессов. Это позволяет завершить все подготовительные действия, необходимые для перехода к следующему этапу: запуску демонов и порождению getty.

Строки 33 и 34 вызывают два сценария, выполняющихся при переходе системы в состояние 1, иначе говоря, в однопользовательский режим. Программа sulogin нужна для того, чтобы пользователь, обладающий возможностью физического доступа к системе, не смог получить привилегии суперпользователя без знания соответствующего пароля. В противном случае любой желающий смог бы получить полную власть над системой, просто перезагрузив ее в однопользовательском режиме. Без этой строки система, загрузившись в однопользовательском режиме, не станет запрашивать пароль у пользователя, а сразу же предоставит ему доступ к командной оболочке с привилегиями суперпользователя. Однако при наличии физического доступа к системе защиту с использованием программы sulogin можно обойти, о чем подробней рассказывается в следующей главе.

Строки с 36 по 42 выполняются при смене уровней выполнения. Каждая такая строка представляет собой вызов сценария с именем rc и номером уровня, на который переходит система, в качестве аргумента. Так, при переходе с уровня 5 на уровень 3 будет выполнена строка 39, которая вызовет rc с аргументом 3.

Строка 44 выводит запрос на ввод пароля суперпользователя. Данная строка является мерой предосторожности на случай, если после перехода на уровень выполнения 6 система не перезагрузится должным образом.

Строка 47 перехватывает комбинацию клавиш Ctrl+Alt+Del, осуществляя по ней перезагрузку системы. При желании вместо перезагрузки можно получить останов системы, для чего в этой строке нужно заменить -r на -h. Или же можно просто убрать -r, и тогда по нажатию этих клавиш система будет переходить в однопользовательский режим, предоставляющий доступ к командной оболочке с правами суперпользователя.

Строка 50 перехватывает комбинацию клавиш Alt+T. На данный момент никаких осмысленных действий по ней не делается, а лишь выводится сообщение о том, что ее нужно настраивать самостоятельно. При установке OpenLinux по умолчанию, то есть с использованием bash, данная последовательность команд игнорируется (это также относится и к большинству других дистрибутивов, в которых bash является оболочкой по умолчанию). Поэтому сначала нужно изменить настройку соответствия клавиш таким образом, чтобы bash передавал эту комбинацию, не интерпретируя ее, после чего можно заменять ту часть строки, где стоит echo, на удобную для вас команду.

Строки 56, 59, 60 и 63 представляют интерес лишь для тех, у кого есть источник бесперебойного питания, совместимый с демоном питания powerd. Демон питания может наблюдать за состоянием последовательно порта и в случае возникновения тех или иных событий, связанных с питанием, этот демон способен выполнять действия, указанные в файле inittab. В комплект поставки OpenLinux демон питания не входит. Проблема в том, что в настоящее время лишь очень немногие источники бесперебойного питания работают с демоном, независимо от того, какой у вас комплект Linux.

Строки с 67 по 72 порождают процессы getty для виртуальных терминалов. Обратите внимание, что в однопользовательском режиме доступен лишь один виртуальный терминал. Если хочется иметь побольше свободной памяти, то можно уменьшить число виртуальных терминалов на уровне выполнения 5. Например, до одного: в графическом режиме они особенно и не нужны, поэтому одного будет вполне достаточно. Для увеличения числа виртуальных терминалов нужно просто добавить дополнительные строки к уже существующим (с уникальными идентификаторами, разумеется).

В строках 50, 56, 59 и 60 поле уровня выполнения содержит пустую строку. Это означает, что данные строки выполняются на любом уровне.

Наконец, в последней строке с номером 75 на уровне выполнения 5 порождается процесс kdm (KDE Display Manager). В комплекте Red Hat вместо kdm может использоваться xdm или gdm. В Debian всегда используется уровень выполнения 2, а xdm запускается не через inittab, а точно так же, как и все остальные серверы.

ПРИМЕЧАНИЕ

Если ваша система загрузилась в однопользовательский режим, хотя уровень выполнения по умолчанию у нее отличен от 1, значит, сценарий reboot обнаружил проблему с жестким диском, требующую ручного запуска fsck. После исправления проблемы система вновь начнет загружаться в многопользовательском режиме.

Сценарии rc, часть первая

В OpenLinux все сценарии инициализации хранятся в каталоге /etc/rc.d. В нем также имеются подкаталоги rc0.d - rc6.d, где номер соответствует уровню выполнения, а также подкаталог init.d. В каталогах rc#.d (# означает номер уровня исполнения) находятся символические ссылки, указывающие на сценарии из подкаталога /etc/rc.d/init.d (листинг 7.2). Всем сценариям из init.d можно передавать аргументы start и stop, а некоторым — еще два аргумента: reload и restart. Хотя реализация аргументов restart/reload несложна, по крайней мере один демон (dhcp) по сигналу SIGHUP неправильно считывает свой конфигурационный файл. Сигнал SIGHUP (от английского слова hangup — повесить телефонную трубку, дать отбой) означает, что демон должен заново прочитать файл конфигурации и продолжить работу с новой конфигурацией.

СОВЕТ

Сигналы, адресуемые работающим в системе процессам, являются средством передачи команд, предписывающих выполнение этими процессами определенных действий. Чаще всего они применяются для посылки фоновым процессам команды завершения работы. Разрешается использовать сигналы, номера которых лежат в диапазоне от 1 до 31. Команде завершения работы соответствует сигнал SIGTERM (15). Другими часто используемыми сигналами являются SIGKILL (9) и SIGHUP (1). Полный список сигналов можно получить при помощи команды man 7 signal. Посылкой сигналов процессам занимается команда kill, принимающая в качестве параметров имя или номер сигнала и идентификатор процесса, например kill -SIGHUP PID или kill -1 PID. При отсутствии параметра-сигнала посылается сигнал SIGTERM.

Листинг 7.2. Частичный список файлов каталога /etc/rc.d/

```
drwxr-xr-x              2 root          root        1024 Aug 30 07:25 init.d
-rwxr-xr-x1 root          root          5336 Aug 16 22:45 re
-rwxr-xr-x1 root          root          8930 Jul    7 08:55 re.boot
-rwxr-xr-x1 roo                                                    t       root            478 Aug 27 08:48 re.local
-rwxr-xr-x1 root          root          2809 Jul  14 16:45 re.modules
-rwxr-xr-x1 root          root          5586 Aug 16 22:45 rc.orig
-rwxr-xr-x1 root          root          10903 Mar 12   1999 re.serial
.    drwxr-xr-x         2 root          root        1024 Aug 11 23:04 rcO.d
drwxr-xr-x              2 root          root        1024 Aug 11 23:04 rcl.d
drwxr-xr-x              2 root          root        1024 Aug 11 23:04 rc2.d
drwxr-xr-x              2 root          root        1024 Aug 11 23:04 rcS.d
drwxr-xr-x              2 root          root        1024 Aug 11 23:04 rc4.d
drwxr-xr-x              2 root          root        1024 Aug 30 07:25 rcS.d
drwxr-xr-x              2 root          root        1024 Aug 30 07:25 rc6.d
-rwxr-xr-x1 root                                                                    root            846 Jun 29 11:02
unconfigured.sh
init.d/
-rwxr-xr-x1 root          root          2144 Jul  14 16:56 bigfs
-rwxr-xr-x1 rootroot            864 Jan 28    1999 cron
-rwxr-xr-x        1 root          root   1364 Apr 13 13:38 dhcpd
-rwxr-xr-x        1 root          root   6920 Jul 14 16:55 functions
-rwxr-xr-x        1 root    root       833 Jul 14 22:46 gpm
-rwxr-xr-x        1 root          root   1296 Aug 16 22:45 halt
-rwxr-xr-x        1 root    root       983 May 11 11:24 httpd
-rwxr-xr-x        1 root          root   1243 Jul 14 15:52 inet
-rwxr-xr-x        1 root    root       978 Jul 7 05:30 keytable
lrwxrwxrwx        1 root    root        11 Aug 5 12:35 local -> ../re.local
-rwxr-xr-x        1 root    root       804 Jul 14 23:09 logoutd
-rwxr-xr-x        1 root    root       931 Nov 4 1998 Ipd
-rwxr-xr-x        1 root          root   1720 Jul 14 22:04 mta
-rwxr-xr-x        1 root          root   1294 Jul 27 23:16 named
-rwxr-xr-x        1 root          root   2260 Jul 14 16:05 netmount
-rwxr-xr-x        1 root          root   2072 Jul 14 16:23 network
-rw-r-xr-x        1 root    root       791 Jul 27 23:22 news
-r-xr-xr-x        1 root          root   1863 Jun 22 07:57 nfs
-rwxr-xr-x        1 root          root   1232 Jan 7 1998 nis-client
-rwxr-xr-x        1 root    root      831 Jan 8 1998 nis-server
-rwxr-xr-x        1 root          root   1489 Apr 3 00:26 ntp
-r-xr-xr-x        1 root          root   3157 Aug 27 10:55 pcmcia
-rwxr-xr-x        1 root    root      649 May 26 1998 ppp
lrwxrwxrwx        1 root    root       4 Aug 5 12:35 reboot -> halt
-rwxr-xr-x        1 root    root      238 Oct 2 1997 rmnologin
-rwxr-xr-x        1 root    root       780 Oct 2 1997 rstatd
-rwxr-xr-x        1 root          root   1130 Jul 14 22:06 rusersd
-rwxr-xr-x        1 root          root   1130 Jul 14 22:05 rwalld
-rwxr-xr-x        1 root          root   1130 Jul 14 22:06 rwhod
-rwxr-xr-x        1 root          root   1211 Jul 15 03:10 samba
-rwxr-xr-x        1 root    root       969 Nov 25 1997 single
-rwxr-xr-x        1 root          root   1159 Apr 9 1998 skeleton
-rwxr-xr-x        1 root    root      607 Apr 29 04:15 skipped
-rwxr-xr-x        1 root          root   1714 Jul 14 20:08 squid
-rwxr-xr-x        1 root          root   1147 Mar 25 06:28 syslog
-rwxr-xr-x        1 root          root   1060 Mar 16 1999 urandom
-r-xr-xr-x        1 root          root   6949 Aug 30 07:25 vmware
-rwxr-xr-x        1 root    root       259 Jul 21 09:27 webmin
-rwxr-xr-x        1 root    root       990 Aug 10 17:48 xdm
-rwxr-xr-x        1 root    root       948 Mar 25 07:53 zap
rc2.d:
lrwxrwxrwx        1 root          root        14 Aug   5 12:35 KO5news -> ../init.d/news
lrwxrwxrwx        1 root          root        15 Aug   5 12:35 K09samba -> ../init.d/samba
lrwxrwxrwx        1 root          root        15 Aug   5 23:51 K12squid -> ../init.d/squid
lrwxrwxrwx        1 root          root        13 Aug   5 12:35 K25gpm -> ../init.d/gpm
lrwxrwxrwx        1 root          root        15 Aug   5 09:01 K25httpd -> . ./init.d/httpd
lrwxrwxrwx        1 root          root        17 Aug   5 12:35 K3Ologoutd -> ../init.d/logoutd
lrwxrwxrwx        1 root          root        16 Aug   5 12:35 K39rwalld -> ,/init.d/rwalld
lrwxrwxrwx        1 root          root        16 Aug   5 12:35 K40rstatd -> ../init.d/rstatd
lrwxrwxrwx        1 root          root        15 Aug   5 12:35 K44dhcpd -> ../init.d/dhcpd
```

```
lrwxrwxrwx    1 root    root    17 Aug  5 12:35 K47rusersd -> ../init.d/rusersd
lrwxrwxrwx    1 root    root    15 Aug  5 12:35 K48rwhod -> ../init.d/rwhod
lrwxrwxrwx    1 root    root    13 Aug  5 12:35 K5Omta -> ../init.d/mta
lrwxrwxrwx    1 root    root    13 Aug  5 12:35 K60nfs -> ../init.d/nfs
lrwxrwxrwx    1 root    root    13 Aug  5 12:35 K70ntp -> ../init.d/ntp
lrwxrwxrwx    1 root    root    17 Aug  5 12:35 K73ipxripd -> ../init.d/ipxripd
lrwxrwxrwx    1 root    root    20 Aug   5 12:35 K79nis-client -> . ./init.d/nis-client
lrwxrwxrwx    1 root    root    20 Aug  5 12:35 K80nis-server -> . ./i nit. d/nis-server
lrwxrwxrwx    1 root    root    14 Aug  5 12:35 K85inet -> ../init.d/inet
lrwxrwxrwx    I root    root   15 Aug5 21:25 K90named -> . ./init.d/named
lrwxrwxrwx    1 root    root   17 Aug5 12:35 S0lnetwork -> ../Init.d/network
lrwxrwxrwx    1 root    root   16 Aug5 12:35 S0lpcmcia -> . ./imt.d/pcracia
lrwxrwxrwx    1 root    root   16 Aug5 12:35 S05syslog -> ../init.d/syslog
lrwxrwxrwx    1 root    root   17 Aug5 12:35 S05urandom -> ../init.d/urandom
lrwxrwxrwx    1 root    root   18 Aug5 12:35 S20netmount -> ../init.d/netmount
lrwxrwxrwx    1 root    root   13 Aug5 09:02 S26ipx -> ../init.d/ipx
lrwxrwxrwx    1 root    root   13 Aug5 12:35 S351pd -> ../init.d/lpd
lrwxrwxrwx    1 root    root   14 Aug5 12:35 S40cron -> ./init.d/cron
lrwxrwxrwx    1 root    root   13 Aug5 12:35 S41atd -> ../init.d/atd
lrwxrwxrwx    1 root    root   18 Aug5 12:35 S75keytable •> ../init.d/keytable
lrwxrwxrwx    1 root    root   15 Aug5 12:35 S981ocal   -> . ./irrit.d/local
lrwxrwxrwx    1 root    root   15 Aug5 12:35 S99bigfs -> ../init.d/bigfs
lrwxrwxrwx    1 root    root   19 Aug5 12:35 S99rmno1ogin -> ../init.d/rmnologin
lrwxrwxrwx    1 root    root   17 Aug5 12:35 S99skipped -> ../init.d/skipped
lrwxrwxrwx    1 root    root   13 Aug5 12:35 S99zap -> ../init.d/zap
```

Имена всех файлов из каталогов /etc/re.d/rc#.d начинаются либо с буквы S (от английского слова *start* — запуск), либо с буквы К (от английского слова *kill* — уничтожение). Соответственно, сценарии первой категории занимаются запуском демонов, а второй категории — их остановом. Вслед за первым символом идет число, определяющее относительный порядок выполнения сценариев, за которым следует собственно информационная часть имени, которая, как правило, совпадает с именем соответствующего сценария из init.d, на который указывает данная ссылка. Рассмотрим, например, файл S35lpd. Это символическая ссылка, указывающая на файл ../init.d/lpd, используемая сценарием rc для вызова сценария lpd из init.d с аргументом start, в результате чего начинает работу демон печати. Запустить этот сценарий можно из командной строки:

```
/etc/rc.d/init.d/lpd start
```

Именно в этом и состоит преимущество инициализации в стиле System V: пользователь root получает возможность запускать, останавливать и, в некоторых случаях, перезапускать и перезагружать произвольные демоны простым вызовом сценария из init.d с соответствующим аргументом. Выполнять это действие может только суперпользователь. Инициализация в стиле BSD не поддерживает подобного единого подхода к управлению различными демонами. Если в вашей системе используется стиль инициализации BSD, чтобы запустить или остановить тот или иной демон, вы должны знать, как именно этот демон управляется из командной строки, а это далеко не всегда ограничивается простым запуском исполняемого файла демона.

Если сценарий /etc/re.d/rc запускается без аргумента, он при помощи команды run level узнает текущий и предыдущий уровни выполнения и на основании этой информации определяет, какие сценарии и как ему следует запустить. Для этого он сравнивает содержимое каталогов, соответствующих предыдущему и текущему уровням выполнения, и, основываясь на различиях между ними, останавливает одни демоны и запускает другие. Говоря более подробно, сценарий rc сначала останавливает все те демоны, которые выполняются на момент его вызова и подлежат останову при переходе с предыдущего уровня выполнения на текущий, после чего стартует все те демоны, которые должны выполняться на текущем уровне, но сейчас не выполняются. Таким образом, сначала выполняются сценарии с префиксом К в порядке от младших номеров к старшим, а затем сценарии с префиксом S тоже от младших номеров к старшим. Благодаря этому запуск и остановка демонов осуществляются в нужном порядке. Например, прежде чем запускать sendmail (mta) или web-сервер Apache (httpd), следует запустить сетевую подсистему. И наоборот, прежде чем останавливать сетевую подсистему, сначала нужно остановить все зависящие от нее службы.

Такое поведение является отличительной чертой комплекта OpenLinux компании Caldera. В комплекте Red Hat и в других комплектах каталоги не сравниваются, вместо этого просто выполняются сценарии останова и запуска из каталога, соответствующего новому уровню выполнения.

Сценарии rc, часть вторая

Теперь, когда вы получили общее представление об инициализации системы, от ядра и до программы init, я расскажу вам о некоторых деталях этого процесса подробнее. В данном разделе главы я более

подробно расскажу о механизме запуска-останова демонов при смене уровней выполнения. Поскольку программа init запускается ядром, она обладает привилегиями суперпользователя. Таким образом, всякий процесс, запущенный из строк 36-42 листинга 7.1, будет запущен на уровне привилегий суперпользователя, если, конечно, он явно не был запущен от имени другого (непривилегированного) пользователя. Если злоумышленнику удастся вставить туда команду запуска программы, сохраняющей коды клавиш, нажимаемых в ответ на приглашение входа в систему, и время от времени отсылающей этот файл по электронной почте на указанный им адрес, то вскоре в распоряжении злоумышленника окажутся имена и пароли всех активных пользователей системы. Если пользователь сможет в одном из каталогов /etc/rc.d/rc[l-5].d/ создать ссылку на какой-нибудь посторонний файл, то во время инициализации системы этот файл может быть запущен на выполнение от имени суперпользователя.

ВНИМАНИЕ

Все файлы, расположенные в каталоге /etc/red/ и его подкаталогах и запускаемые в процессе инициализации системы, выполняются от имени суперпользователя (в дистрибутиве Debian это замечание относится к подкаталогам rc.boot/, rc[0-6].d/ и init.d/, расположенным непосредственно в каталоге /etc). Если это ссылка, то независимо от того, ведет ли она за пределы /etc/rc.d/ или нет, файл, на который она указывает, все равно выполняется от имени суперпользователя. Любые изменения в существующих сценариях или добавления новых могут вступить в силу при следующей перезагрузке или смене уровня выполнения. Вывод очевиден: правом на запись в файлы и каталоги, используемые при инициализации системы, должен обладать только суперпользователь и никто другой.

Далее идет очень короткий обзор некоторых ключевых сценариев, включая один сценарий запуска-останова демона. Обучение чтению сценариев не входит в задачи данного текста, поэтому при возникновении трудностей следует обратиться к книге, посвященной разработке сценариев командной оболочки. Первым будет рассмотрен сценарий /etc/rc.d/rc (см. листинг 7.3[5]). Номера строк в самом сценарии отсутствуют, они добавлены мною для удобства дальнейшего изложения.

Листинг 7.3. Сценарий re (номера строк добавлены искусственно) 1 #\/bin/bash 2#

```
3 # rc   This file is responsible for starting/stopping
4 #        services when the run!eve! changes.
5#
6 #       Temporary feature:
7 #   If the action for a particular feature in the new run-level
8 #   is the same as the action in the previous run-level, this
9 #   script will neither start nor start that feature, since that
10#   would have no effect except to thrash the system for no reason.
11#   Once all scripts are converted to use start-stop-daemon
12#   to _start_ their daemons (most of them only use it to kill
13#   them), this feature can be removed.
14#
15#   $Id: rc.v 1.7 1999/07/14 21:36:04 ray Exp $
161
17#   Author: Miquel van Smoorenburg, <miquels@drinkel .ow.org>
18#   Hacked to bits by Bruce Perens <Bruce@Pixar.com>
19#   Modified for COL by Raymund Will <ray@lst.de>
20#
21
22 export RC_DEBUG=false
23 export RC_VERBOSE=true
24 LOG=/dev/tty12
25
26 true() { return 0; }
27 false() { return 1; }
28 Echo() {
29    local a=$l; shift
30    local o=$l; shift
31    local i
32
33    echo -n "$a" >> $LOG
34    for i in "$@"; do
35   echo -n " '$i'" >> $LOG
36    done
37   echo "$o" >> $LOG
38   }
39
40    # check for new-style boot-logger
```

— [5] Комментарии в листинге оставлены без перевода, чтобы не нарушить нумерацию строк. Описание работы сценария см. далее в книге. — *Примеч. перев.*

```
41    export SVIBooter=/sbin/booter
42    [ -x $SVIBooter ] | SVIBooter=false
43
44  if $SVIBooter test; then
45        export SVIuseBooter=true
46        CMDS="add start"
47
48        # redirect STDOUT and STDERR
49        exec - >> $LOG 2>&1
50
51        #DEBUGGING
52        Booter() {
53           local c=$l; shift
54           local s
55           local i
56
57  case "$c" in
58   add)
59     s="$l"; shift
60   eval "$s" $c
61   ;;
62   start)
63              s="$l": shift
64              eval "$s" $c "$@"
65   case $? in
66     0) $SVIBooter ok;;
67     1) $SVIBooter fail;;
68     2) $SVIBooter skip;;
69     *) $SVIBooter "N/A" ;;
70   esac
71   ;;
72   stop)
73   s="$l"; shift
74   Echo "# Booter " "." "$s" $c "$@"
75     eval "$s" $c "$@"
76   ;;
77   *)
78     $SVIBooter $c "$@"
79   ;;
80  esac
81        }
82        [  -z "$PREVLEVEL" ] && {PREVLEVEL=N
83   else
84        SVIuseBooter=false
85        CMDS="start"
86
87        # Set onlcr to avoid staircase effect.
88        stty onlcr 0>&1
89
90   Booter() {
91           local c="$1"; shift
92           [ "$c"  != "start" -a "$c"  != "stop" ] && return 0
93           local s="$l"; shift
94           Echo "# eval " "." "$s" $c "$@"
95           eval "$s" $c "$@"
96   }
97   fi
98
99   # Now find out what the current and what the previous run!eve! are.
100
101    runlevel=$RUNLEVEL
102    # Get first argument. Set new run!eve! to this argument.
103    [ -n "$1" ] && runlevel=$l
104
105  previous=$PREVLEVEL 106
107     Echo "runlevel=$runlevel previous=$previous" "."
108     export runlevel previous
109
110    RCD=/etc/rc.d
111    # Is there an re directory for this new runlevel?
112  if [ -d "$RCD/rc$runlevel.d" ]; then
113                                        avoid="" # A list of start scripts I don't have to run.
114
```

```
    # First, run the KILL scripts.
    if [ "$previous" != N ]; then
for i in $RCD/rc$rumevel.d/K[0-9][0-9]*; do
          # Check if the script is there.
          [ -f "$i" ] || continue

       suffix=${i#$RCD/rc$runlevel .d/K[0-9][0-9]}

          # Generate the name of the start script corresponding
          # to this stop script, the start script in the previous
          # level, and the stop script in the previous level.
          # Check these files, and see if the previous level's
          # files are links to the ones for this level.
          # If they are, this level treats this feature the same
          # as the previous level, and I don't have to run these
          # files.
          stopIt=true
            start=$RCD/rc$runlevel.d/S[0-9][0-9]$suffix
            previous_start=$RCD/rc$previous.d/S[0-9][0-9]$suffix
            previous_stop=$RCD/rc$previous.d/K[0-9][0-9]$suffix

  if [ -f $previous_stop ] && [ $i -ef $previous_stop ]; then
              stopIt=false
              if [ -f $start ] || [ -f $previous_start ]: then
       if [ -f $start ] &&
                    [ -f $previous_start ] &&
                    [ $start -ef $previous_start ]; then
      stopIt=true
       else
          avoid=$avoid" "$start
       fi
    fi
  fi

          # Kill it.
          $stopIt && Booter stop $i
done
fi

  Booter list "RUNLEVEL Run-level change..."
  Booter add_menu "BLANK4"
  Booter add_menu "T4 Entering run-level $runlevel:"
  for cmd in $CMDS; do

    # Now run the START scripts for this runlevel.
    for i in $RCD/rc$runlevel.d/S*; do
      # Check if the script is there.
      [ -f "$i" ] || continue

      startIt=true
      case " $avoid " in
  *\ $i\ *) startIt=false;;
      esac
      if $startIt; then
          suffix=${i#$RCD/rc$runlevel.d/S[0-9][0-9]}
          previous_start=$RCD/rc$previous.d/S[0-9][0-9]$suffix
          stop=$RCD/rc$runlevel .d/K[0-9][0-9]$suffix
  if [ -f $previous_start ] &&
              [ $i  -ef $previous_start ] &&
              [ !  -f $stop ]; then
     startlt-false
  fi
fi

$startlt && Booter $cmd $i
done

if [ "$cmd" != "add" ]; then
Booter complete RUNLEVEL
else
       Booter end
       Booter activate RUNLEVEL
fi
done
```

```
189 fi
190
191 [ $SVIuseBooter = false ] && exit 0
192
193
194 if [ $runlevel  != 5 ]: then
195    sleep 1
196    /usr/bin/chvt 1
197 else
198    /sbin/booter list "FINAL"
199    /sbin/booter add_menu "BLANK5"
200    /sbin/booter add "KDE Starting KDE"
201    /sbin/booter end
202    /sbin/booter activate "FINAL"
203    /sbin/booter item "KDE"
204    sleep 1
205    ( trap "" SIGHUP
206       sleep 10
207       echo -e "\n\nPlease switch to a different virtual console for login!\n\n"
208 ) > /dev/tty7 &
209 fi
210
211Booter quit
212# eof /etc/rc.d/rc
```

Строки, начинающиеся с символа #, считаются комментариями (за исключением самой первой строки, в которой указывается программа, используемая для выполнения команд сценария) и при выполнении сценария игнорируются. В строках 2-117 происходит инициализация переменных, определяются текущий и предыдущий уровни выполнения, а также сравниваются сценарии запуска-останова из соответствующих этим уровням подкаталогов каталога /etc/rc.d.

В строках начиная с 119 и до конца файла сначала выполняются сценарии останова демонов, ненужных на новом уровне выполнения. После этого осуществляется выполнение сценариев запуска демонов, которые должны выполняться на новом уровне, но сейчас не выполняются.

Следующим сценарием, который мы рассмотрим, будет типичный сценарий запуска-останова демона. Множество подобных сценариев расположено в каталоге /etc/re.d/init.d. Большинство таких сценариев очень похожи на сценарий из листинга 7.4[6].

Листинг 7.4. Сценарий старта-останова демона named

```
1 #!/bin/sh
2 #
3 # named      This shell script takes care of starting and stopping
4 #              named (BIND DNS server).
5#
6
7  NAHE=named
8  DAEMON=/usr/sbin/$NAME
9
10   # Source function library.
11   . /etc/rc.d/init.d/functions
12
13# Source networking configuration.
14 . /etc/sysconfig/network
15
16# Check that networking is up.
17[ ${NETWORKING} = "no" ] && exit 0
18
19 [        -r /etc/sysconfig/daemons/$SUBSYS ] || ONBOOT=Yes
20 [        ! -r /etc/sysconfig/daemons/$SUBSYS ] || . /etc/sysconfig/daemons/$SUBSYS
21 [        "$ONBOOT" = "no" -a "$PROBABLY" = "booting" ] && exit 0
22
23 [ -x $DAEMON ] || exit 0
24
25
26# See how we were called.
27 case "$1" in
28   start)
29   [ -e /var/lock/subsys/$SUBSYS ] && exit 1
30
```

[6] Комментарии в листинге оставлены без перевода, чтобы не нарушить нумерацию строк. Подробное описание работы сценария см. далее в книге. — *Примеч. перев.*

```
31  [ -f /etc/named.conf ] |] exit 0
32
33    # Start daemons.
34    echo -n "Starting BIND DNS server: "
35    start-stop-daemon -S -n $NAME -x $DAEMON -- $OPTIONS
36    echo "."
37    touch /var/lock/subsys/$SUBSYS
38    ;;
39
40  stop)
41  [ -e /var/lock/subsys/$SUBSYS ] || exit 0
42
43    # Stop daemons.
44    echo -n "Stopping BIND DNS server: "
45    start-stop-daemon -K -p /var/run/$NAME.pid -n $NAME
46    echo "."
47    rm -f /var/lock/subsys/$SUBSYS
48    ;;
49
50  restart)
51  [ -e /var/lock/subsys/$SUBSYS ] || exit 0
52
53    echo -n "Re-starting BIND DNS server: "
54    start-stop-daemon -K -s 1 -p /var/run/$NAME.pid -n $NAHE
55    echo "."
56    ;;
57
58  *)
59    echo "Usage: named {start|stop|restart}"
60    exit 1
61  esac
62
63   exit 0
```

Как и ранее, строки, начинающиеся с символа #, адресуются читающему сценарий пользователю и при выполнении игнорируются.

В строках 7 и 8 инициализируются некоторые переменные, а строки 11 и 14 включают в исходный текст сценария некоторые глобальные переменные из конфигурационного каталога системы (такой каталог присутствует далеко не во всех комплектах Linux). Строка 17 проверяет, запущена ли сетевая подсистема, и если нет и в случае, если сеть не работает, прекращает выполнение сценария (бессмысленно запускать DNS-сервер в отсутствие сети). Строка 23 позволяет убедиться в том, что файл демона существует и является исполняемым.

После этого выполняется обработка аргумента, переданного в сценарий. Если использован аргумент start, выполняются строки 29-38. Если использован аргумент stop и демон запущен (это проверяется в строке 41), то выполняются строки 44-48. Если аргументом является restart и демон запущен (это проверяется в строке 51), то выполняются строки 53-56, посылающие ему сигнал SIGHUP (1). Этот сигнал предписывает демону заново прочесть свой конфигурационный файл и продолжить работу.

Наконец, если аргумент не был распознан или попросту отсутствует, выдается подсказка по использованию сценария.

Примерно так же работают и все остальные сценарии старта-останова в стиле System V. Детали зависят от дистрибутива, но общий подход везде один и тот же. Инициализация в стиле BSD происходит несколько по-другому: вместо многих небольших сценариев в этой системе используется несколько больших и отсутствует механизм start/stop/restart.

Заключение

В этой главе была рассмотрена инициализация в стиле System V. И хотя изложение было ориентировано на систему Caldera Open Linux 2.3, представленный в ней материал применим и ко многим другим комплектам Linux, использующим инициализацию в стиле System V. Также был затронут круг вопросов, связанных с начальной загрузкой системы. Были рассмотрены уровни выполнения, которые являются средством организации демонов в группы. Вы также узнали об использовании сценариев инициализации для запуска/останова демонов при переходе с одного уровня выполнения на другой.

Кроме того, из этой главы явствует, что если злоумышленнику удастся подменить сценарий инициализации или изменить уже существующий, благодаря этому во время инициализации ОС он сможет

запустить любую программу с привилегиями суперпользователя.

8 Физическая безопасность и консольные атаки

В данной главе рассматриваются следующие вопросы:

- уязвимые места системы, которыми можно воспользоваться до загрузки ядра;
- загрузчик LILO;
- параметры загрузки на непредвиденный случай;
- восстановление позабытого пароля учетной записи root;
- резервное копирование системы;
- защита вашей сети.

Прочитав предыдущую главу, вы, должно быть, уже получили представление о том, какими элементами системы может воспользоваться злоумышленник, желающий взломать систему и обладающий физическим доступом к взламываемому компьютеру. В данной главе будет продемонстрировано, как именно реализуется подобный взлом (если вы не закрыли пару-другую пробелов в вашей системе безопасности). Узнав о том, с какой простотой этого можно добиться, вы поймете, почему можно считать, что каждый, обладающий физическим доступом к компьютеру, фактически «владеет» им.

До загрузки ядра

Система беззащитна перед консольными атаками в любое время, даже до того, как произойдет ее начальная загрузка. Все что нужно для организации подобной атаки — это перезагрузить систему. Перезапуск можно выполнить либо при помощи обычной команды shutdown, либо щелкнув BRS (BRS — Big Red Switch — большой красный переключатель, то есть переключатель подачи электропитания). Если вы просто выдернете шнур электропитания из розетки, вы получите тот же самый результат. Таким образом, выполнить принудительную остановку работающей системы совсем несложно. Как только система прекратила свою работу, вы можете загрузиться либо с гибкого диска, либо с компакт-диска CD-ROM.

Да, да, конечно, вы можете войти в BIOS и настроить компьютер так, чтобы единственным допустимым загрузочным диском был диск C: (/dev/hdc), то есть для загрузки будет использоваться только главная загрузочная запись MBR. После этого, чтобы предотвратить изменение этого параметра, вы можете защитить BIOS при помощи пароля. Теперь для того, чтобы войти в систему, злоумышленник должен обладать паролем BIOS, так как иначе он не сможет войти в BIOS и, стало быть, не сможет загрузить систему так, как ему это нужно.

ПРИМЕЧАНИЕ

Настройка пароля для BIOS — это не то же самое, что настройка пароля для суперпользователя. Возможно, для вас будет удобным сделать пароль BIOS таким же, как и пароль пользователя root (суперпользователя), так как пароль BIOS используется чрезвычайно редко, поэтому его очень легко позабыть. Однако если вы решили использовать подобную политику, никогда не забывайте менять пароль BIOS синхронно со сменой пароля root, иначе вы легко можете запутаться в паролях.

Если спустя несколько месяцев (или лет) вы обнаруживаете, что забыли пароль для входа в BIOS, не волнуйтесь. Вы можете обратиться к документации (если она у вас, конечно, есть) установленной в вашем компьютере материнской платы для того, чтобы узнать, как можно сбросить содержимое CMOS (то есть энергонезависимой памяти, в которой хранятся значения параметров BIOS). Как правило, для этого требуется замкнуть специальный установленный на материнской плате переключатель. В результате содержимое CMOS устанавливается равным значениям по умолчанию, при этом материнская плата забывает о существовании пароля BIOS и вы сможете беспрепятственно проникнуть в BIOS, чтобы заново настроить CMOS. Эта возможность поддерживается большинством современных материнских плат. Однако некоторые достаточно старые модели такой возможности не поддерживают. В этом случае вам придется отключить систему от питания, отключить от материнской платы специальную батарейку CMOS

и подождать около 20 минут (30 минут, чтобы уж точно не ошибиться). При этом значения параметров CMOS (где, собственно, и хранится пароль) станут равными значениям по умолчанию. Это означает, что все остальные значения аппаратной конфигурации, в соответствии с которыми система была настроена ранее, будут утеряны. Вам придется заново настраивать BIOS. Однако здесь важно обратить внимание на то, что если вы можете проделать такую процедуру с содержимым CMOS, значит, это может сделать и любой другой, кто обладает физическим доступом к компьютеру. Иными словами, пароль BIOS не является достаточно надежной защитой вашей системы.

СОВЕТ

На случай потери содержимого CMOS я рекомендую вам сохранить значения параметров BIOS в надежном месте. Вы можете переписать их на бумагу или сохранить где-либо в электронном виде, например, в виде снимков различных экранов BIOS. Имейте в виду, что содержимое CMOS может быть утеряно не только в результате необходимости сбросить пароль BIOS. Причиной утери содержимого CMOS может стать севшая батарейка, это может произойти в случае, если вы отключите систему от электропитания на достаточно длительное время.

LILO

После того как компьютер завершает процедуру начального тестирования POST, он приступает к поиску операционной системы, которую следует загрузить. Если вы настроили BIOS таким образом, чтобы компьютер мог загрузиться только с диска C:, происходит обращение к главной загрузочной записи MBR этого диска. Из MBR в оперативную память компьютера загружается исполняемый код начального загрузчика. В данной книге я предполагаю, что Linux является единственной операционной системой, установленной на компьютере, а в качестве загрузчика используется LILO, являющийся стандартным загрузчиком в среде Linux. Начальные загрузчики большинства современных операционных систем работают аналогичным образом и используют сходный набор параметров. В частности, если вы имеете дело с системой, оснащенной несколькими ОС (например, Linux и Windows, Solaris, BeOS, OS/2 и т. п.), как правило, у вас есть возможность выбрать одну из установленных систем для того, чтобы загрузить ее в память компьютера. Выбор осуществляется с использованием LILO или другого аналогичного менеджера начальной загрузки. Более подробно об этом вы можете узнать из инструкций, прилагаемых к используемому вами менеджеру загрузки. Подробнее о LILO можно узнать в любом хорошем руководстве по администрированию системы Linux.

По умолчанию, если в MBR содержится LILO, значит, в глобальном разделе файла /etc/lilo.conf содержится пара стандартных строчек:

```
prompt
timeout= 50
```

Эти параметры предназначены для осуществления двух функций: первая из них обеспечивает вам возможность выбора между несколькими образами загрузки (ядрами ОС), вторая обеспечивает временную задержку для осуществления выбора. Первый параметр lilo.conf (параметр prompt) указывает на то, что в процессе начальной загрузки менеджер загрузки LILO должен отобразить на экране компьютера приглашение LILO:. Если параметр prompt не указывается в файле lilo.conf,

значит, никакого приглашения на экран не выводится. Второй параметр сообщает LILO о том, какое количество времени (в десятых долях секунды) следует подождать, прежде чем приступить к загрузке ядра ОС по умолчанию. Благодаря этому осуществляется поддержка автоматической загрузки ОС на вашем компьютере. Иными словами, чтобы загрузить ОС по умолчанию, вы можете включить компьютер и больше не предпринимать никаких действий — LILO предложит вам выбрать одну из ОС; если вы не воспользуетесь этой возможностью и никак не прореагируете на предложение, LILO подождет в течение указанного времени (значение параметра timeout в рассмотренном ранее примере соответствует пяти секундам), а затем приступит к загрузке ядра ОС по умолчанию.

ВНИМАНИЕ

Если в файле /etc/lilo.conf вы используете параметр prompt и при этом забыли добавить (или удалить) параметр timeout=, ваша система не будет обладать возможностью загрузки без участия пользователя.

Если вы удалите из файла lilo.conf обе строки, система всегда будет напрямую загружать ядро ОС по умолчанию. Вы не сможете выбирать, какую из ОС вам хотелось бы загрузить, кроме того, вы не сможете

передавать ядру ОС какие-либо параметры загрузки. В этом случае все необходимые параметры загрузки должны располагаться в разделе image файла lilo.conf.

ВНИМАНИЕ

Если загрузчик LILO настроен на прямую загрузку ядра ОС по умолчанию и если вы меняете ядро, вы должны либо перенастроить LJLO, либо создать загрузочный диск со старым ядром, готовым к загрузке. Этот диск следует держать наготове до тех пор, пока вы не убедитесь в том, что новое ядро загружается без каких-либо проблем и сбоев.

Если удаление обеих этих строк невозможно (как правило, из-за того, что на вашем компьютере установлена еще одна операционная система, будь то Linux или любая другая ОС), вы можете защитить паролем загрузку любого из образов ОС. Настройка пароля выполняется для каждого из ядер по отдельности.

ПРИМЕЧАНИЕ-

По умолчанию файл /etc/lilo.conf может прочитать кто угодно, поэтому если вы намерены защитить паролями загрузку каких-либо ядер ОС, скорее всего, будет разумным выполнить в отношении этого файла команду chmod 600. Существует еще один вариант: после запуска LILO вы можете переместить файл /etc/lilo.conf на гибкий диск, так как он не требуется для того, чтобы продолжить загрузку системы. Помните, что если ядро ОС по умолчанию (которое указывается в lilo.conf в самой первой позиции) защищено паролем, система не сможет выполнить автоматическую загрузку без участия пользователя, в связи с этим, возможно, будет удобнее защитить паролем только те ядра, которые не являются ядрами ОС по умолчанию.

Параметры загрузки для непредвиденных ситуаций

Иногда в результате даже самой аккуратной переделки ядра или тщательно проверенного редактирования инициализационных файлов нечто препятствует корректному завершению процедуры инициализации. Конечно же, если вы редактируете файл inittab или любой из сценариев rc, вы делаете это с большой осторожностью. Вы тщательно продумываете каждое вносимое вами изменение, а также выполняете тестирование. Однако даже самые лучшие тесты не могут полноценно имитировать реальную загрузку системы, и сценарий, который выглядит вполне корректным и работоспособным, в процессе реального запуска системы может не выполниться или, того хуже, привести к зависанию системы. Причинами могут быть самые разные факторы, однако чаще всего неприятности возникают из-за того, что определенные действия выполняются в неправильной последовательности.

Для примера рассмотрим сценарий, который используется для запуска процесса kerneld на начальных стадиях загрузки системы, которая использует ядро версии 1.2.13 с модулями. После обновления ядра системы до версии 2.0.25 я решил использовать тот же самый сценарий. К сожалению, в результате выполнения этого сценария в момент загрузки нового модуля kerneld, который работает совместно с новой версией ядра, система зависает. После непродолжительных исследований я обнаружил, что новая версия процесса kerneld должна знать имя узла (hostname) системы, однако на момент загрузки kerneld имя узла системе еще не известно. Чтобы решить проблему, необходимо просто запустить процесс определения имени узла раньше, чем загружается kerneld (то есть в файле /etc/rc.d/rc.boot). Описанная история произошла лично со мной, однако она могла произойти с кем угодно. Подчас достаточно позабыть добавить нужный ключ или упустить полный путь к исполняемому файлу, и отредактированный вами сценарий перестает корректно работать.

СОВЕТ

Если вам кажется, что система зависла, это вовсе не значит, что она действительно зависла. Прежде чем паниковать, жать на <Reset> или кнопку отключения электропитания, как минимум, следует подождать, пока истечет IP-тайм-аут. Как правило, длительность этого тайм-аута составляет 2 минуты, но он может длиться и несколько дольше. Если вы удалите строку vga=274, вы сможете увидеть имя каждого из сценариев rc по мере того, как они будут выполняться. Обратите внимание на то, какой из них был запущен последним. Возможно, именно он является причиной проблемы. В дальнейшем для отладки вы должны в первую очередь обратиться именно к этому сценарию.

К счастью, если в файле lilo.conf вы используете строки prompt и timeout, вы сможете передавать программе init параметры. Когда система загружается и вы видите на экране приглашение LILO:, вы можете нажать клавишу <Shift>, а затем клавишу <TaЬ> для того, чтобы увидеть набор меток ядра, доступных для загрузки. После этого вы можете набрать на клавиатуре одну из меток, а за ней указать любой набор параметров, который вы желаете использовать для загрузки системы. Любые используемые

ядром параметры извлекаются из введенной вами строки, оставшиеся в строке параметры передаются далее по цепочке программе init. Например, если ваш компьютер оснащен оперативной памятью объемом 128 Мбайт и при этом вы желаете, чтобы из этой памяти использовались только первые 96 Мбайт, вы можете указать параметр запуска в виде: mem=96MB (или любой другой объем RAM). Этот параметр будет воспринят и обработан ядром. Однако если вы укажете параметр -Ь, ядро не будет его обрабатывать и передаст этот параметр программе init. Это относится к любому однозначному числу или буквам s или q либо в верхнем, либо в нижнем регистре.

СОВЕТ

Параметр - Ь используется для запуска системы в режиме обслуживания, то есть без выполнения каких-либо сценариев rc. Это очень удобно в случае, если вы подозреваете, что выполнение одного из этих сценариев нарушает корректную работу системы.

Передав любой из допустимых номеров или букв, обозначающих один из уровней запуска, вы отменяете действие значений по умолчанию, определенных в файле inittab. Большая часть этих букв или цифр делают то, что полагается, если они передаются из командной строки работающей системы. Однако аргумент -b выполняет особую функцию. Этот параметр является специальным параметром на случай сбоев. Если вы передаете программе init параметр -b, эта программа выполняет чтение файла inittab, однако при этом ни один из сценариев rc не выполняется. Этот параметр также заставляет систему перейти на уровень запуска 1 (режим обслуживания). Таким образом, сценарии rc не запускаются. В результате вы получаете возможность монтировать систему для чтения/записи и исправить ее. Однако даже при использовании ключа -b некоторые строки файла inittab все же обрабатываются. Такими строками являются строки с идентификаторами (id), начинающимися с символа ~, например ~~ или ~1. Если вы намерены добавить в initttab команду запуска некоторого сценария и присвоить соответствующей строке идентификатор, начинающийся с символа ~, будет лучше проверить этот сценарий с каким-либо другим id, и лишь убедившись в его работоспособности, назначить ему id, начинающийся с ~. В страницах электронной документации man вы не найдете описания этой возможности, так как она не является документированной.

Взгляните на листинг 7.1 в главе 7. В этом листинге присутствуют следующие строки:

```
~1:S:wait:/etc/rc.d/rc   1 ~~:S:wait:/sbin/sulogin
```

Вне зависимости от того, используете ли вы ключ -b или нет, эти две строки файла inittab будут исполнены. Вторая строка заставляет систему отобразить приглашение на ввод пароля учетной записи root. И все же это не обеспечивает полной безопасности.

ВНИМАНИЕ

В следующем абзаце рассказывается о возможности загрузки непосредственно в командную оболочку. При выполнении этой процедуры могут возникнуть необратимые повреждения файловой системы, поэтому данную возможность следует использовать только в самом крайнем случае!

Если же несмотря на все ваши усилия вы обнаруживаете, что даже при передаче загрузчику LILO ключа -b система подвисает в процессе загрузки или вы просто забыли пароль учетной записи root, не отчаивайтесь. В подобной ситуации после появления на экране приглашения LILO: вы можете передать загрузчику параметр init=/bin/sh. Этот аргумент используется ядром. Если вы помните, прежде чем приступить к работе в фоновом режиме, ядро запускает одну и только одну программу: init. Если же вы передадите ядру указанный ранее аргумент, вместо init ядро запустит исполняемый файл командной оболочки. Эта процедура чрезвычайно опасна, однако благодаря этому вы сможете смонтировать корень файловой системы в режиме для чтения и записи (mount -n -o remountrw /), отредактировать /etc/inittab или /etc/shadow, после этого выполнить синхронизацию при помощи sync и перезагрузить систему. В данном случае, прежде чем выполнять перезагрузку, я рекомендую запустить sync дважды. При обращении к sync «грязные» буферы будут сброшены на диск, благодаря чему вы можете быть уверены в том, что любые внесенные вами изменения действительно сохранены на диске. Данный метод исправления важных инициализационных файлов должен использоваться только в самом крайнем случае, когда у вас не остается больше никакой другой альтернативы, кроме переустановки системы. Дело в том, что в ходе выполнения этой процедуры вы можете нарушить целостность вашей файловой системы и таким образом сделать переустановку ОС неизбежной.

Восстановление пароля root

Прежде чем вы попытаетесь сделать то, о чем только что говорилось, вы должны попытаться воспользоваться загрузочным гибким диском Caldera OpenLinux (или загрузочным диском, входящим в ваш комплект поставки Linux) или установочным компакт-диском (большинство таких дисков являются загрузочными) и попытаться загрузить систему. Как только перед вами появится экран, на котором вам предлагается выбрать диск, на который вы хотите установить Linux, остановитесь. Если вы продолжите, будет запущена программа fdisk, а это не то, что вам нужно. Как правило, при помощи загрузочного диска вы можете выполнить загрузку ОС, а затем, не продолжая установку, перейти в режим командной строки. В разных вариантах поставок Linux для этой цели требуется выполнить разные действия, однако, как правило, все они похожи. Вы должны свериться с руководством, которое входит в используемый вами комплект поставки.

В частности, если вы используете Caldera OpenLinux, на экране, предлагающем вам выбрать диск для установки, вы должны нажать комбинацию клавиш <Ctrl>+<Alt>+<F2>, после чего следует подключиться к системе с использованием учетной записи root (при этом вам не потребуется вводить пароль). Как и другие разновидности поставки, Caldera OpenLinux использует виртуальный диск в оперативной памяти (ramdisk), который представляет собой полную корневую файловую систему, в которую входит подкаталог с именем target. Вы можете смонтировать вашу старую корневую файловую систему в каталоге target, перейти в подкаталог target/etc (cd target/etc) и затем отредактировать теневой файл паролей. Если вы забыли пароль, вы должны просто удалить хэшированный пароль из файла (вы не должны оставить между двумя двоеточиями ни одного символа, даже пробела). После этого перейдите обратно в корневой каталог (cd /), размонтируйте файловую систему, которую вы смонтировали в каталоге target (это очень важно), а затем перезагрузите систему как обычно, то есть с использованием вашей старой файловой системы. В результате вы сможете в систему с использованием учетной записи root, не сообщая при этом пароля.

ВНИМАНИЕ

Без пароля учетной записи root ваша система является совершенно беззащитной. Выполняя описанную процедуру, вы должны отключиться от сети (просто выдерните кабель Ethernet из разъема сетевой карты), а войдя в систему, в самую первую очередь немедленно восстановите пароль учетной записи root.

Не забывайте о том, что любой человек, обладающий физическим доступом к вашему компьютеру, может проделать все, о чем здесь рассказывалось.

Резервное копирование

Наверное, наиболее недооцененной областью системной безопасности является резервное копирование данных. Просматривая любые инструкции и руководства, связанные с Linux, вы очень часто будете сталкиваться с фразой: «Наверняка у вас уже есть свежая резервная копия данных, не так ли?». Резервное копирование рекомендуется делать всем, даже домашним пользователям. Однако далеко не у всех есть устройство записи информации на магнитную ленту (стриммер). Хороший стриммер до сих пор является, наверное, самой дорогостоящей частью системы. Зачастую такие устройства используются только на коммерческих предприятиях.

У многих из вас может возникнуть желание использовать для резервного копирования диски Zip, сменные жесткие диски или даже записываемые компакт-диски. Все эти варианты вполне могут применяться для хранения архивов, однако вы должны знать о некоторых весьма серьезных недостатках всех этих носителей по сравнению с магнитной лентой. В частности, диски Zip на самом деле недостаточно хорошо защищены от ошибок. Они не обладают столь же надежными механизмами обнаружения и коррекции ошибок, которые используются в жестких дисках. Если вы намерены использовать для резервного копирования Zip-диск, будьте готовы к потерям данных. Использование сменных жестких дисков может показаться весьма приемлемым вариантом, однако вы должны учитывать, что сменные жесткие диски столь же беззащитны перед скачками напряжения, как и обычные жесткие диски. Таким образом, при неудачном стечении обстоятельств вы можете потерять данные как на своем основном жестком диске, так и на сменном жестком диске, на котором хранилась резервная копия. Наконец, записываемые компакт-диски обладают невысокой скоростью записи, кроме того, их емкость не может быть более 660 Мбайт.

Существуют самые разнообразные рекомендации относительно того, как следует выполнять резервное копирование. Вы можете каждый вечер создавать полную резервную копию всей системы, вы можете делать это один раз в неделю и при этом ежедневно или три раза в неделю выполнять *добавочное* (incremental) резервное копирование. Многие компании организуют хранение еженедельных или

ежемесячных резервных копий на стороне, то есть вне здания фирмы, в которой функционирует система. Таким образом, ленты, на которых хранятся резервные копии, размещаются как в здании фирмы, так и в другом здании, благодаря чему надежность хранения данных возрастает. Для этой цели вы должны выбрать компанию с хорошей репутацией и хранить свои резервные копии так, чтобы никто кроме вас не обладал бы физическим доступом к вашим лентам резервного копирования.

К сожалению, во многих местах, где мне приходилось бывать, ленты резервного копирования зачастую забываются вставленными в стримеры или хранятся так, что любой может получить к ним доступ. На самом деле физическим доступом к лентам резервного копирования должны обладать только те люди, которые по долгу службы должны знать пароль учетной записи root. На деле ленты резервного копирования должны защищаться от постороннего доступа даже лучше, чем сами системы.

Ленты резервного копирования, в особенности те из них, которые хранят на себе полную резервную копию всей системы, содержат полный образ системы. Их можно использовать для того, чтобы в точности воссоздать систему. Фактически хорошая резервная копия системы — это даже лучше, чем возможность физического доступа к системе. Дело в том, что содержимое ленты можно загрузить на любой другой системе. Для этого достаточно знать название утилиты, при помощи которой была создана данная резервная копия (на самом деле многие графические коммерческие утилиты резервного копирования просто-напросто выполняют функции интерфейса для упрощения работы с традиционными утилитами резервного копирования Linux, такими как tar, cpio или dump).

Если злоумышленник получает возможность загрузить на какую-либо другую систему всего один файл /etc/shadow, значит, он получает доступ не только к тем сведениям, которые хранятся в системе в данный момент (все эти сведения и так записаны на ленту с резервной копией), но и к тем данным, которые появятся в системе в будущем. В этом случае все пароли, включая пароль root, следует считать раскрытыми. Загрузив файл теневых паролей на свой компьютер, злоумышленник может не спеша использовать в отношении этого файла любые программы автоматического взлома. В скором времени ему станет известно большинство паролей. Однако наиболее важным будет пароль root.

СОВЕТ

Если вы подозреваете, что содержимое резервной копии стало доступным для неавторизированного лица либо в результате кражи, либо в результате несанкционированного копирования, вы должны сменить все пароли на данной системе, кроме того, все системы, которые доверяют данной системе, должны рассматриваться как уязвимые.

Охрана сети Linux

Исходя из всего вышесказанного становится очевидно, что разные системы должны защищаться по-разному. О защите сетей и уязвимых местах сетевой архитектуры Linux будет рассказано в следующей части книги, здесь же мы остановимся на физическом доступе к клиентским системам.

Если вы имеете дело с сетью любых размеров, скорее всего, у вас в сети существует несколько различных категорий систем. В частности, у вас наверняка есть клиентские системы, с которыми работает большая часть пользователей. В сети существует также сервер (или два), который хранит на себе пользовательские файлы и обеспечивает доступ к сетевым службам печати (как правило, домашние каталоги всех пользователей при помощи NFS монтируются в рамках клиентских файловых систем, то же самое происходит с электронной почтой и другими данными). Если сеть обладает подключением к Интернету, скорее всего, в ней установлена система, выполняющая функции брандмауэра, а также, возможно, функции маскировки внутренних IP-адресов (IP-masquerading).

Давайте рассмотрим некоторые уязвимые места, присущие клиентам Linux, работающим в вашей сети, и их влияние на защиту серверов.

В зависимости от используемой сетевой модели на клиентах Linux могут существовать учетные записи различных видов. Как правило, на клиентских машинах размещается существенно сокращенный файл /etc/password. Однако вне зависимости от того, используете ли вы NIS, клиенты с полными запаролеными учетными записями или xdm, благодаря чему клиенты подключаются напрямую к серверу, файл /etc/shadow, размещенный на клиентских машинах, следует считать уязвимым местом. Это означает, что вы *обязаны* включить в него учетную запись root в самой первой позиции, однако пароль этой учетной записи на клиентских машинах должен отличаться от пароля учетной записи root на сервере, который в свою очередь должен отличаться от пароля учетной записи root на системе, выполняющей функции брандмауэра, который опять же должен отличаться от пароля учетной записи root на любой из машин, напрямую подключенных к Интернету.

При этом вы также должны убедиться в том, что клиенты доверяют серверам, но при этом серверы не доверяют клиентам. Не существует никаких причин, по которым сервер должен доверять клиентам. Необходимо также всегда рассматривать клиентские системы как наиболее уязвимые элементы сети.

Заключение

Данная глава завершает собой дискуссию о загрузке операционной системы, которая была начата в предыдущей главе. Ознакомившись с этим материалом, вы, должно быть, поняли, насколько уязвимой является система в процессе начальной загрузки. Вы узнали о том, что любой знающий человек может использовать перезагрузку системы для того, чтобы получить доступ ко всем элементам системы, даже тем, которые доступны только для учетной записи root, в том числе и самой учетной записи root.

Данная глава затрагивает вопросы, связанные с безопасностью не только самой системы, но и безопасностью резервных копий. Как вы узнаете позднее, чем больший объем информации вы предоставляете кому-либо о системе, тем проще для этого человека получить полный доступ к этой системе.

Часть II
Ваша сеть

9

Основные сведения о сети

В данной главе рассматриваются следующие вопросы:
- основы функционирования сетей;
- протокол IP;
- протокол ICMP;
- сетевая маршрутизация;
- технология CIDR;
- маршрутизация IP;
- внедрение CIDR с использованием VLSM;
- использование ifconfig; - использование route.

Во второй части книги вы узнаете о том, как устроена сеть и какие в ней есть уязвимые места. Необходимо учитывать, что большинство атак будет осуществляться через сеть, так как вряд ли ваша система установлена прямо на многолюдной улице, где каждый желающий может получить к ней доступ. Если ваша система установлена в публично доступном месте, прочитайте главу 8 для того, чтобы узнать об уязвимых местах, связанных с консольными атаками.

Несколько лет назад для того, чтобы атаковать систему с некоторой реальной вероятностью получить к ней доступ, вы должны были быть профессиональным опытным системным инженером, действительно разбирающимся в том, как работает система. Со временем количество системных инженеров росло, и среди них все чаще стали появляться люди с искаженными этическими представлениями. Эти люди стали заниматься разработкой программ, специально предназначенных для автоматизированного взлома систем с использованием существующих слабых мест в защите. С распространением Интернета подобные программы стали свободно доступны для всех, кто обладает доступом к Интернету. Многие подобные программы можно легко найти на http://www.rootshell.com/ и других аналогичных web-узлах. Официально они предназначены для инженеров и сетевых администраторов, однако загрузить их может любой желающий. Таким образом, инструменты, предназначенные для взлома системы безопасности, попадают в руки людей, которые не являются ни программистами, ни компьютерными инженерами, у многих из них представления об этических нормах весьма сомнительны.

Среди этих людей множество весьма устоявшихся категорий. В частности, многие из них — это подростки с огромным количеством свободного времени и тотальным отсутствием уважения к чему-либо, в особенности к взрослым. Все они уверены в том, что действуют в рамках Интернета абсолютно анонимно. Любой подобный молокосос, который подчас даже плохо понимает, что он делает, способен стать причиной головной боли для профессиональных системных администраторов. В последнее время подобные отбросы общества стали настоящим бедствием для всего Интернета. Средства массовой информации и некомпетентные в компьютерах люди ошибочно называют их «хакерами» (hackers), хотя на самом деле этот сброд зачастую даже не умеет программировать. Эти люди лишь хотят выглядеть в чужих глазах крутыми компьютерными взломщиками и нарушителями общепринятого порядка, фактически в их деятельности нет ничего конструктивного, что можно было бы обозначить термином «хакинг» (hacking).

В данной главе речь пойдет об основах, на которых базируется рабочая среда, в которой действуют все эти люди. В глубоком понимании принципов работы сети нет надобности, однако знакомство с

наиболее основополагающими принципами необходимо. Я расскажу вам об устройстве технологии IPv4 (Internet Protocol version 4), на которой базируется весь современный Интернет. Эту технологию для краткости зачастую называют просто IP. Ядро Linux поддерживает также и другие протоколы, включая IPv6 — дальнейшее развитие IPv4, которое до сих пор находится в стадии экспериментирования. На момент написания данной книги IPv6 все еще не получил широкого распространения.

Основополагающие понятия

Для того чтобы понять, как работает сеть, необходимо начать с простого, а затем постепенно переходить к более сложному. Абсолютно любые данные внутри компьютера представляются в виде комбинаций нулей и единиц. Два состояния — «ноль» и «единица», «включено» и «выключено», «установлено» и «сброшено» — описывают абсолютно все, о чем «знает» компьютер. Однако используя только одно подобное значение (либо 0, либо 1), невозможно описать все, что нужно.

ПРИМЕЧАНИЕ

Единственное цифроместо, которое может принимать только одно из двух значений (1 или 0), называется битом. Можно сказать, что бит — это переменная, которой можно присвоить одно из двух возможных значений. Это определение является фундаментальным для всей остальной последующей дискуссии.

Бит представляет собой лишь одно двоичное знакоместо. Любая более сложная информация кодируется комбинацией из нескольких битов. Группа из восьми битов называется *байтом* или *октетом.* Не следует путать биты и байты. К сожалению, это часто происходит, так как два этих термина обозначаются очень похоже и используются в очень похожих ситуациях. Например, когда говорят о скорости работы модемов, как правило, имеют в виду количество битов, передаваемых модемом в секунду. В частности, зачастую приходится слышать о том, что модем поддерживает скорость передачи данных 56 Кбит/с. Следует иметь в виду, что эта цифра обозначает количество бит, но не байт.

В большинстве других ситуаций, как правило, приходится иметь дело с байтами. Один октет (8 бит) может содержать в себе десятичное число от 0 до 255. Это следует из того, что октет состоит из восьми знакомест, каждое из которых может быть либо нулем, либо единицей. Какое количество различных комбинаций нулей и единиц может быть представлено одним октетом? Два в степени восемь, то есть 256 комбинаций. Таким образом, при помощи одного октета можно закодировать 256 различных чисел. Если использовать шестнадцатеричную систему счисления, получим, что одним октетом можно закодировать числа от 00 до FF. На всякий случай напомню, что такое шестнадцатеричная система счисления. В двоичной системе счисления каждый разряд может принимать одно из двух значений: либо 0, либо 1; в десятичной системе каждый разряд числа может принимать одно из десяти значений: от 0 до 9; наконец, в шестнадцатеричной системе каждый разряд может принимать одно из шестнадцати значений: от 0 до 9 и дальше A, B, C, D, E, F. То есть в шестнадцатеричной системе буквой F обозначается десятичное число 15, а шестнадцатеричное число 10 равно десятичному числу 16. В табл. 9.1 показаны первые семнадцать шестнадцатеричных чисел и их десятичные эквиваленты.

Таблица 9.1. Соответствие десятичных и шестнадцатеричных чисел

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10

ПРИМЕЧАНИЕ

Для обозначения шестнадцатеричных чисел в начале такого числа, как правило, ставится префикс Ox, например, число OxF равно десятичному 15. В данной книге это соглашение будет применяться там, где это необходимо для ясности, однако читатель и без префикса может понять, что число FF — это шестнадцатеричное число, равное десятичному 255.

Таким образом, следует отметить, что для удобства людей все числа представляются в десятичной системе счисления, в то время как внутри компьютера они преобразуются в двоичную систему. Шестнадцатеричная система используется для упрощенного представления многоразрядных двоичных чисел. Большинство программ предполагают ввод чисел либо в десятичной, либо в шестнадцатеричной системе счисления. Выбор зависит от общепринятых стандартов, а также от автора-разработчика этой программы. В данной книге я стараюсь сделать так, чтобы читатель всегда знал, какая именно система используется для представления того или иного числа.

Базовые сведения об IP

Протокол IP (Internet Protocol), являющийся основным протоколом для передачи данных через Интернет, описывается в RFC 791. IP — это базовый протокол, который используется для передачи через сеть других протоколов, таких как TCP и UDP. Иными словами, каждый пакет IP является как бы конвертом, внутрь которого вкладывается пакет TCP, UDP или другого протокола. Заголовок пакета IP включает в себя всю информацию, необходимую для того, чтобы передать содержащиеся внутри пакета данные из одного места сети в другое место сети. IP является наиболее часто используемым в Интернете протоколом, однако этот протокол далеко не единственный. Наряду с IP через сетевые каналы могут передаваться пакеты протоколов RIP (Routing Information Protocol), BGP (Border Gateway Protocol), ICMP (Internet Control Message Protocol), GGP (Gateway-to-Gatewaye Protocol) и ARP (Address Resolution Protocol). За исключением ICMP и ARP, остальные протоколы являются протоколами маршрутизации. Данная книга не обсуждает подробно вопросы, связанные с маршрутизацией, поэтому связанные с этим протоколы обсуждаться не будут. Однако помимо IP чуть позже мы будем обсуждать служебные протоколы ICMP и ARP.

Каким же образом работает сеть? Когда некоторое приложение желает вступить в контакт с сервером, работающим на удаленном компьютере, происходит цепочка событий. Прежде всего приложение должно обладать IP-адресом удаленного компьютера. Этот IP-адрес либо известен приложению изначально, либо получается в результате обращения к службе доменных имен DNS (Domain Name System). Служба DNS используется для преобразования доменного имени удаленного компьютера в соответствующий IP-адрес. Помимо IP-адреса удаленной системы приложение, обращающееся к серверу, должно указать, какой протокол будет использоваться для связи (TCP или UDP), а также какой порт удаленной системы следует использовать для подключения к удаленной системе.

ССЫЛКА

Более подробно о портах и о том, как они работают, рассказывается в главе 10, посвященной общеизвестным службам.

Информация, которую следует передать серверу, упаковывается в пакет TCP или UDP. Этот пакет вкладывается внутрь пакета IP. Исходя из сведений, хранящихся во внутренней таблице маршрутизации, ядро передает полученный IP-пакет соответствующему сетевому интерфейсу (это может быть сетевая карта Ethernet, модем, карта Token Ring и т. п.). Для дальнейшей передачи данных через сетевой канал сетевому интерфейсу, возможно, потребуется еще раз упаковать IP-пакет. Например, если речь идет об Ethernet, сетевая карта должна определить местоположение получателя пакета в локальной сети. Получателем может быть либо обычный сетевой узел, либо шлюз, соединяющий локальную сеть с другой сетью, которую называют удаленной сетью. Сетевой узел, являющийся получателем пакета, идентифицируется при помощи специального адреса MAC. Адрес MAC — это уникальное число, однозначно идентифицирующее любую сетевую карту. Каждой сетевой карте Ethernet соответствует свой собственный уникальный адрес MAC. Этот адрес жестко записывается внутрь каждой сетевой карты ее производителем. Чтобы один сетевой узел смог узнать MAC-адрес другого сетевого узла, используется специальный сетевой протокол ARP (Address Resolution Protocol). Как только MAC-адрес получателя определен и установлена возможность связи с соответствующим удаленным сетевым узлом, пакет передается по каналу связи получателю.

При использовании Ethernet подразумевается, что любая передача данных осуществляется в рамках локальной сети и что любые две карты, входящие в эту сеть, могут быть напрямую соединены каналом передачи данных. Иными словами, любая карта Ethernet, являющаяся передатчиком, «видит» другую карту Ethernet, являющуюся приемником. Как уже было отмечено, для идентификации карт Ethernet используются MAC-адреса. Соответствие между известными системе IP-адресами и соответствующими им MAC-адресами хранится в специальной таблице. Содержимое этой таблицы формируется при помощи протокола ARP. Любым IP-адресам, которые не входят в состав локальной сети, ставится в соответствие MAC-адрес специального сетевого узла, который называется шлюзом. Иными словами, если сетевой интерфейс не знает, по какому MAC-адресу следует передать пакет с известным IP-адресом назначения, он передает его шлюзовому сетевому узлу, а шлюз в дальнейшем решает, что делать с этим пакетом. Таким образом, на самом нижнем уровне сетевых карт Ethernet передача данных осуществляется на основе адресов MAC. Когда пакет принимается по месту назначения, IP-адрес обрабатывается принимающей системой. Если на самом деле пакет адресован другой системе, он передается обратно сетевому интерфейсу для отсылки в сеть Ethernet. Если компьютер, принявший пакет, является шлюзом, пакет принимается через один сетевой интерфейс и отправляется через другой сетевой интерфейс. Система, принявшая пакет и не

являющаяся шлюзом, может выполнить дальнейшую пересылку этого пакета в сеть, однако в реальности этого не происходит, так как по умолчанию в Linux дальнейшая пересылка принятых пакетов отключена. То есть не следует разыскивать шлюз, передавая в сеть пакеты случайным образом.

Заголовок пакета IP

Из всего ранее сказанного можно сделать вывод, что каждый пакет IP состоит из двух частей: заголовка и данных. Заголовок IP-пакета обладает жестко заданным форматом, который проверяется каждый раз в процессе обработки пакета. Процедура обработки пакета также предусматривает модификацию заголовка IP. Каждая система, через которую передается IP-пакет, проверяет заголовок этого пакета на целостность. Для этой цели используется контрольная сумма заголовка. Если фактическая контрольная сумма заголовка пакета не совпадает со значением, хранящимся в самом заголовке, система считает, что во время передачи данных через сетевой канал пакет был поврежден (как правило, это происходит из-за коллизий пакетов). В этом случае пакет отбрасывается, то есть игнорируется. Подобное может произойти и по другим причинам, например, в случае, если целевой сетевой узел недоступен, целевая сеть недоступна, пакет слишком большой, в связи с чем его нельзя передать через очередной сетевой сегмент, при этом для пакета установлен флаг DNF (Do Not Fragment — не фрагментировать) и т. п. Если пакет отбрасывается по какой-либо из причин за исключением искажения в процессе передачи, система, которая отбросила пакет, отправляет по адресу отправителя пакета (этот адрес указывается в составе заголовка пакета) сообщение ICMP с указанием причины, по которой пакет не может быть передан дальше.

Диаграмма заголовка IP демонстрируется на рис. 9.1. Обратите внимание, что заголовок разделен на несколько частей. Это сделано по двум причинам: во-первых, заголовок слишком длинный, и его сложно рассматривать как единую непрерывную последовательность битов (а именно в виде такой непрерывной последовательности он передается через линию связи); во-вторых, заголовок удобнее читать и о нем удобнее рассказывать, если он разделен на шесть четырехбайтовых разделов.

ПРИМЕЧАНИЕ

Разделение заголовка на 32-битные сегменты выполнено намеренно. Дело в том, что системы, основанные на процессоре Pentium, за один раз способны передать по основной шине одно 32-битное число. Такое сочетание из 32 битов называется машинным словом. Таким образом, системы, основанные на Pentium, используют слово длиной четыре байта. В отличие от них более старые системы, основанные на процессорах 286, использовали слово длиной в два байта. Двухбайтное слово использовалось также и в системах, основанных на процессорах 386SX, несмотря на то, что внутри самих процессоров 386SX обработка данных осуществлялась с использованием четырехбайтных слов.

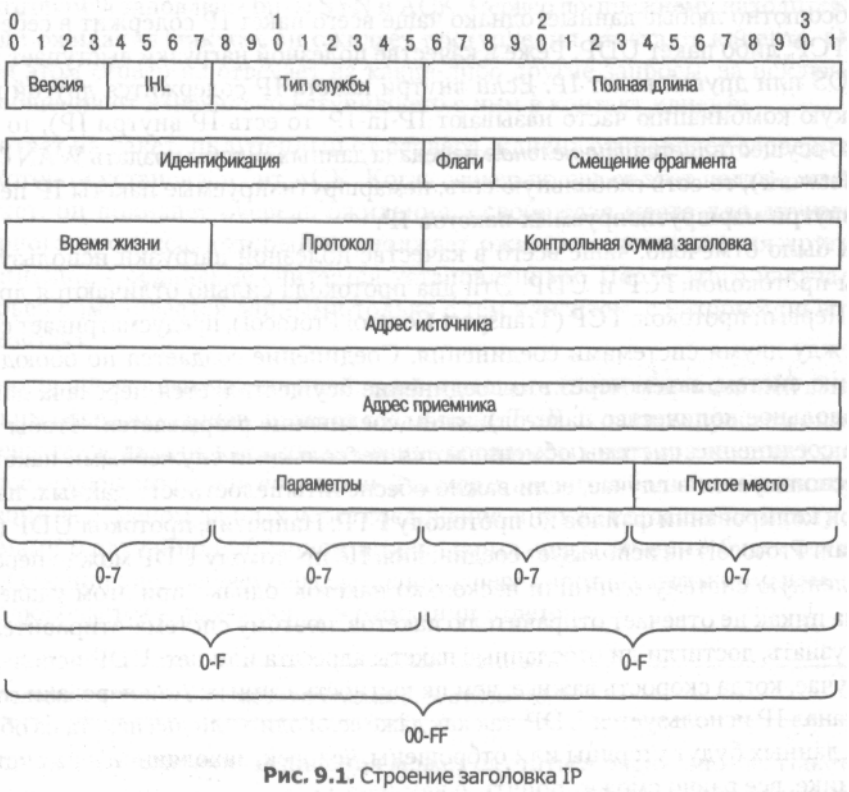

Рис. 9.1. Строение заголовка IP

Можно заметить, что длина почти всех полей заголовка кратна четырем битам. То есть в заголовок

входят поля длиной 4,8, 16 и 32 бита. Это сделано намеренно, так как фактически все системы работают с порциями данных, кратными байту, а байт, как уже было сказано, равен восьми битам. Таким образом, если длина поля кратна четырем битам, его удобнее обрабатывать. Исключениями являются поле флагов и поле смещения фрагмента. Длина этих полей по отдельности не кратна четырем битам, однако вместе они составляют последовательность из 16 бит, то есть из двух байтов. В данной главе в основном мы будем обсуждать поля адреса источника и адреса приемника пакета. Однако в дальнейшем, в особенности в главе 16, нам придется обратить внимание и на другие поля заголовка IP.

ПРИМЕЧАНИЕ

Как видно, заголовок пакета содержит в себе большой объем информации, однако наиболее важными сведениями, хранящимися в заголовке, являются сведения о том, откуда был послан пакет (адрес источника) и кому он адресован (адрес приемника). В конце концов, именно для маршрутизации сообщений и был придуман протокол IP.

Пакет данных IP

Сразу же за заголовком IP в пакете IP размещается раздел данных. Этот раздел часто называют полезной нагрузкой. В качестве полезной нагрузки могут выступать абсолютно любые данные, однако чаще всего пакет IP содержит в себе либо пакет TCP, либо пакет UDP. Реже в качестве полезной нагрузки выступает пакет NetBIOS или другой пакет IP. Если внутри пакета IP содержится другой пакет IP (такую комбинацию часто называют IP-in-IP, то есть IP внутри IP), то говорят, что осуществляется *туннельная* передача данных. Чтобы создать WAN (Wide Area Network), то есть глобальную сеть, немаршрутизируемые пакеты IP передаются внутри маршрутизируемых пакетов IP.

Как было отмечено, чаще всего в качестве полезной нагрузки используются пакеты протоколов TCP и UDP. Эти два протокола сильно отличаются друг от друга. Первый протокол, TCP (Transport Control Protocol), предусматривает создание между двумя системами соединения. Соединение создается по обоюдному согласию систем, затем через это соединение осуществляется передача данных (произвольное количество пакетов), затем соединение разрывается. Чтобы установить соединение, системы обмениваются несколькими служебными пакетами. TCP используется в случае, если важно обеспечить целостность данных, например, при копировании файлов по протоколу FTP. Напротив, протокол UDP (User Datagram Protocol) не использует соединения. По протоколу UDP можно передать на удаленную систему один или несколько пакетов, однако при этом удаленная система никак не отвечает отправителю пакетов, поэтому система-отправитель не может узнать, достигли ли отосланные пакеты адресата или нет. UDP используется в случае, когда скорость важнее, чем целостность данных. Для передачи голоса через канал IP используется UDP, так как даже если один или два пакета из общего потока данных будут утеряны или отброшены, человек, находящийся на системе-приемнике, все равно сможет понять, о чем идет речь.

Если вы взглянете внутрь файла /etc/services, вы увидите, что для большинства служб перечислены как порт UDP, так и порт TCP. Несмотря на то, что службе соответствует два порта (TCP и UDP), большинство служб используют только один из них. В частности, протокол UDP используется такими службами, как SNMP (Simple Network Management Protocol), TFTP (Trivial File transfer Protocol), RLP (Resource Location Protocol) и некоторыми другими. Протокол TCP используется такими службами, как FTP (File Transfer Protocol), telnet, SMTP (Simple Mail Transport protocol) и многими другими. Служба доменных имен DNS (Domain Name System) использует как TCP, так и UDP. Если запрос DNS небольшой, ответ будет упакован в виде одного пакета UDP. Если же речь идет о крупном запросе, например о передаче содержимого зоны (zone transfer), для этой цели используется TCP.

Как отмечалось ранее, протокол TCP устанавливает соединение между двумя системами. Такое соединение гарантирует целостность передаваемых данных. Соединение формируется при помощи трехэтапной процедуры «рукопожатия» между двумя системами.

1. Клиент вступает в контакт с сервером (серверным процессом) и запрашивает соединение (бит SYN установлен).

2. Сервер, находящийся в очереди ожидания, отправляет обратно клиенту пакет, в котором установлены биты SYN и ACK. Сервер по-прежнему находится в очереди ожидания, так как он ожидает поступления ответа от клиента. Однако при этом сервер не отвечает на какие-либо другие запросы, за исключением сообщения от изначально вступившего с ним в контакт клиента.

3. В ответ на пакет, полученный от сервера, клиент отправляет серверу пакет, в котором установлен бит ACK. Когда сервер получает этот подтверждающий пакет, он покидает очередь ожидания, освобождая место для другого серверного процесса, который продолжает ожидание поступления других соединений. Соединение считается установленным. После этого изначальный сервер обменивается данными только с тем клиентом, с которым он вступил в контакт.

Данная последовательность действий — это достаточно упрощенный взгляд на то, каким образом устанавливается соединение TCP. В данном простом описании опущены такие детали, как использование

порядковых номеров пакетов (для каждого последующего пакета порядковый номер становится больше на единицу) для слежения за обменом данных через соединение, а также соглашение между двумя системами о случайно выбранном номере порта, через который будет осуществляться соединение. Сервер передает данные через порт с другим номером, и клиент подключается к этому порту в ожидании ответов.

ССЫЛКА
Более подробно о портах и службах рассказывается в главе 10.

В процессе установки соединения может случиться очень многое. Пакеты могут потеряться или исказиться и в связи с этим они могут быть отброшены; пакеты могут быть отброшены также потому, что они слишком большие; случайно выбранный сервером порт может оказаться занятым и т. п. Сервер будет ждать в течение определенного времени, а затем попробует передать данные снова. Со временем соединение будет продолжено, а после завершения обмена данными — завершено.

ССЫЛКА
Одна из возможных атак называется атака SYN DoS. Эта атака основана на том обстоятельстве, что сервер ожидает в очереди до тех пор, пока не получит ответ от клиента. Более подробно об этой атаке рассказывается в главе 13.

Коротко об ICMP

В отличие от протоколов TCP и UDP пакеты протокола ICMP (Internet Control Message Protocol) не передаются через сеть в составе пакетов IP. Протокол ICMP обладает собственным заголовком пакета. Если вы когда-либо использовали утилиту ping, значит, вы уже знакомы с ICMP, так как работа этой утилиты основана на использовании данного протокола. Однако ICMP используется не только для утилиты ping. Сетевой ICMP-пакет ping на самом деле является двумя отдельными пакетами. Как указано в RFC 792, сообщение эхо-запроса (Echo Request Message или ping) — это пакет ICMP типа 8. Ответом на это сообщение является сообщение эхо-ответа (Echo Reply Message или pong), то есть пакет ICMP типа 0. Полный список разновидностей пакетов ICMP, извлеченный из RFC 792, в переводе на русский язык выглядит следующим образом:

ICMP Message Types and Codes (Типы и коды сообщений ICMP)

Тип 0 Echo Reply Message (эхо-ответ)
Тип 3 Destination Unreachable Message (приемник недоступен) Код
- 0 = сеть недоступна;
- 1 = сетевой узел недоступен;
- 2 = протокол недоступен;
- 3 = порт недоступен;
- 4 = необходима фрагментация и флаг DF установлен;
- 5 = исходный маршрут не работает.

Тип 4 Source Quench Message (заглушить источник)
Тип 5 Redirect Message (перенаправить) Код
- 0 = перенаправлять дейтаграммы для сети
- 1 = перенаправлять дейтаграммы для сетевого узла
- 2 = перенаправлять дейтаграммы для типа службы и сети
- 3 = перенаправлять дейтаграммы для типа службы и сетевого узла

Тип 8 Echo Message (эхо-запрос)
Тип 11 Time Exceeded Message (время истекло) Код
0= при передаче сообщения истекло время жизни (time to live)
0= истекло время формирования сообщения из фрагментов (fragment reassembly time)
Тип 12 Parameter Problem Message (проблема с параметром)
Код
0 = проблема, связанная с параметром
Тип 13 Timestamp Message (отметка времени)
Тип 14 Timestamp Reply Message (ответ на отметку времени)
Тип 15 Information Request Message (запрос информации)
Тип 16 Information Reply Message (ответ на запрос информации)

Если ранее вы уже работали с компьютерами, подключенными к сети, возможно, вы уже сталкивались с сообщением «сетевой узел недоступен» (host unreachable), информирующим вас о том, что удаленный компьютер, с которым вы пытаетесь вступить в контакт, по тем или иным причинам не может

выйти на связь. Однако несмотря на то, что другие сообщения ICMP зачастую остаются незаметными для вас, они активно используются в процессе функционирования сетей, основанных на IP. Протокол ICMP является чрезвычайно важным механизмом, обеспечивающим функционирование системы в сети. Linux использует ICMP для определения величины MTU (Maximum Transmission Unit). Сетевые карты Ethernet передают данные большими блоками. Каждый такой блок называется единицей передачи (transmission unit). По умолчанию размер блока составляет 1500 байт. Благодаря использованию столь больших блоков обмен данных в скоростных сетях существенно ускоряется. Однако в медленных сетях использование столь крупных блоков малоэффективно. Дело в том, что при передаче столь крупного блока возрастает вероятность ошибки, а если при передаче блока возникает ошибка, блок отбрасывается и его необходимо передать заново. Таким образом, в медленных сетях слишком много крупных блоков потребовалось бы передавать заново. Чтобы снизить вероятность ошибки при передаче блока, размер блока уменьшают. Параметр MTU для каналов телефонной связи, как правило, составляет около 296 байт. Чтобы определить оптимальную величину MTU для некоторого соединения, система Linux устанавливает флаг DF (Don't Fragment — не фрагментировать). Если блок установленного размера невозможно передать через выбранную линию связи или во время передачи блока возникает ошибка, пакет отбрасывается и системе-источнику отсылается сообщение ICMP типа 3 с кодом 4 (ICMP Type 3 Code 4 Fragmentation Needed and DF Set), то есть система-источник извещается о том, что требуется фрагментация, а бит DF установлен. В этом случае система Linux уменьшает MTU и пытается выполнить передачу снова. Таким образом, Linux самостоятельно настраивает размер пакета для получения оптимальной скорости передачи данных через канал.

Какая взаимосвязь между ICMP и безопасностью? Взломщики систем для начала зачастую исследуют сеть, в которую они проникают, с использованием эхо-пакетов ping для того, чтобы получить перечень работающих систем. Сетевой брандмауэр можно настроить таким образом, чтобы он отбрасывал пакеты ICMP. Однако при этом не следует отбрасывать абсолютно все пакеты, в противном случае вы рискуете не получать важных сообщений для корректной настройки MTU, а также таких важных служебных сообщений, как Host Unreachable (сетевой узел недоступен). Брандмауэр необходимо настроить таким образом, чтобы он не пропускал сообщения ICMP типа 8. К сожалению, в наши дни большинство взломщиков уже отлично знает о том, что сообщения эхо-запроса ICMP Type 8 (Echo Request) не пропускаются сквозь брандмауэры. Для обнаружения работающих в сети систем взломщиками используются другие более изощренные способы.

Сеть и маршрутизация в Интернете

Маршрутизация (routing) — это метод, благодаря которому пакеты ICMP и IP находят путь от одного компьютера к другому. В Интернете данные передаются не напрямую от источника к приемнику, а по цепочке. Пакеты передаются через сеть от системы к системе из интерфейсного устройства одной системы в интерфейсное устройство другой системы до тех пор, пока они не достигнут системы-приемника. Это происходит при условии, что во время пути с пакетами ничего не происходит. В данной книге подразумевается, что все интерфейсные устройства являются сетевыми картами Ethernet, однако интерфейсное устройство может быть также устройством подключения к каналу типа «точка-точка». Интерфейсными устройствами могут быть модемы, сетевые карты Token Ring и т. п. Важным обстоятельством является то, что каждому интерфейсному устройству должен быть поставлен в соответствие уникальный IP-адрес. Если между двумя интерфейсными устройствами существует маршрут, эти интерфейсные устройства не могут обладать одним и тем же IP-адресом. Многим IP-адресам ставится в соответствие имя. Одному и тому же имени можно поставить в соответствие несколько IP-адресов, однако в этом случае вы должны предпринять некоторые дополнительные организационные действия, чтобы обеспечить корректное функционирование подобной схемы.

В заводских условиях в каждую сетевую карту Ethernet жестко записывается уникальный численный идентификатор, который называется аппаратным адресом MAC. MAC-адрес — это число, состоящее из шести двухзначных шестнадца-теричных чисел, например: 01:23:ab:cd:4e:5f. Как уже отмечалось ранее, на более верхнем уровне для идентификации сетевых узлов используются IP-адреса. Чтобы получить возможность вступить в контакт с удаленным узлом, для которого известен IP-адрес, система должна знать соответствие между IP-адресом и MAC-адресом для этого удаленного узла. Информация о соответствии между IP-адресами и MAC-адресами для известных системе удаленных сетевых узлов формируется при помощи протокола ARP (Address Resolution Protocol — протокол сопоставления адреса) и хранится в специальной внутренней таблице, которая называется ARP-кэшем. Чтобы связаться с узлом, для которого известен IP-адрес, система обращается к этой таблице и извлекает из нее соответствующий MAC-адрес. ARP-кэш используется в течение определенного времени (как правило, пара минут), после чего он

считается устаревшим и стирается из памяти. После этого система вновь формирует ARP-кэш с использованием протокола ARP. Протокол ARP используется также в случае, если информация о некотором IP-адресе отсутствует в ARP-кэше. Любым IP-адресам, не принадлежащим локальной сети, ставится в соответствие MAC-адрес сетевого узла, который является шлюзом для локальной сети.

ПРИМЕЧАНИЕ

На самом деле IP-адреса назначаются не компьютерам, а коммуникационным устройствам. Коммуникационные устройства напрямую подключаются к сети. Это могут быть сетевые карты (NIC, Network Interface Card), модемы, спулеры принтеров HP JetDirect и т. п. Сам по себе компьютер не является коммуникационным устройством. Каждое коммуникационное устройство получает свой собственный IP-адрес. Один компьютер может быть оснащен несколькими коммуникационными устройствами.

Протокол ARP можно использовать и для решения обратной задачи (в этом случае говорят о протоколе RARP — Reverse Address Resolution Protocol). Благодаря этому система может узнать свой собственный IP-адрес. Сетевые карты HP JetDirect, установленные в принтерах HP LaserJet и спулерах принтеров HP JetDirect, по умолчанию слушают сеть, чтобы сообщить сетевому узлу свой MAC-адрес и получить от него соответствующий IP-адрес.

По умолчанию, когда говорят о сетевом узле, входящем в состав локальной сети, подразумевают, что этот сетевой узел напрямую подключен к тому же самому проводу или сетевому сегменту, что и все остальные компьютеры локальной сети. Если для того, чтобы передать пакет к месту назначения, требуется воспользоваться услугами еще одного промежуточного сетевого узла (шлюза или маршрутизатора), значит, сетевой узел, являющийся приемником, не принадлежит к локальной сети, даже если он подключен к тому же самому сетевому проводу. Это относится также и к машинам, которые принадлежат к одному и тому же диапазону адресов класса C, однако используют маску подсети с переменной длиной (Variable Length Subnet Masking, VLSM) и, таким образом, относятся к разным подсетям. Подробнее об этом рассказывается чуть позже, при обсуждении технологии CIDR.

Размер кэша ARP может достигать 254 записей (по одной записи для каждого IP-адреса), в случае если речь идет о полной сети класса C, однако обычно размер таблицы ARP меньше. Таблицы ARP, как правило, не вырастают до значительных размеров, если только вы не имеете дело с маршрутизатором, подключенным к нескольким подсетям. В этом случае кэш ARP может достигнуть значительных размеров.

Что такое CIDR

Технология CIDR (Classless Inter-Domain Routing — бесклассовая междоменная маршрутизация) позволяет максимизировать использование ограниченного адресного пространства, доступного в рамках существующей реализации стандарта IPv4 (Internet Protocol version 4). Прочитав материал данного и последующего разделов, вы получите хорошее представление о том, как настраивается сетевая конфигурация компьютера, даже если раньше вы никогда не настраивали компьютер, подключенный к сети.

Предпосылки

С середины 1990-х и по сей день технология CIDR является наиболее распространенной тенденцией развития маршрутизации в сетях, основанных на IP. Эта концепция появилась в 1993 году для того, чтобы компенсировать недостатки существующей схемы распределения адресов IPv4 до тех пор, пока не будет введена в строй следующая версия протокола IP под названием IPv6 (также называемая IPng, то есть IP next generation — *следующее поколение* IP).

В настоящее время технология IPv6 проходит тщательное тестирование. Когда она вступит в строй, адресное пространство IP будет расширено на несколько порядков. В рамках IPv6 также реализованы специальные более совершенные механизмы защиты. Те из читателей, которые пожелают принять участие в формировании будущего уже сейчас, могут попробовать IPv6 в действии, так как операционная система Linux поддерживает IPv6 на уровне ядра. Но до тех пор, пока IPv6 не получит широкого распространения, благодаря CIDR вы можете максимально эффективно использовать то, чем вы ограничены в рамках IPv4.

Чтобы лучше понять, зачем вообще нужна технология CIDR, давайте вернемся назад по шкале времени в конец 1980-х годов. В то время для поиска и идентификации компьютерных систем в сети использовался протокол IPv4. Однако в то далекое время к Интернету было подключено относительно немного компьютеров. Количество компьютеров, которые нуждались в подключении к Интернету, также было небольшим. В действительности огромное количество систем использовали для передачи данных протокол UUCP (UNIX-to-UNIX Copy Protocol). В рамках протокола UUCP компьютеры устанавливали между собой связь в заранее определенное время, происходил обмен электронной почтой, после чего связь

разрывалась. В то далекое время запас IP-адресов, доступных в рамках IPv4, казался неисчерпаемым. Однако это было до того, как появился первый web-браузер под названием Mosaic. С появлением Mosaic Интернет стал стремительно расти. По мере его роста увеличивалась потребность в новых IP-адресах. Чем большее количество компьютеров подключалось к Интернету, тем большее количество уникальных IP-адресов требовалось для идентификации машин в Интернете.

Базовые сведения о маршрутизации IP

Те из читателей, которые хорошо знакомы с механизмом маршрутизации, основанной на классах, могут пропустить данный раздел и перейти к изучению следующего. Компьютеры понимают только две базовых цифры: 1 и 0, в то время как большинство людей оперируют десятью базовыми цифрами (от 0 до 9). Для того чтобы облегчить работу людей, имеющих дело с компьютерами, компьютерные инженеры пошли на компромисс. Каждый компьютер в Интернете обладает уникальным IP-адресом, который может быть представлен в виде строки нулей и единиц. Другими словами, в современном Интернете IP-адрес — это набор из 32 битов. Для удобства восприятия эту последовательность разбивают на четыре группы по восемь битов. Каждая такая группа называется *октетом.* Таким образом, получается четыре числа, каждое из которых может принимать любое из значений в диапазоне от 0 (восемь двоичных нулей) до десятичного 255 (восемь двоичных единиц). Эти четыре числа, представляющие собой IP-адрес, записываются в формате: XXX.XXX.XXX.XXX (где X — это один десятичный разряд). Например, 192.213.150.205. Такое обозначение называется *точечной десятичной нотацией* (dotted decimal notation). Такой формат записи облегчает восприятие IP-адресов для людей. Все IP-адреса делятся на четыре категории, называемых *классами.* Классы обозначаются латинскими буквами A, B, C, D. Разделение IP-адресов на классы осуществляется в соответствии со старшими четырьмя битами самого старшего октета IP-адреса:

- Класс A = старший октет равен от 0 до 127 (первые четыре бита 0000) — в каждой сети 16 777 216 сетевых узлов;

- Класс B = старший октет равен от 128 до 191 (первые четыре бита 1000) -в каждой сети 65 534 сетевых узла;

- Класс C = старший октет равен от 192 до 223 (первые четыре бита 1100) -в каждой сети 256 сетевых узлов;

- Класс D = старший октет равен всем оставшимся значениям (первые четыре бита 1110) — количество сетевых узлов по умолчанию не определено.

В каждом классе определяется некоторое количество IP-сетей, в состав каждой из которых может входить не более некоторого максимального количества сетевых узлов. Для нумерации сети используются старшие биты IP-адреса, а для нумерации сетевых узлов в сети используются младшие биты этого же самого IP-адреса. Таким образом, один и тот же IP-адрес идентифицирует как некоторую сеть, так и определенный сетевой узел в данной сети. Количество битов, используемых для идентификации сети и сетевого узла, в разных классах разное. В классе A для идентификации сети используются старшие восемь битов 32-битного IP-адреса. Остальные 24 бита используются для идентификации сетевого узла. Например, сеть класса A может обладать следующим адресом: 10.XXX.XXX.XXX. В классе B для идентификации сети используются два старших октета IP-адреса. Например, сеть класса B может обладать следующим адресом: 172.32.XXX.XXX. В классе C для идентификации сети используются три старших октета IP-адреса. Например, сеть класса C может обладать следующим адресом: 192.168.1.XXX. Класс D зарезервирован для тестовых целей. Как обозначено символами XXX, для идентификации сетевых узлов в классе A используются целых три октета (именно поэтому в каждой сети класса A может содержаться до 16 777 216 компьютеров). В сети класса B для идентификации сетевых узлов используется два октета (получается, что в сети класса B может одновременно работать до 65 534 сетевых узлов). Наконец, в сети класса C для идентификации сетевого узла используется только один самый младший октет, то есть всего восемь битов (именно поэтому в состав любой сети класса C может входить не более 256 сетевых узлов). Как можно заметить, в рамках данной схемы граница между частью IP-адреса, идентифицирующей сеть, и частью IP-адреса, идентифицирующей сетевой узел, всегда проходит по границе между октетами. Иными словами, адрес сети и адрес сетевого узла в этой сети всегда состоят из целого количества октетов. Чтобы разделить 32-битный IP-адрес на две этих составляющие, используется сетевая маска. *Сетевая маска* — это 32-битное число, для каждого из битов которого известно, что если бит равен 1, значит, соответствующий ему бит IP-адреса идентифицирует сеть, а если бит сетевой маски равен 0, значит, соответствующий ему бит IP-адреса идентифицирует сетевой узел. Например, сетям класса B соответствует сетевая маска 255.255.0.0, а сетям класса C соответствует сетевая маска 255.255.255.0.

ПРИМЕЧАНИЕ

Напомню, что один байт равен восьми битам. Один байт может хранить значение от 0 до десятичного 255. Десятичное 255 соответствует шестнадцатеричному FF. Один шестнадцатеричный разряд может принимать значения от 0 до F: 0,1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Теперь, когда вы познакомились с основами адресации IP, базирующейся на классах сетей, я приведу набор определений, которые потребуются вам для дальнейшего знакомства с материалом книги.

- *Адрес сетевого узла* (host address) — это уникальный адрес, который ставится в соответствие коммуникационному устройству, установленному в компьютере. Если компьютер оснащен несколькими коммуникационными устройствами (например, несколько сетевых карт Ethernet и несколько модемов), каждому из этих устройств назначается свой собственный уникальный адрес. Это означает, что сетевой узел (компьютер или маршрутизатор) может обладать несколькими сетевыми узлами (это обозначается английским термином multi-homed). Того же самого эффекта можно добиться иначе: вы можете присвоить несколько разных IP-адресов одному и тому же коммуникационному устройству. Эта процедура называется назначением IP-псевдонимов (IP aliasing).

- *Адрес сети* (network address) — IP-адрес, в котором несколько старших битов (в соответствии с сетевой маской) идентифицируют сеть, а остальные (младшие) биты, предназначенные для идентификации сетевого узла в данной сети, равны нулю. Иными словами, адрес сети является самым младшим адресом из всех IP-адресов адресного диапазона некоторой IP-сети. Например, в сети класса C с адресами от 192.168.1.0 до 192.168.1.255 адрес сети будет равен 192.168.1.0. Очевидно, что адрес сети нельзя использовать для идентификации какого-либо сетевого узла в этой сети.

- *Широковещательный адрес* (broadcast address) — IP-адрес, в котором несколько старших битов (в соответствии с сетевой маской) идентифицируют сеть, а остальные (младшие) биты, предназначенные для идентификации сетевого узла в данной сети, равны единице. Иными словами, адрес сети является самым старшим адресом из всех IP-адресов адресного диапазона некоторой IP-сети. Например, в сети класса C с адресами от 192.168.1.0 до 192.168.1.255 широковещательный адрес будет равен 192.168.1.255. Любой пакет, отосланный по широковещательному адресу, будет получен всеми сетевыми узлами, работающими в рамках данной сети. Очевидно, что широковещательный адрес для некоторой сети нельзя использовать для идентификации какого-либо сетевого узла в этой сети.

- *Сетевая маска* (netmask) — 32-битное число, указывающее, какая часть IP-адреса используется для идентификации сети, а какая часть IP-адреса используется для идентификации сетевого узла в рамках данной сети. Старшие биты сетевой маски равны единице, а младшие — нулю. Если некоторый бит сетевой маски равен 1, значит, соответствующий бит IP-адреса служит для идентификации сети, если некоторый бит сетевой маски равен 0, значит, соответствующий бит IP-адреса служит для идентификации сетевого узла. Для сетей класса C (например, для сети из приведенного ранее примера) сетевая маска равна 255.255.255.0. Если сеть принадлежит классу B, сетевая маска для нее будет равна 255.255.0.0. Иными словами, чтобы получить сетевую маску для некоторой сети, необходимо определить часть IP-адреса, которая остается неизменной для всех сетевых узлов в рамках данной сети, и заменить эту часть единицами. Часть IP-адреса, которая меняется от узла к узлу, заменяется нулями.

Десятилетие назад к Интернету было подключено очень небольшое количество персональных компьютеров, поэтому в конце 1980-х схема адресации IPv4 казалась неисчерпаемым источником адресов даже с учетом того, что далеко не все IP-адреса из всего допустимого диапазона 32-битных чисел могут использоваться для идентификации конкретных сетевых узлов.

СОВЕТ

Сетевая маска определяет сеть. Если две системы принадлежат одной локальной сети, но при этом используют различные сетевые маски, они будут работать так, как будто принадлежат разным сетям IP.

Реализация CIDR с использованием VLSM

Описанная в предыдущем разделе схема адресации IP, основанная на диапазонах адресов в соответствии с четырьмя классами, устарела. Термины, используемые для ее описания — класс A, класс B, класс C, — тоже устарели, однако они по-прежнему широко используются, так как их смысл легко усвоить. В данной книге мы также будем распространены эти термины для простоты изложения. Я рассказал вам о схеме адресации, основанной на классах, чтобы было проще перейти к рассказу о CIDR — новой, модифицированной схеме адресации IPv4. Название CIDR расшифровывается как Classless Inter-Domain Routing — бесклассовая междоменная маршрутизация. Из названия CIDR можно сделать вывод, что эта схема не основана на классах. Однако чтобы понять CIDR, вы должны прежде всего узнать, что означает

маска подсети переменной длины (Variable Length Subnet Masking, VLSM). В рамках старой, основанной на классах схемы адресации каждому IP-адресу ставится в соответствие маска подсети по умолчанию. Для IP-адреса, принадлежащего диапазону адресов класса C, старшие 24 бита (три старших октета) маски подсети равны единице, а остальные (младшие) 8 битов маски подсети равны нулю. Таким образом, маска подсети для любого адреса класса C равна 255.255.255.0 (шестнадцатеричное: ffffff00). Для классов A и B маска подсети равна 255.0.0.0 и 255.255.0.0 соответственно. Таким образом, если вы получаете в свое распоряжение адрес сети класса C, вы получаете 256 уникальных IP-адресов, из которых два зарезервированы: один является адресом сети, а другой — широковещательным адресом. В рамках схемы адресации, основанной на классах, каждому, кто нуждается в IP-адресе, выдается адрес сети одного из классов: A, B или C. Таким образом, IP-адреса распределяются наборами по 256 (класс C), 65 536 (класс B) или 16 777 216 (класс A) адресов. В современных условиях недостатка IP-адресов такой подход неэкономичен. Бесклассовая схема адресации позволяет разделить эти наборы на множество менее крупных диапазонов. Схема адресации, основанная на классах, предусматривает использование маски подсети, в которой количество битов, равных единице, обязательно должно быть кратно восьми. Например, для сети класса C маска подсети выглядит следующим образом (слева стоит двоичное число, а справа приводится точечная десятичная нотация):

```
1111111.1111111.1111111.00000000  =  255.255.255.0
```

Как видите, старшие 24 разряда (8 битов умножить на 3 октета) равны единице. Таким образом, адрес сети в схеме адресации, основанной на классах, всегда включает в себя количество битов, кратное восьми. Схема VLSM снимает это ограничение.

Для примера представим, что вам выделили один IP-адрес класса C (192.168.0.0). Вам надо организовать работу двух географически удаленных друг от друга офисов, в составе каждого из которых работает по 50 систем. Вы планируете организовать сетевой обмен данными по протоколу IP между всеми системами сети. В рамках схемы адресации, основанной на классах, вы можете объединить все 100 систем в единую IP-сеть с адресом класса C. Но, к сожалению, вы не можете обеспечить прямой скоростной обмен данными между обоими офисами. Иными словами, для обмена данными между всеми имеющимися у вас компьютерами нельзя использовать один сетевой провод. Однако при этом вы вынуждены использовать для всех компьютеров один и тот же адрес сети. Говоря иначе, компьютеры каждого из офисов считают, что все компьютеры другого офиса являются локальными. Если все ваши компьютеры рассматриваются как локальные по отношению друг к другу, значит, подразумевается, что передача данных из одного офиса в другой может осуществляться напрямую, то есть без использования шлюза. Шлюз требуется только в случае, если необходимо передать данные узлу, который не входит в локальную сеть. К сожалению, в вашем случае физическое расположение офисов к этому не располагает.

Шлюз (gateway) — это компьютерная система (обычный компьютер или специализированный маршрутизатор), которая обладает двумя или большим количеством адресов сети — по крайней мере, один из этих адресов соответствует локальной сети, а все остальные соответствуют другим сетям. Таким образом, шлюз напрямую подключен к нескольким сетям. Шлюз выполняет передачу пакетов из сети в сеть. Любые полученные шлюзом пакеты, которые не предназначены для локальной сети, передаются в один из сетевых интерфейсов, подключенных к другим сетям. Выбор интерфейса осуществляется в соответствии с информацией, хранящейся в таблице маршрутизации. В рамках схемы маршрутизации, основанной на классах, для каждого из ваших офисов потребовалось бы выделять отдельный адрес сети класса C. В условиях недостатка IP-адресов такой подход нельзя назвать экономичным.

В рамках CIDR вы можете разбить одну сеть с адресом класса C на любое удобное для вас количество более мелких диапазонов IP-адресов. VLSM позволяет использовать маску подсети, в которой часть, идентифицирующая сеть, может состоять из любого количества битов, которое не обязательно должно быть кратным восьми. В маске подсети класса C старшие 24 бита маски подсети равны единице. Таким образом, маска подсети класса C равна 255.255.255.0 (11111111. 11111111.11111111.00000000). Теперь представьте, что вы присвоили самому старшему биту самого младшего октета значение 1 вместо 0. Теперь в составе маски подсети 25 единиц и 7 нулей, и она становится равной 255.255.255.128. Если теперь вы будете использовать эту сетевую маску для обоих офисов, вы сможете использовать половину диапазона адресов класса C для одного офиса и вторую половину диапазона адресов — для другого офиса. В этом случае компьютеры, установленные в одном из офисов, уже не будут воспринимать компьютеры из другого офиса как сетевые узлы локальной сети. Таким образом, для передачи данных из одного офиса в другой офис по умолчанию будет использоваться шлюз. В результате вы получаете две сети, IP-адреса одной из них простираются от 192.168.0.0 до 192.168.0.127, а вторая включает в себя IP-адреса от 192.168.0.128 до 192.168.0.255. В качестве адреса первой сети будет использоваться адрес 192.168.0.0, а в качестве адреса второй сети — адрес 192.168.0.128. Наконец, адрес 192.168.0.127 будет широковещательным адресом для первой сети, а адрес 192.168.0.255 будет широковещательным адресом для второй сети.

Используя подобный подход, вы можете разбить предоставленный вам диапазон адресов класса C на

четыре отдельных сети, на восемь сетей, 16 сетей, 32 сети и т. д. На самом деле для идентификации сети можно использовать любое количество двоичных разрядов IP-адреса, начиная с восьми и заканчивая 30-ю. Дело в том, что в стандарте IPv4 каждый IP-адрес — это 32-битное число. Значит, если вы будете использовать для идентификации сети все 32 бита (сетевая маска 255.255.255.255), то в результате получится сеть, в которой может использоваться всего один адрес, таким образом, отпадает надобность в адресе сети и в широковещательном адресе. Если для идентификации сети используется 31 бит (сетевая маска 255.255.255.254), значит, у вас получится сеть с двумя IP-адресами, однако при этом вы должны использовать один из них для адресации самой сети, а второй — в качестве широковещательного адреса. В результате у вас не остается ни одного адреса для адресации сетевых узлов — это неприемлемо.

В рамках схемы адресации, основанной на классах, сетевая маска всегда начинается с октета 255. То есть самый старший октет всегда равен 255. Далее идут октеты, равные 255, до тех пор пока не встретится первый октет, равный 0. После этого до самого младшего из октетов идут октеты, равные 0.

В рамках CIDR самый старший октет сетевой маски должен оставаться равным 255, однако первый нулевой октет можно заменить на любое из чисел: 128, 192, 224, 240, 248, 252 или 254 (за исключением последнего октета), после этого «пограничного» октета все остальные более младшие октеты должны быть равны нулю. Адрес сети и широковещательный адрес выбираются в соответствии с подсетью. Теперь любая подсеть может быть определена при помощи маски подсети с переменной длиной. Для того чтобы определить подсеть, достаточно просто указать количество битов, используемое для идентификации сети. Это значение может быть от 8 до 32 (исключая 31). Данное значение, как правило, записывается через дробь. Например: /8 или /21. Таким образом, любой сетевой узел можно идентифицировать при помощи IP-адреса и размера маски VLSM, благодаря чему сразу становится ясно, как выглядит адрес сети, широковещательный адрес и сетевая маска.

Например, если некоторому компьютеру предлагается присвоить адрес 192. 168.0.50/27, сразу становится ясно, что адрес сети 192.168.0.32, широковещательный адрес 192.168.0.63, а сетевая маска 255.255.255.224. Для тех, кому сложно с ходу привыкнуть к такой форме записи, я привожу таблицу, при помощи которой вы сможете упростить эту процедуру.

Таблица 9.2. Таблица трансляции адресов CIDR

Маска	A	B	C	Сети	Кратно	Шестнадцатеричное
.0	/8	/16	/24	1	нет	00
.128	/9	/17	/25	2	128	80
.192	/10	/18	/26	4	64	CO
.224	/11	/19	/27	8	32	EO
.240	/12	/20	/28	16	16	FO
.248	/13	/21	/29	32	8	F8
.252	/14	/22	/30	64	4	FC
.254	/15	/23	не используется	128	2	FE
.255			/32	0		FF

Бесклассовая адресация используется не только для того, чтобы организовать работу географически удаленных друг от друга сетей. CIDR позволяет изолировать различные департаменты в крупных организациях для того, чтобы обеспечить более надежную защиту (с использованием внутренних брандмауэров), разбить сеть на несколько сегментов и снизить трафик между ними, уменьшив тем самым количество коллизий и снизив время реакции системы.

Использование ipconfig

Утилита ipconfig используется для настройки любых сетевых интерфейсов, установленных на сетевом узле. Однако помимо настройки интерфейса вы можете использовать эту утилиту для получения дополнительных сведений об интерфейсе. Любую информацию об интерфейсе можно модифицировать, однако вы должны учитывать, что не все драйверы поддерживают изменение любых параметров. Некоторые важные сведения о безопасности, отображаемые утилитой ipconfig, обсуждаются позднее. Далее приводится пример вывода этой утилиты, полученный с использованием команды ipconfig -a. Ключ -a заставляет утилиту ipconfig отобразить сведения обо всех интерфейсах, даже о тех, которые не работают. Вот пример вывода утилиты ipconfig:

```
eth0   Link encap:Ethernet HWaddr 00:10:5A:8B:OC:FA
       inet addr:209.127.112.185 Beast:209.127.112.191 Mask=255.255.255.192
       UP BROADCAST RUNNING PROMISC MULTICAST MTU:1500 Metric:!
       RX packets:0 errors:0 dropped:0 overruns:0 frame:0
       TX packets:12 errors:0 dropped:0 overruns:0 carrier:12
       collisions:0 txqueuelen:100
```

```
        Interrupt:9 Base address:0x300

eth0:l  Link encap:Ethernet HWaddr 00:10:5A:8B:OC:FA
        inet addr:207.199.127.84 Beast:207.199.127.255 Mask-.255.255.255.0
        UP BROADCAST RUNNING PROMISC MULTICAST MTU:1500 Metric:1
        Interrupt:9 Base address:0x300

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        UP LOOPBACK RUNNING MTU:3924 Metric:1
        RX packets:3615 errors:0 dropped:0 overruns:0 frame:0
        TX packets:3615 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
pppO    Link encap:Point-to-Point Protocol
        inet addr:208.143.150.16 P-t-P:208.143.150.9 Mask:255.255.255.255
        UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1
        RX packets:3237 errors:5 dropped:0 overruns:0 frame:5
        TX packets:3354 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen: 10

ppp1    Link encap:Point-to-Point Protocol
        inet addr:192.168.10.10 P-t-P:192.168.10.11 Mask:255.255.255.255
         POINTOPOINT NOARP MULTICAST MTU:1500 Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:10
Vmnet0  Link encap:Ethernet HWaddr 00:50:56:80:00:00
        BROADCAST MULTICAST MTU:1500 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:100
vmnetl  Link encap:Ethernet HWaddr 00:50:56:8A:00:00
        BROADCAST MULTICAST MTU:1500 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:100
```

Проанализировав информацию об интерфейсах, вы можете заметить, что интерфейс ethO является интерфейсом инкапсуляции линии связи (Link encap) типа Ethernet. Аппаратный адрес (HWaddr) — это MAC-адрес, о котором мы уже говорили ранее. Аппаратный адрес жестко записан внутри сетевой карты Ethernet, однако при желании вы можете переопределить его из командной строки. Таким образом, если вы знаете IP-адрес и MAC-адрес некоторой системы, вы можете настроить свой компьютер так, чтобы он исполнял роль этой самой системы. Такая необходимость, как правило, возникает в случае сбоев или когда требуется выполнить тестирование. Однако при этом клиенты не должны заимствовать адрес сервера. Чтобы узнать MAC-адрес удаленной системы, можно заглянуть в кэш ARP. Содержимое кэша ARP выводится на экран при помощи команды arp -a. Если вы не видите в кэше информации об интересующей вас системе, попробуйте соединиться с ней при помощи утилиты ping, а затем снова воспользуйтесь командой arp -a. Дело в том, что по истечении некоторого времени записи, к которым за это время не было ни одного обращения, удаляются из кэша ARP. Если вы обратитесь к удаленной системе, соответствующая ей запись вновь появится в кэше. Однако данный прием определения MAC-адреса работает только для локальных сетевых узлов.

Во второй строке, относящейся к интерфейсу ethO, в точечной десятичной нотации отображаются IP-адрес, сетевая маска и широковещательный адрес. Со всеми этими параметрами мы с вами уже хорошо знакомы. Третья строка представляет больший интерес. Прежде всего обратите внимание на запись PROMISC. Это значение указывает на то, что карта Ethernet работает в *режиме прослушивания* сети, который иногда называют также *смешанным режимом* (promiscuous mode). Сетевой интерфейс, работающий в режиме прослушивания, принимает абсолютно всю информацию, которая передается в пределах локального сегмента Ethernet между входящими в этот сегмент сетевыми узлами. Данный интерфейс напоминает человека, который читает чужие письма. В обычном состоянии сетевой интерфейс принимает только те пакеты, которые адресованы непосредственно ему, то есть посланы на соответствующий интерфейсу IP-адрес. Однако карта Ethernet, работающая в режиме прослушивания, принимает абсолютно все пакеты, передаваемые по кабелю Ethernet, то есть даже те, которые ей не адресованы. Переключение карты в такой режим осуществляется специальными программами, например tcpdump. Если интерфейс отмечен меткой PROMISC, значит, в системе работает «подслушивающая» программа. Если вы планируете использовать подобное программное обеспечение в сети, вы должны быть очень осторожны. Некоторые системы до сих пор подвержены таким атакам, как атака «big ping». Если сетевая карта работает в смешанном режиме и через сеть передается сообщение big ping, даже если это

сообщение не адресовано системе, оно будет принято системой, и если система подвержена атаке такого типа, она зависнет. Некоторые системы зависают, даже если только самый последний фрагмент сообщения big ping принимается ими. Более подробно об этом рассказывается в главе 16.

Большая часть всех остальных сведений об интерфейсе является статистическими данными. Однако обратите внимание на последнюю строку в разделе, посвященном интерфейсу ethO. В ней указывается базовый аппаратный адрес и канал прерывания IRQ. Имейте в виду, что если драйвер поддерживает такую возможность, вы можете изменить адрес I/O и номер IRQ, однако на практике в этом редко когда возникает необходимость.

Второй раздел, отображаемый командой ipconfig, соответствует интерфейсу ethO:l. Это псевдоним интерфейса ethO. Иными словами, интерфейсу ethO соответствуют два IP-адреса. Запись, соответствующая псевдониму интерфейса, выглядит в точности как и сам интерфейс, однако статистические данные в ее составе не отображаются, так как вся статистика указывается только для основного интерфейса.

Третьим разделом является раздел lo. Этот раздел соответствует адресу localhost (локальный сетевой узел). Любая система обладает записью localhost, которой соответствует IP-адрес 127.0.0.1. В реальности каждая система обладает диапазоном адресов класса A *с* адресами от 127.0.0.0 до 127.255.255.255. Таким образом, если по какой-либо причине вы запускаете в сети «подслушивающую» программу (sniffer) и обнаруживаете адрес 127.0.0.0/8, имейте в виду, что кто-то выставляет свою систему за какой-то другой сетевой узел. Иными словами, он намеренно изменил свой адрес так, чтобы создавалось впечатление, что данные передаются откуда-либо из другого места (в данном случае с узла localhost). Благодаря этому у него появляется возможность получить права на доступ, которыми он не обладает. Часто это выглядит как атака на брандмауэр, когда некто с другой стороны брандмауэра маскируется под адресом, принадлежащим адресному пространству по эту сторону брандмауэра. Более подробно об этом рассказывается в главе, посвященной сетевым фильтрам.

Утилита ifconfig сообщает нам о существовании двух интерфейсов pppX. Это означает, что компьютер оснащен одним или двумя модемами. Обратите внимание, что интерфейс ppp0 активен, а интерфейс ppp1 — нет. На подобные записи следует обращать внимание в случае, если внутренняя политика запрещает неав-торизированные подключения к Интернету и если бы при этом данный компьютер не был бы авторизирован для установки соединений через телефонную связь.

Два интерфейса vmnet интересны тем, что на самом деле они не существуют. При этом вы можете видеть, что каждому из них соответствует MAC-адрес, а также остальные атрибуты, свойственные любой Ethernet-карте. В настоящий момент ни одному из этих интерфейсов не поставлен в соответствие IP-адрес, однако как только они становятся активными, они получают свой собственный IP-адрес при помощи протокола DHCP (Dynamic Host Configuration Protocol). Такие виртуальные интерфейсы существуют только в виде программного кода и принадлежат некоторому программному пакету. С точки зрения всей остальной внешней сети, когда эти интерфейсы становятся активными, они выглядят как отдельные системы (обратите внимание, что, как правило, существует еще два виртуальных интерфейса таким образом, чтобы общее их количество равнялось четырем, однако я сократил листинг для краткости).

В системе могут существовать также и другие интерфейсы, например ipppX для ISDN PPP и trX для Token Ring. Однако для всех подобных интерфейсов утилита ifconfig отображает приблизительно один и тот же набор сведений.

Использование route

С точки зрения безопасности утилита route не представляет особого интереса, однако, как правило, она используется совместно с ifconfig, поэтому здесь я скажу об этой утилите пару слов. По умолчанию route отображает на экране сведения из таблицы маршрутизации с указанием символьных имен, однако в большинстве случаев IP-адреса оказываются более информативными, поэтому чаще совместно с route используется ключ -n.

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
208.143.150.9	0.0.0.0	255.255.255.255	UH	0	0	0	ppp0
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo
0.0.0.0	208.143.150.9	0.0.0.0	UG	0	0	0	pppO

Наиболее интересной колонкой является колонка Flags, в которой отображается состояние маршрута. Буква U указывает на то, что маршрут активен (Up). Буква H указывает на сетевой узел (Host), а буква G — на шлюз (Gateway). Реже в этой колонке можно видеть буквы R, D и M, они обозначают «восстановленный» (Reinstate), «динамический» (Dynamic) и «модифицированный» (Modified) соответственно. Изредка в этой колонке можно увидеть символ восклицательного знака «!», который означает «отказ» (Reject). *Это вовсе не означает, что на маршруте установлен брандмауэр.* Символом восклицательного знака обозначается маршрут, через который не проходят пакеты. Таким образом, если вы

посылаете пакеты с локального узла узлу или сети, напротив которых стоит символ восклицательного знака, вы получите сообщение о том, что маршрут неактивен (Down). Однако это не означает, что по этому маршруту пакеты не могут передаваться в обратном направлении. Вы сможете принимать оттуда поступающие к вам пакеты UDP. По маршруту нельзя переправлять исходящие пакеты UDP, а также нельзя создать соединение TCP.

Заключение

В данной главе я очень коротко рассказал о сетях и некоторых аспектах сетевой безопасности. Вы узнали о том, как именно системы взаимодействуют друг с другом. Вы также узнали немного об утилите ifconfig и о параметрах, связанных с безопасностью.

10

Стандартные службы

В данной главе рассматриваются следующие вопросы:

- что такое службы;
- - утилита netstat.

Службы являются фундаментальной составляющей сетевой рабочей среды. Если бы не было служб, на любом компьютере можно было бы запускать не более одной программы, к которой клиенты могли бы подключаться через сеть по одному за один раз. Работа служб основана на концепции портов — специальной добавки к IP-адресу, позволяющей более точно идентифицировать метод взаимодействия клиента с сервером. Большинство пользователей знакомы с портами лишь поверхностно, так как при работе с сетью в большинстве случаев достаточно указать лишь имя или IP-адрес узла, а номер порта можно не указывать. Происходит это потому, что при подключении клиентов к серверам почти всегда используются номера портов по умолчанию. Таким образом, пользователь может не беспокоиться о выборе подходящего порта — необходимый порт выбирается автоматически. Однако знание о том, что такое порты и что такое службы, чрезвычайно важно для понимания основных концепций сетевой безопасности.

Что такое службы

Представьте себе некоторую организацию, размещенную в одном большом здании. Пусть эта организация обладает единым телефонным номером, специально предназначенным для обслуживания входящих звонков. Данный телефонный номер способен обслуживать множество поступающих звонков одновременно. Внутри организации поддерживается множество внутренних добавочных телефонных номеров — все они распределены между сотрудниками организации в индивидуальном порядке. Когда кто-нибудь звонит по единому телефонному номеру этой организации, голос автоответчика предлагает сообщить внутренний добавочный номер. После того как звонящий указывает этот внутренний номер, он автоматически соединяется с соответствующим сотрудником организации или с секретарем (если же введенный добавочный номер оказывается некорректным, автоматический автоответчик сообщает звонящему, что введенный им добавочный номер некорректен). Клиентское программное обеспечение работает приблизительно также. Общий единый телефонный номер организации подобен IP-адресу серверного компьютера, а внутренний добавочный номер — это номер порта. На сервере может работать множество серверных программ, каждой из которых соответствует свой собственный порт. Когда вы соединяетесь с сервером по указанному вами IP-адресу (или с использованием доменного имени, что почти одно и то же), ваша клиентская программа сообщает серверу номер порта, с которым она хотела бы соединиться. Номер порта по умолчанию хранится внутри клиентской программы, поэтому пользователю не приходится вводить его самостоятельно. Именно по этой причине многие пользователи и не подозревают о существовании портов.

Итак, для того чтобы установить соединение с сервером, необходимо указать IP-адрес этого сервера и порт, через который будет установлено соединение. Как правило, эти параметры указываются в формате: IP-адрес:Порт. Каждая из серверных программ настроена на использование определенного порта по умолчанию. Клиентская программа, предназначенная для взаимодействия с определенной серверной

программой, хранит внутри себя номер порта по умолчанию для связи с серверной программой. Клиент пытается подключиться к серверу с использованием порта по умолчанию, и если сервер поддерживает подключение через указанный порт, начинается начальная стадия взаимодействия. На этой стадии сервер может потребовать от клиента аутентификации. Если клиент не может аутентифи-цировать себя или аутентифицирует себя неправильно, сервер может разорвать соединение. Однако следует учитывать то, что выполнение аутентификации без установки соединения невозможно. Иными словами, на момент выполнения аутентификации соединение между клиентом и сервером уже установлено.

Благодаря механизму портов вы можете запустить на одном серверном компьютере несколько серверных приложений (служб). Каждому из этих приложений будет соответствовать отдельный порт. Например, вы можете запустить FTP, telnet, HTTP и т. п. в одно и то же время. Если бы портов не было, на каждом компьютере можно было бы запустить только одну службу.

На рис. 10.1 графически показана работа этого механизма.

Когда вы хотите получить от компании какую-либо информацию, вы звоните в компанию. Используя рассмотренную аналогию, вы указываете добавочный телефонный номер отдела, с которым вы хотите связаться. Если вы хотите обсудить покупку продукта, вы соединяетесь с отделом продаж, но не с отделом технического обслуживания. Однако если у вас есть вопрос, связанный с техническим обслуживанием, вы звоните в отдел технического обслуживания, но не в отдел продаж. Каждое подразделение организации выполняет определенную функцию. Сотрудники одного отдела компании могут вообще ничего не знать о каком-либо другом отделе. Возможно, в разговоре по телефону сотрудники разных отделов будут использовать разный жаргон. Та же самая концепция действует и в отношении разных серверных приложений, однако в случае серверных приложений жаргоном является протокол, при помощи которого происходит обмен данными с клиентом.

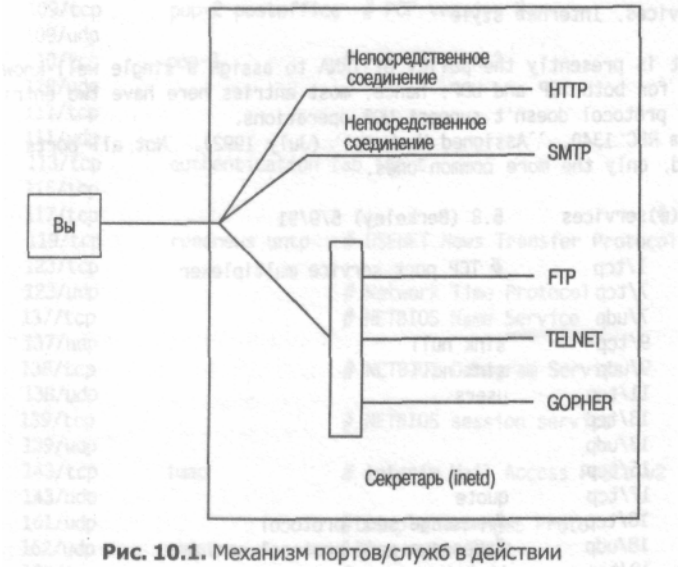

Рис. 10.1. Механизм портов/служб в действии

В отношении компьютеров происходит то же самое. Вы сидите перед компьютером (который выполняет функции клиента), запускаете web-браузер и вводите имя компании, от которой вы хотите получить некоторую информацию. Предположим, вы ввели www.calderasystems.com. Клиент начинает поиск адреса web-узла компании Caldera Systems. В данном случае это будет IP-адрес. Когда IP-адрес обнаружен, браузер пытается установить связь с этим сетевым узлом. Для этого необходимо указать порт, через который будет установлена связь. Известно, что по умолчанию любой web-сервер использует порт 80. Это значение по умолчанию используется всеми серверами Web, так как служба WWW является общеизвестной службой (Well Known Service, WKS) и, как для любой другой общеизвестной службы, ей поставлен в соответствие конкретный стандартный номер порта. Для упрощения процедуры подключения номер порта 80 хранится внутри программы web-браузера и используется по умолчанию, поэтому вам не потребуется вводить его вручную. Вы также не сможете использовать web-клиент (то есть браузер) для соединения с сервером telnet или почтовым сервером, так как эти серверы используют совершенно другой протокол и совершенно другой набор правил коммуникации. Таким образом, коммуникация между web-клиентом и сервером telnet невозможна (позже будет показано, что это утверждение не совсем истинно). Это все равно, как если бы вы соединились с отделом продаж и попытались бы с помощью сотрудников этого отдела решить возникшую у вас техническую проблему. Подобный обмен данными вряд ли был бы полезным для вас.

Теперь, должно быть, вы понимаете, что порт является столь же важной частью коммуникационного процесса, как и IP-адрес. Давайте заглянем внутрь файла /etc/ services, в котором перечислены некоторые общеизвестные службы:

```
#    SNetBSD:   services,v 1.18 1996/03/26 00:07:58 mrg Exp $
#
# Network services, Internet style
#
# Note that it    is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
# Updated from         RFC 1340, "Assigned Numbers" (July 1992). Not all ports
# are included, only the more common ones.
#
#     from: @(#)services   5.8 (Berkeley) 5/9/91
#
tcpmux          1/tcp                   # TCP port service multiplexer
echo            7/tcp
echo            7/udp
discard         9/tcp                   sink null
discard         9/udp                   sink null
systat          11/tcp                  users
daytime         13/tcp
daytime         13/udp
netstat         15/tcp
qotd            17/tcp                  quote
msp             18/tcp                  # message send protocol
msp             18/udp                  # message send protocol
chargen         19/tcp                  ttytst source
chargen         19/udp                   ttytst source
ftp-data        20/tcp                   # default ftp data port
ftp             21/tcp
ssh             22/tcp
ssh             22/udp
telnet          23/tcp
#               24 - private
smtp            25/tcp                  mail
#               26 - unassigned
time            37/tcp                  timserver
time            37/udp                  timserver
rlp             39/udp                   resource        # resource location
nameserver      42/tcp                  name            # IEN 116
whois           43/tcp                  nicname
domain          53/tcp                  nameserver      # name-domain server
domain          53/udp                  nameserver
mtp             57/tcp                                  # deprecated
bootps          67/tcp                  # BOOTP server
bootps          67/udp
bootpc          68/tcp                  # BOOTP client
bootpc          68/udp
tftp            69/udp
gopher          70/tcp                  # Internet Gopher
gopher          70/udp
finger          79/tcp
www             80/tcp                  http            # WorldWideWeb HTTP
www             80/udp                  # HyperText Transfer Protocol
link            87/tcp                   ttylink
kerberos        88/tcp                  krb5            # Kerberos v5
kerberos        88/udp
#               100 - reserved
hostnames       101/tcp                 hostname        # usually from sri-nic
rtelnet         107/tcp                 # Remote Telnet
rtelnet         107/udp
pop2            109/tcp                 pop-2 postoffice        # POP version 2
pop2            109/udp
pop3            110/tcp                 pop-3           # POP version 3
pop3            110/udp
sunrpc          111/tcp
sunrpc          111/udp
auth            113/tcp                 authentication tap ident
sftp            115/tcp
uucp-path       117/tcp
nntp            119/tcp                 readnews untp  # USENET News Transfer Protocol
ntp             123/tcp
ntp             123/udp                                 # Network Time Protocol
netbios-ns      137/tcp                                 # NETBIOS Name Service
netbios-ns      137/udp
netbios-dgm     138/tcp                                 # NETBIOS Datagram Service
```

```
netbios-dgm     138/udp
netbios-ssn     139/tcp                         # NETBIOS session service
netbios-ssn     139/udp
imap2           143/tcp         imap            # Interim Mail Access Proto v2
imap2           143/udp
snmp            161/udp                         # Simple Net Mgmt Proto
snmp-trap       162/udp         snmptrap        # Traps for SNMP
xdmcp           177/tcp                         # X Display Mgr. Control Proto
xdmcp           177/udp
bgp             179/tcp                         # Border Gateway Proto.
bgp             179/udp
irc             194/tcp                         # Internet Relay Chat
irc             194/udp
at-rtmp         201/tcp                         # AppleTalk routing
at-rtmp         201/udp
at-nbp          202/tcp                         # AppleTalk name binding
at-nbp          202/udp
at-echo         204/tcp                         # AppleTalk echo
at -echo         204/udp
at-zis          206/tcp                         # AppleTalk zone information
at-zis          206/udp
z3950           210/tcp         wais            # NISO 239.50 database
z3950           210/udp         wais
wais ipx        213/tcp                         # IPX
ipx             213/udp
imap3           220/tcp                         # Interactive Mail Access
imap3           220/udp                         # Protocol v3
#
# UNIX specific services
#
exec            512/tcp
biff            512/udp         comsat
login           513/tcp
who             513/udp         whod
shell           514/tcp         cmd             # no passwords used
syslog          514/udp
printer         515/tcp         spooler         # line printer spooler
talk            517/udp
ntalk           518/udp
route           520/udp         router routed   # RIP
timed           525/udp         timeserver
tempo           526/tcp         newdate
courier         530/tcp         rpc
conference      531/tcp         chat
#    temporary entry (not officially registered by the Samba Team!)
swat            901/tcp         # Samba Web Administration Tool
#
#    AppleTalk OOP entries (OOP: Datagram Delivery Protocol)
#
rtmp            1/ddp                           # Routing Table Maintenance Protocol
nbp             2/ddp                           # Name Binding Protocol
echo            4/ddp                           # AppleTalk Echo Protocol
zip             6/ddp                           # Zone Information Protocol
```

Чтобы не загромождать книгу, я несколько сократил файл services. Вы должны использовать файл services, который входит в состав используемого вами комплекта поставки Linux. Наиболее свежий перечень общеизвестных служб содержится в документе RFC 1340, однако в будущем этот документ наверняка будет заменен более новой версией.

ССЫЛКА

На прилагаемом к книге компакт-диске в подкаталоге col/security размещается программа под названием rfc, которую можно использовать для того, чтобы найти наиболее свежую версию интересующего вас документа RFC.

Вы можете добавлять в файл services новые службы и новые уникальные номера портов, однако при этом не следует добавлять в этот файл службы, которые уже в нем существуют. Например, вы можете захотеть использовать для службы www порт 8080, однако если вы добавите соответствующую запись в конец файла services, ничего не изменится, так как служба www уже связана с портом 80. Когда система Linux ищет соответствие между именем службы и номером порта, она просматривает этот файл с самого начала в направлении к концу и использует самую первую подходящую запись. Все последующие записи, содержащие подходящее имя службы, игнорируются.

ВНИМАНИЕ

Не следует удалять записи о WKS (Well Known Services) из файла /etc/services. Если вы удалите или закомментируете какую-либо строку, программы, нуждающиеся в соответствующей службе, не смогут к ней обратиться.

Удаление WKS из файла /etc/services — тоже плохая идея. Файл sercices содержит важную информацию о соответствии имен и портов служб, и многие программы и утилиты обращаются к этому файлу. Если вы измените имя или номер порта WKS, вы можете нарушить работоспособность соответствующих служб. Однако вы можете добавлять новые службы и ставить им в соответствие не использующиеся номера портов. Во-первых, благодаря этому утилиты наподобие netstat смогут преобразовывать номера портов в символьные имена служб, а во-вторых, некоторые службы не могут работать в случае, если в файле /etc/services не будет соответствующей записи.

Относительно служб следует сделать важное замечание: в операционной среде Linux все порты с номерами ниже 1024 рассматриваются как привилегированные. Только учетная запись root (UID 0) может выполнить связывание служб с этими портами. Любые неиспользуемые порты с номерами выше 1024 могут использоваться кем угодно. Вообще некоторые «неофициальные» общеизвестные порты есть и в диапазоне выше 1024. Такие порты используются по всеобщему соглашению, а не в рамках жестко заданного стандарта, определенного в RFC, поэтому официальный документ RFC не содержит сведений об этих портах. Одним из таких портов является порт 8080, который часто используется в качестве пользовательского порта web-сервера. Существует также несколько исключений из данного правила. Например, протокол X, тот самый, который используется для функционирования графического интерфейса GUI вашей системы, использует порты с номерами 6000-6010. Это происходит потому, что протокол X не является защищенным, и поэтому для него нет необходимости выделять один из портов из диапазона 0-1024.

Всего доступно для использования 65 536 портов. Номера портов выше 65 535 использовать не следует. Это связано с тем, что для портов существует такая же проблема «зацикливания», как и для идентификаторов UID. Иными словами, для хранения номера порта используется 16 бит. В переменной размером 16 бит можно хранить число от 0 до 65 535. Если вы попытаетесь записать в эту переменную число большее, чем 65 535, то на самом деле в нее запишется число из начала диапазона номеров портов. Например, при попытке записать в переменную число 65 700 на самом деле в переменную запишется число, равное разнице: 65 700 минус 65 536, то есть 164, а это номер из диапазона привилегированных портов. Конечно же, для осуществления подобного обхода системы защиты требуется модификация исполняемого кода системы, однако это известный трюк, который иногда срабатывает.

ПРИМЕЧАНИЕ

Раз мы заговорили об атаках, хочу рассказать вам о том, что некоторые номера портов используются взломщиками в качестве «ключевых слов», или «маркеров». Например, очень часто приходится сталкиваться с портами 31337 или 31173 или какими-либо подобными комбинациями. Попробуйте заменить в этих числах цифру 3 на букву «e», цифру 1 на букву «I» или «i», а цифру 7 на букву «t». Вы увидите вариацию слова «elite» — элита. Если вы обнаружите, что ваша система открыла один из этих портов для обслуживания удаленных клиентов, можете быть уверены в том, что ваша система подверглась атаке и оказалась взломанной.

Некоторые читатели могут подумать, что 65 000 портов — это чрезмерно много. Однако на самом деле это не так. Вспомним описание формирования соединения TCP на основе последовательности SYN-ACK. Клиент обращается к серверу с запросом, сервер сообщает ему случайно выбранный номер порта с номером выше 1024, через который информация будет пересылаться обратно клиенту. Более подробно об этом рассказывается в следующем разделе *«Утилита netstat».* Таким образом, каждое соединение TCP на самом деле использует два порта. Например, если вы имеете дело с 30-ю одновременными подключениями к службе www вашей системы, фактически в системе будет задействован 31 порт: 30 из них будут случайно выбранными портами с номерами выше 1024, а один — порт 80, назначенный службе www по умолчанию. Именно благодаря случайно выбранным портам и порядковым номерам пакетов система получает возможность отслеживать 30 соединений в одно и то же время и обеспечивать их раздельное функционирование. Если вы видите в этом уязвимое место, значит, вы начинаете думать как взломщик — именно так, ставя себя на их место, и следует бороться со злоумышленниками.

Идентифицировать адреса IP и порты, используемые для соединения между двумя системами, — это не такая уж и сложная задача, однако чтобы получить возможность «вставить» себя между двумя системами, взломщик должен угадать код последовательности. Для большинства систем Unix это чрезвычайно сложная задача. Однако некоторые системы для генерации изначального порядкового номера пакета используют функцию, основанную на времени. В этом случае подобрать нужный порядковый номер существенно проще. Атаки такого характера называют *man in the middle* (человек между). К счастью, реализовать подобное на практике удается далеко не каждому, поэтому среднестатистический дилетант с

замашками компьютерного хакера на подобное, скорее всего, не способен.

Утилита netstat

Утилита netstat является наилучшим инструментом анализа состояния сети на любом сетевом узле. Если вы запустите утилиту netstat без каких-либо аргументов, она отобразит на экране состояние всех сетевых соединений вашей системы. По умолчанию все IP-адреса и номера портов будут заменены символьными именами узлов и служб. Для этого будут использованы стандартные процедуры разрешения имен и файл /etc/services.

Отображаемая информация

В данном разделе мы рассмотрим пример вывода утилиты netstat, чтобы вы смогли понять, что означают те или иные отображаемые netstat сведения.

Первая строка, отображаемая утилитой netstat, всегда напоминает вам о том, сведения какого характера отображаются в данный момент этой утилитой. В приведенном далее примере в первой строке указывается: Active Internet connections (w/o servers). То есть в переводе на русский: *активные соединения с Интернетом (без серверов)*. Эта строка может меняться в зависимости от аргументов, передаваемых утилите. Термином servers (серверы) в данном случае обозначаются программы, открывшие порт и ожидающие поступления запросов на соединение. Говорят, что такие программы *«слушают в очереди ожидания»* (wait queue listening), однако при этом они еще не обмениваются данными с клиентом. Таким образом, в данном случае отображается информация только о существующих соединениях.

```
Active Internetconnections (w/o servers)
Proto Recv-Q Send-Q Local Address      Foreign Address          State
tcp    0      0 pppl5.chi.pty.com:2034 209.207.224.222:www  ESTABLISHED
tcp    0      0 pppl5.chi.pty.com:2033 209.207.224.222:www  ESTABLISHED
tcp    0      0 pppl5.chi.pty.com:2032 209.207.224.222:www  ESTABLISHED
tcp    0      0 ppp!5.chi.pty.com:2031 209.207.224.222:www  ESTABLISHED
tcp    1      1 pppl5.chi.pty.com:2030 209.207.224.245:www  CLOSING
tcp    0      0 pppl5.chi.pty.com:2026 mail03.dfw.mindspr:pop3 TIME_WAIT
Active UNIX domainsockets (w/o servers)
Proto        RefCnt Flags      Type          State    I-Node Path
unix  1              [ ]       STREAM        CONNECTED      26740      00000013f
unix  1              [ ]       STREAM        CONNECTED      652    000000004
unix  1  [N]         STREAM    CONNECTED 27587  000000162
unix  1              [ ]       STREAM        CONNECTED      26744      000000140
unix  1              [ ]       STREAM        CONNECTED      642    000000003
unix  1              [ ]       STREAM        CONNECTED      28666      /tmp/.Xll-unix/XO
unix  1              [ ]       STREAM        CONNECTED      27588      /tmp/.Xll-unix/XO
unix  1              [ ]       STREAM        CONNECTED      27527      /tmp/.Xll-unix/XO
unix  0              [ ]                     STREAM      27523
unix  1              [ ]       STREAM        CONNECTED      26760      /tmp/.Xll-unix/XO
unix  1              [ ]       STREAM        CONNECTED      26754      /tmp/.Xll-unix/XO
unix  1              [ ]       STREAM        CONNECTED      653    /dev/log
unix  1              [ ]       STREAM        CONNECTED      643    /dev/log
unix  1              [ ]       STREAM        CONNECTED      597    /dev/log
```

Вторая строка — это заголовки для первого раздела вывода netstat. Заголовок Proto обозначает *protocol* — протокол. Чаще всего в этой колонке фигурирует либо tcp, либо udp, однако, как было отмечено в файле /etc/services, в этой колонке могут быть также raw и другие протоколы. Третья и четвертая колонки обладают заголовками Revc-Q и Send-Q, что означает Receive Queue (очередь на прием) и Send Queue (очередь на передачу) соответственно. В этих колонках показывается соответственно количество пакетов, принятых из сети, но не скопированных пользовательской программой, подключенной к сокету, и количество байтов, прием которых не подтвержден удаленной системой. После того как сокет TCP закрывается, как правило, удаленный сетевой узел не подтверждает приема одного пакета. Это происходит потому, что соединение разрывается и завершающий ACK не может быть подтвержден.

В следующих двух колонках указывается локальный (Local) и удаленный (Foreign) IP-адреса, между которыми устанавливается соединение. Адреса указываются в форме адрес:порт. В рассматриваемом выше примере узел pppl5.chi.pty.com:2034 (полное доменное имя иногда обрезается, чтобы уместиться в отведенном для этого пространстве) соединен с узлом 209.207.224.222:www, то есть через порт 80. Протокол TCP подключает сервер к клиенту таким образом, что данные посылаются по одному порту (в данном случае с номером 80) и принимаются по другому порту (в данном случае с номером 2034). Порт

возврата данных не задан жестко. Это может быть любой незанятый порт с номером выше 1024.

В процессе первого соединения клиент сообщает серверу номер выбранного порта и ожидает поступления на этот порт пакета SYN-ACK, принятого от сервера. Если сервер не может использовать этот порт, потому что он занят, клиенту передается сообщение ICMP Type 3 Code 3 Port Unreachable (порт недоступен). При этом клиент случайным образом выбирает другой порт. Когда клиент через выбранный им порт наконец принимает от сервера пакет SYN, он возвращает серверу ACK. На этом соединение считается установленным и начинается передача данных.

В приведенном ранее примере вывода netstat видно, что с удаленным узлом установлено несколько соединений. При этом соединение через клиентский порт 2030 было установлено самым первым. Судя по всему, на загруженной с сайта web-страничке располагалось несколько рекламных баннеров, в связи с чем браузер был вынужден открыть несколько дополнительных соединений, чтобы загрузить необходимые картинки. Каждое соединение устанавливается в результате выполнения ранее рассмотренной последовательности процедур, в результате которой открывается очередной коммуникационный порт. Когда соединение разрывается, осуществляется похожая последовательность процедур. В последней колонке одной из записей указывается метка CLOSING (соединение закрывается), однако вместо этого в данной колонке может содержаться также метка LAST_ACK (последнее подтверждение). Когда клиент посылает сообщение LAST_ACK, а сервер принимает его, сервер разрывает соединение. Так как соединение больше не существует, ответ на это сообщение передать невозможно.

В последней колонке, в которой указывается состояние соединения, может быть одна из следующих меток:

ESTABLISHED
SYN_SENT
SYN_RECV
FIN_WAIT1
FIN_WAIT2
TIME_WAIT
CLOSED
CLOSE_wAIT
LAST_ACK
LISTEN
CLOSING
UNKNOWN

Какие-то из этих меток могут использоваться только на стороне сервера, другие — только на стороне клиента, однако некоторые метки могут применяться как на сервере, так и на клиенте.

Если к команде netstat добавить ключ -e, к выводу команды добавится еще одна колонка User (пользователь). В этой колонке будет указываться идентификатор UID пользователя, от имени которого запущен процесс, создавший данное соединение.

В начале следующего раздела, отображаемого на экране утилитой netstat, стоит строка Active UNIX domain sockets (w/o servers), что означает *активные доменные сокеты Unix (без серверов)*. После этого идет строка заголовков колонок. Обратите внимание, что данный раздел вывода утилиты netstat существенно сокращен. Заголовки колонок выглядят следующим образом: Proto (протокол), RefCnt (счетчик ссылок), Flags (флаги), Type (тип), State (состояние), I-Node (дескриптор inode) и Path (путь).

Рассмотрим смысл каждой из колонок.

- Proto — обозначает *protocol,* то есть «протокол». Для сокетов в стиле Unix единственным допустимым значением в данной колонке может быть unix.

- RefCnt — счетчик ссылок. Здесь указывается количество подсоединенных процессов. Обычно это значение должно быть равно 1, однако может быть равно и 0.

- Flags — набор флагов. Как правило, это поле должно быть пустым, однако если поле RefCnt равно 0 и соответствующие процессы ожидают поступления запросов на соединение, поле флагов может быть равно ACC (то есть SO_ ACCEPTION — принятие), что означает, что сокет готов к приему запроса на соединение. В некоторых ситуациях могут возникать и другие флаги, например W (SO_ WAITDATA - ожидание данных) или N (SO_NOSPACE - нет места).

- Type - тип сокета. Как правило, здесь стоит метка STREAM (сокет с созданием

соединения), однако также допускаются следующие метки: DGRAM *(Datagram* -сокет без создания соединения), RAW (сокет передачи данных без транспортного протокола), RDM *(Reliably Delivered Message* — надежно передаваемое сообщение), SEQPACKET *{Sequental Packet* — последовательно передаваемый пакет), PACKET (пакет простого доступа к интерфейсу) или UNKNOWN (неизвестный тип сокета для будущих усовершенствований).

- State — состояние сокета. Здесь могут использоваться следующие метки: FREE (сокет свободен),

LISTENING (ожидание поступления запроса), CONNECTING (соединение устанавливается), CONNECTED (соединение установлено), DISCONNECTING (соединение разрывается) или UNKNOWN (неизвестное состояние). Данное поле может быть также вообще пустым. При нормальном функционировании системы сокет не может находиться в состоянии UNKNOWN.

- I-Node — не представляет интереса. Здесь отображается номер дескриптора I-Node, соответствующий соединению. Однако этот I-Node существует в каталоге /proc только в случае, если соединение используется, таким образом, поиск данного I-Node не приведет к желаемым результатам.

- Path — здесь отображается процесс, подключенный к данному сокету.

В предыдущем примере не показано ни одного соединения, находящегося в одном из наиболее распространенных состояний: LISTENING (слушание). Это состояние соответствует ожиданию запроса на подключение. Фактически соединения еще нет — серверная программа ожидает поступления от клиентов запросов на подключение. Информация о соединениях в состоянии LISTENING отображается утилитой netstat только в случае, если вы используете ключ -a. При использовании этого ключа первая строка в каждом из двух разделов меняется. Там, где раньше был отображен текст Active Internet connection (w/o servers), то есть без серверов, теперь стоит текст Active Internet connection (including servers), то есть включая серверы. Соответственно для второго раздела сведений, отображаемых утилитой netstat, заголовок меняется на Active UNIX domain sockets (including servers) — *активные сокеты доменов Unix (включая серверы)*. При этом на экране будет показан существенно более длинный список, так как теперь утилита netstat отображает также информацию о серверных программах, которые не соединены ни с одним из клиентов, но ожидают поступления запроса на подключение.

```
Active Internet connections (servers and  established)
Proto Recv-Q Send-Q Local AddressForeign Address    State
tcp   1      0 208.143.150.24:2034     209.207.224.222:80     CLOSE_WAIT
tcp   1      0 208.143.150.24:2033     209.207.224.222:80     CLOSE_WAIT
tcp   1      0 208.143.150.24:2032     209.207.224.222:80     CLOSEJJAIT
tcp   1      0 208.143.150.24:2031     209.207.224.222:80     CLOSE_WAIT
tcp   1      1 208.143.150.24:2030     209.207.224.245:80     CLOSING
tcp   0          0 0.0.0.0:6000                0.0.0.0:*     LISTEN
tcp   0          0 0.0.0.0:51966               0.0.0.0:*     LISTEN
tcp   0          0 0.0.0.0:3306                0.0.0.0:*     LISTEN
tcp   0          0 0.0.0.0:80                  0.0.0.0:*     LISTEN
tcp   0          0 0.0.0.0:1024                0.0.0.0:*     LISTEN
tcp   0          0 0.0.0.0:10000               0.0.0.0:*     LISTEN
tcp   0          0 0.0.0.0:25                  0.0.0.0:*     LISTEN
tcp     0    0 0.0.0.0:111       0.0.0.0:*                   LISTEN
tcp     0    0 0.0.0.0:901       0.0.0.0:*                   LISTEN
tcp     0    0 0.0.0.0:79                  0.0.0.0:*         LISTEN
tcp     0    0 0.0.0.0:143       0.0.0.0:*                   LISTEN
tcp     0    0 0.0.0.0:110       0.0.0.0:*                   LISTEN
tcp     0    0 0.0.0.0:21                  0.0.0.0:*         LISTEN
tcp     0    0 0.0.0.0:37                  0.0.0.0:*         LISTEN
tcp     0    0 0.0.0.0:19                  0.0.0.0:*         LISTEN
tcp     0    0 0.0.0.0:13                  0.0.0.0:*         LISTEN
tcp     0    0 0.0.0.0:9                   0.0.0.0:*         LISTEN
tcp     0    0 0.0.0.0:7                   0.0.0.0:*         LISTEN
udp     0    0 0.0.0.0:1025      0.0.0.0:*
udp     0    0 0.0.0.0:177     . 0.0.0.0:*
udp     O    O 0.0.0.0:37        0.0.0.0:*
udp     O    O 0.0.0.0:19        0.0.0.0:*
udp     O    O 0.0.0.0:13        0.0.0.0:*
udp     O    O 0.0.0.0:9                   0.0.0.0:*
udp     O    O 0.0.0.0:7                   0.0.0.0:*
raw     O    O 0.0.0.0:1                   0.0.0.0:*         7
raw     0    0 0.0.0.0:6                   0.0.0.0:*         7
Active UNIX domain sockets (servers and established)
Proto                    RefCnt Flags    Type    State     I-Node Path
unix                     0    [ ACC ]   STREAM      LISTENING   593  /dev/log
unix                     1    [ ]     STREAM     CONNECTED   26740 00000013f
unix                     1    [ ]     STREAM     CONNECTED   25943 000000137
unix                     0    [ ACC ]   STREAM      LISTENING   3864  /tmp/.axnetipc
unix                     0    [ ACC ]   STREAM      LISTENING   670   /dev/gpmctl
unix                     0    [ ACC ]   STREAM      LISTENING   17104 /tmp/.Xll-unix/XO
unix                     1    [ ]     STREAM     CONNECTED   25957 00000013a
```

```
unix                    0   [ACC]   STREAM LISTENING   1032  /tmp/mysql.sock
unix                    1   [ N ]    STREAM   CONNECTED   27526 000000160
unix                    1   [ ]     STREAM      CONNECTED   26753 000000142
unix                    1   [ ]     STREAM     CONNECTED   652  000000004
unix                    1   [ N ]    STREAM   CONNECTED   27587 000000162
unix                    1   [ ]     STREAM      CONNECTED   26744 000000140
Unix                    1   [ ]     STREAM      CONNECTED   642   000000003
unix                    1   [ ]     STREAM     CONNECTED   28666 /tmp/.Xll-unix/XO
unix                    1   [ ]     STREAM     CONNECTED   27588 /tmp/.Xll-unix/XO
unix                    1   [ ]     STREAM     CONNECTED   27527 /tmp/.Xll-unix/XO
unix                              0   [ ]    STREAM       27523
unix                    1   [ ]     STREAM     CONNECTED   26760 /tmp/.Xll-unix/XO
unix                    1 •[ ]     STREAM     CONNECTED   653  /dev/log
unix                    1   [ ]     STREAM     CONNECTED   643  /dev/log
unix                    1   [ ]     STREAM     CONNECTED   597  /dev/log
```

Изучите этот список и сравните вывод утилиты netstat со списком процессов, запущенных в вашей системе. Вы можете обнаружить, что некоторые из процессов, ожидающих поступления запроса на соединение, не обладают соответствующими им демонами, связанными с некоторым портом. Но как программа может ожидать поступления запроса, если она при этом даже не загружена в память? Все очень просто: ожиданием поступления запроса для этой программы на самом деле занимается метадемон inetd, который порождает демонов для подобных портов тогда, когда в этом возникает необходимость. В частности, это относится, например, к порту telnet.

ССЫЛКА

Метадемон inetd обсуждается в следующей главе 11.

Если утилиту netstat запустить с ключом -v, вы получите сведения о протоколах, которые не поддерживаются данной системой. Последние несколько строчек вывода будут выглядеть следующим образом:

```
netstat: no support for 'AF IPX' on this system.
netstat: no support for 'AF AX25' on this system.
netstat: no support for 'AF NETROM' on this system.
```

Из этих строк можно сделать вывод, что данная система не поддерживает протоколы IPX, AX25 и NETROM.

Проанализировав сведения, полученные в результате выполнения команды netstat -a, вы сможете быстро понять, какие из портов подготовлены к получению запросов, адресованных вашей системе. Любые порты, для которых нет соответствующей записи в файле /etc/services, будут обладать процессом, ожидающим поступления запроса через этот порт. Этот процесс будет обладать индивидуальной записью в таблице процессов. В файл /etc/services вовсе не обязательно включать записи абсолютно для всех запускаемых на вашем компьютере служб, однако описав все используемые вами службы в файле /etc/services, вы предотвратите возникновение борьбы за порт между несколькими службами. Если некоторый порт уже упомянут в файле /etc/services и в этом файле ему поставлена в соответствие некоторая служба, ни одна другая служба не будет использовать этот порт.

Если вы обнаруживаете некоторый открытый порт и не знаете, зачем он нужен и кто с ним связан, вы должны попробовать выяснить, кем используется этот порт. Это может быть программа из категории «троянский конь» или какое-либо другое скрытое средство удаленного доступа. Все подобные программы, как правило, используют один из произвольных свободно доступных портов.

Если порт закрыт (то есть через него не осуществляется соединение и он не используется для ожидания поступления запросов), этот порт нельзя использовать для входа в систему. Если ни один из портов не используется для ожидания поступления запросов, соединение не может быть установлено извне и взломщик будет вынужден оставить вас в покое. Но если на вашем компьютере не открыт ни один порт, это означает, что на вашем компьютере не работает ни одна сетевая служба.

Вы не можете остановить работу абсолютно всех служб — многие из них необходимы или желательны. В последующих главах будет обсуждаться, как именно вы должны организовать безопасное использование открытых портов.

Заключение

В данной главе приведено краткое описание общеизвестных служб. Также я рассказал вам о том, что такое порты, как они связаны со службами и зачем они вообще нужны. В главе я также рассмотрел утилиту

netstat и то, как с ее помощью можно определить порты, используемые в настоящий момент для обмена данными и открытые в ожидании поступлений запросов на подключение. Именно такие порты используются взломщиками для проникновения в вашу систему.

11

inetd, inetd.conf и сетевые атаки

В данной главе рассматриваются следующие вопросы:
- что такое inetd;
- работа с inetd.conf;
- защита от сетевых атак;
- что такое *honey pot* (горшок меда).

В предыдущей главе уже упоминалось, что система обладает способностью в случае необходимости автоматически запускать серверные программы, которые не были запущены в процессе инициализации системы и не упоминаются в таблице процессов до тех пор, пока кто-либо извне не подключится к ним. Для многих новичков в мире Linux эта возможность является полнейшей загадкой. Каким образом удаленный клиент может обратиться к серверной программе, которая не запущена на сервере? В данной главе вы узнаете об этом, а также поймете, как можно модифицировать поведение системы таким образом, чтобы предотвратить запуск нежелательных служб. В первой части главы рассказывается о том, как происходит запуск служб, которые не запускаются в процессе инициализации, и как можно запретить подобный запуск.

Во второй части главы обсуждаются сетевые атаки. Интернет — это срез нашего общества. Эта среда является как дружественной, так и враждебной, так как в ней, как и в обществе, встречаются совершенно разные индивидуумы. Вы также должны учитывать, что Интернет более глобален, чем общество в радиусе 100 и даже 1000 миль вокруг вас. В Интернете действуют враждебные вам правительства, враждебные вам организации и враждебные вам люди.

inetd

Чтобы понять, как работает inetd, проанализируем вывод команды netstat -a. Интересующие нас элементы этого вывода показаны в листинге 11.1.

Листинг 11.1. Сокращенный вывод команды netstat-a

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local AddressForeign Address          State
tcp        0      0 *:6000                  *:*            LISTEN
tcp        0      0 *:3306                  *:*            LISTEN
tcp        0      0 *:www                   *:*            LISTEN
tcp        0      0 *:10000                 *:*            LISTEN
tcp        0      0 *:smtp                  *:*            LISTEN
tcp        0      0 *:sunrpc                *:*            LISTEN
tcp        0      0 *:swat                  *:*            LISTEN
tcp        0      0 *:finger                *:*            LISTEN
tcp        0      0 *:imap2                 *:*            LISTEN
tcp        0      0 *:pop3                  *:*            LISTEN
tcp        0      0 *:ftp                   *:*            LISTEN
```

```
tcp     0      0 *:time                  *:*                     LISTEN
tcp     0      0 *:chargen               *:*                     LISTEN
tcp     0      0 *:daytime               *:*                     LISTEN
tcp     0      0 *:discard               *:*                     LISTEN
tcp     0      0 *:echo                  *:*                     LISTEN
udp     0      0 *:xdmcp                 *:*
udp     0      0 *:sunrpc                *:*
udp     0      0 *:time                  *:*
udp     0      0 *:chargen               *:*
udp     0      0 *:daytime               *:*
udp     0      0 *:discard               *:*
udp     0      0 *:echo                  *:*
raw     0      0 *:icmp                  *:*                     7
raw     0      0 *:tcp                   *:*                     7
```

Вначале внимание следует обратить на строки, в которых фигурируют не имена служб, а номера портов. Можно видеть, что в системе используются порты с номерами 6000, 3306 и 10000. Почему netstat отображает числа, а не имена служб? Дело в том, что соответствующие порты и службы не упоминаются в файле /etc/services, поэтому netstat отображает их в виде номеров портов.

Попытаемся определить, какие серверные программы используют нумерованные порты. Первым таким портом является порт 6000. В предыдущей главе говорилось, что порты от 6000 до 6010 используются X-сервером. Вы знаете, что в данный момент на вашем компьютере запущен X-сервер, поэтому данная запись не должна быть причиной для вашего беспокойства. Порт 3306 используется mysql Так как mysql всегда использует порт с таким номером, вы можете внести соответствующую запись в файл /etc/services, благодаря чему netstat будет снабжать вас более информативным выводом. Наконец, порт 10000 используется программой webmin. И вновь, если вы всегда выделяете этот порт для использования программой webmin, вы можете просто занести сведения об этом в файл /etc/services, тогда в следующий раз вам не придется вспоминать об этом.

ПРИМЕЧАНИЕ

В каждый момент времени некоторый порт может быть связан только с одним процессом. Таким образом, каждый из перечисленных портов обладает одним и только одним связанным с ним процессом.

Все остальные службы в листинге 11.1 отображаются с указанием их символьных имен, так как все они упоминаются в файле /etc/services. Но если вы сравните список серверов, ожидающих соединения, и список серверов, перечисленных в таблице функционирующих процессов, вы обнаружите, что фактически запущенных в системе серверов не хватает. Вы сможете обнаружить сервер, соответствующий порту 80, то есть web-сервер, который в таблице процессов фигурирует под именем httpd. На самом деле вы обнаружите несколько процессов httpd. Вы также обнаружите, что в системе функционирует процесс sendmail, который связан с портом 25 (smtp). Соответствие некоторых других серверов также очевидно. Однако в списке функционирующих процессов нет программ, которые соответствовали бы таким службам, как swat, finger, imap2, pop3, ftp и проч. Никаких процессов, соответствующих этим службам, в настоящий момент в памяти компьютера не находится. Каким же образом все эти службы могут ожидать поступления запросов на подключение через определенный порт, если они даже не загружены в память? Попробуйте запустить клиента FTP и подключиться к серверу FTP, якобы ожидающему поступления запроса на подключение на локальном узле. Сервер откликнется на ваш запрос и поприветствует вас. Пока клиент ftp активен, посмотрите на таблицу запущенных процессов. Вы увидите, что в ней появилась запись, соответствующая FTP-серверу.

ССЫЛКА

Процессы httpd подробнее обсуждаются в главе 20 «Установка и запуск защищенного web-сервера Apache».

Что же произошло? На самом деле за поступлением запросов на подключение через все подобные порты следит программа inetd. Когда через один из этих портов поступает запрос на подключение, программа inetd запускает соответствующую серверную программу. Благодаря такому подходу экономятся ресурсы, так как отпадает необходимость постоянно держать в памяти большое количество редко используемых серверных приложений — всего одна небольшая программка следит за поступлением запросов и запускает необходимые для обслуживания клиентов приложения по мере необходимости. Именно поэтому, возможно, вы обратите внимание на небольшую временную задержку между моментом отправления запроса на подключение и появлением на экране приглашения на ввод пароля. Эта задержка проявляется, по крайней мере, в первый раз: в последующем задержка обычно значительно короче. Информация о том, за какими портами следит inetd и какие серверные программы следует запускать в случае поступления запроса через какой-либо из этих портов, содержится в файле /etc/inetd.conf. Отредактировав этот файл, вы сможете настроить работу inetd в соответствии со своими предпочтениями.

СОВЕТ

Далеко не все службы могут быть запущены и должны запускаться при помощи inetd. Для загрузки и инициализации некоторых служб требуется значительное время. Если вы будете использовать inetd для запуска подобных служб, за время их инициализации вы рискуете потерять соединение с клиентом. Прежде чем использовать какое-либо серверное приложение совместно с inetd, внимательно ознакомьтесь с документацией этого приложения.

Не все службы могут быть запущены с использованием inetd, поэтому все те службы, использование которых совместно с inetd оправдано и рекомендуется, уже упомянуты в файле inetd.conf. Многие из этих служб, скорее всего, окажутся для вас ненужными, поэтому вы можете просто отключить их.

Работа с inetd.conf

Чтобы получить возможность запуска служб с использованием inetd, вы должны понимать внутреннее строение файла inetd.conf. Это конфигурационный файл, который используется для включения или отключения служб, соответствующих портам, не связанным ни с одним из работающих в системе серверных приложений. В этом файле перечисляются службы, запускаемые с использованием inetd, а также указывается, какие действия должен предпринять демон inetd в случае, если к одной из этих служб обращается удаленный клиент.

Содержимое файла inetd.conf показано в листинге 11.2 (комментарии переведены на русский язык). Это достаточно сложный конфигурационный файл, содержащий в себе таблицу, каждая запись которой состоит из семи полей.

Листинг 11.2. Содержимое файла /etc/inetd.conf, используемого в составе Caldera OpenLinux no умолчанию

```
#
#   inetd.conf  В этом файле описываются службы, запуск которых осуществляется
#               при помощи суперсервера INETD TCP/IP. Чтобы перенастроить
#               работающий процесс INETD, отредактируйте данный файл, а затем
#               передайте процессу INETD сигнал SIGHUP.
#

# Версия:               @(#)/etc/inetd.conf             3.10            05/27/93
#
# Авторы: Изначальный код позаимствован из BSD UNIX 4.3/TAHOE.
#       Fred N. van Kempen, <waltje@uwalt.nl.mugnet.org>
#
# Модификацию для Debian Linux выполнил Ian A. Murdock <imurdock@shell.portal.com>
#
# Модификацию для RHS Linux выполнил Marc Ewing <marc@redhat.com>
#
# Дальнейшую модификацию выполнил 0laf Kirch <okir@caldera.com> для Caldera Open Linux
#
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
# Echo, discard, daytime, and chargen используются в основном для тестирования.
#
# Для активации внесенных в файл изменений выполните 'killall -HUP inetd'
#
# Замечание: встроенные службы UDP теперь отбрасывают пакеты для портов < 512.
echo   stream tcp  nowait  root  internal
echo        dgran   udp         wait    root  internal
discard     stream  tcp         nowait root  internal
discard     dgram   udp         wait    root  internal
daytime     stream  tcp         nowait root  internal
daytime     dgram   udp         wait    root  internal
chargen     stream  tcp         nowait root  internal
chargen     dgram   udp         wait    root  internal
time        stream  tcp         nowait root  internal
time        dgram   udp         wait    root  internal
#
#   Это                 стандартные                 службы.
#
ftp     stream tcp  nowait root  /usr/sbin/tcpd in.ftpd -l -a #telnet  stream tcp  nowait root  /usr/sbin/tcpd  in.telnetd
#gopher  stream tcp  nowait root  /usr/sbin/tcpd  gn #
• smtp следует включать, только если только вы *действительно* знаете, что делаете.
• теперь smtp обслуживается демоном sendmail, но не smtpd. Этот демон
• запускается не отсюда, он запускается во время загрузки из файла /etc/rc.d/rc#.d.
#smtp  stream  tcp  nowait root  /usr/bin/smtpd  smtpd
#nntp  stream  tcp  nowait  root  /usr/sbin/tcpd  in.nntpd I
```

```
# Shell, login, exec и talk являются протоколами BSD.
#
#shell stream tcp nowait root /usr/sbin/tcpd in.rshd #login stream tcp nowait root /usr/sbin/tcpd in.rlogind #exec
stream tcp nowait root /usr/sbin/tcpd in.rexecd #talk dgram udp wait nobody.tty /usr/sbin/tcpd in.talkd #ntalk dgram
udp wait nobody.tty /usr/sbin/tcpd in.ntalkd #dtalk stream tcp wait nobody.tty /usr/sbin/tcpd in.dtalkd #
# Почтовые службы pop и imap
I
#pop2 stream tcp nowait root /usr/sbin/tcpd ipop2d pop3 stream tcp nowait root /usr/sbin/tcpd ipop3d imap
stream tcp nowait root /usr/sbin/tcpd imapd t
# Служба UUCP для Интернета.
#
#uucp stream tcp nowait uucp /usr/sbin/tcpd /usr/sbin/uucico -l f
• Служба tftp используется в основном для загрузки ОС. В большинстве случаев
• эта служба запускается только на специальных "серверах загрузки".
• Не включайте эту службу, если она вам не нужна.
#
#tftp dgram udp wait root /usr/sbin/tcpd in.tftpd #bootps dgram udp wait root /usr/sbin/tcpd bootpd f
# cfinger нужен для GNU finger, который в комплекте RHS Linux не используется
I
finger stream tcp nowait nobody /usr/sbin/tcpd in.fingerd -u #cfinger stream tcp nowait root /usr/sbin/tcpd in.cfingerd
#
• Finger, systat и netstat передают посторонним информацию о пользователях
• эти данные могут использоваться "взломщиками системы".
• Зачастую все или некоторые из этих служб отключаются для улучшения зашиты.
#
#systat stream tcp nowait nobody /usr/sbin/tcpd /bin/ps -auwwx #netstat stream tcp nowait nobody /usr/sbin/tcpd
/bin/netstat --inet
#
# Аутентификация
#
#auth stream tcp nowait root /usr/sbin/tcpd in.identd -t120
swat stream tcp nowait.400 root /usr/sbin/tcpd swat
#
# Конец файла inetd.conf
```

Рассмотрим предназначение каждого из полей файла inetd.conf подробнее.

- Первое поле содержит имя службы. Это имя должно соответствовать имени, указанному в первом поле файла /etc/services.

- Во втором поле указывается тип сокета. Здесь можно указать одно из значений: stream, dgram, raw, rdm или seqpacket в зависимости от того, сокет какого типа используется сервером: поточный, дейтаграмма, чистый IP, надежно доставляемое сообщение или пакет, передаваемый в последовательности. В общем случае пакеты TCP передаются через сокеты типа stream, а пакеты UDP передаются через сокеты типа dgram. Настраивая службу на совместную работу с inetd, вы должны использовать одно из этих значений, если только прилагаемые к службе инструкции не предписывают какой-либо иной вариант настройки.

- В третьем поле указывается тип протокола. Здесь может стоять одно из следующих значений: tcp, udp, rpc/tcp или rpc/udp. Это поле должно соответствовать типу протокола, указанному для данной службы в файле /etc/services.

- В четвертом поле может стоять одна из меток: wait (ждать) или nowait (не ждать). Как правило, службы UDP используют метку wait, а службы TCP используют метку nowait. Одним из исключений является служба TFTP. Эта служба использует пакеты UDP, однако она создает псевдосоединение и поэтому использует метку nowait для того, чтобы избежать возникновения скоростных условий (race conditions), о которых рассказывается на следующей врезке. Метка nowait позволяет нескольким экземплярам серверного процесса «наслаиваться» друг на друга. В этом случае серверный процесс не завершается немедленно, а продолжает работать как привилегированный процесс. Метка wait подразумевает, что псевдосоединение разрывается каждый раз после приема очередного пакета, поэтому при поступлении каждого очередного запроса требуется аутентификация. В противном случае процесс, продолжающий ожидание в очереди, мог бы принять это псевдосоединение за предыдущего пользователя. Метки wait и nowait сообщают серверу, как следует реагировать на пакеты, поступающие через порт. Службы, использующие wait, являются однопоточны-ми и остаются в очереди ожидания до тех пор, пока их работа не завершается. Службы, использующие nowait, устанавливают соединение и освобождают очередь ожидания, продолжая при этом обмен данными с клиентом. Благодаря этому другой сервер может продолжить ожидание поступления запросов через порт. В данном поле можно указать необязательный числовой аргумент, например: nowait.400. Это число определяет максимальное количество серверных процессов, которое может быть загружено в память в течение 60-секунд-ного периода. Если этот параметр не указывается, по умолчанию используется значение 40.

- В четвертом поле указывается пользователь, от имени которого демон inetd запускает соответствующую службу. Благодаря этому вы можете запустить серверный процесс на уровне привилегий более низком, чем root В поле пользователя при желании можно указать группу в формате пользователь.группа. Если группа не указывается, используется первичная группа для указанного пользователя.

- В последних двух полях указывается серверная программа и ее аргументы. Благодаря тому, что для запуска программы и указания аргументов используется два отдельных поля, вы можете запустить программу-оболочку TCP (TCP Wrapper) и в качестве аргумента указать собственно запускаемый сервер. В нашем примере по умолчанию используется tcpd — демон tcpwrappers, который используется для вызова необходимого демона.

ССЫЛКА

Более подробно о программах-оболочках TCP (TCP Wrappers) рассказывается в главе 15.

Скоростные условия

Термин «скоростные условия» (race conditions) часто встречается в компьютерной литературе. Такие условия возникают там, где имеют место две составляющих: во-первых, несколько команд выполняются одна за другой, во-вторых, используются команды, в процессе выполнения которых эффективный пользовательский ID меняется на 0.

Например, если работая с файловой системой вы отдаете команду mkdir, эта команда в процессе своего исполнения меняет пользовательский ID на ноль, чтобы создать каталог. Без этого не обойтись, так как для того, чтобы создать файл (или каталог), команда вынуждена напрямую работать с файловой системой, а файловая система принадлежит пользователю root. Поэтому создание каталога (а это многоэтапный процесс) происходит в момент, когда эффективный пользовательский ID становится равным 0. Когда каталог уже создан, права на его владения должны быть изменены таким образом, чтобы этот каталог принадлежал пользователю, инициировавшему исполнение команды mkdir. На самом деле это два отдельных легко идентифицируемых процесса, которые выполняются в результате обращения к mkdir.

Теперь представьте, что система испытывает сильную нагрузку и пользователь обращается к mkdir с использованием nice, чтобы замедлить процесс еще больше. Предположим также, что сразу же за командой mkdir в сценарии командной оболочки стоит команда удаления каталога и создания ссылки на /etc/shadow. Если злоумышленнику удастся создать условия, при которых каталог удаляется и ссылка создается еще до того, как команда mkdir производит создание каталога и перенастраивает права на владение, тогда получается, что пользователь, изначально запустивший программу, становится владельцем файла /etc/shadow.

Таким образом, скоростные условия возникают в ходе выполнения процесса, являющегося цепочкой событий, в ходе выполнения которых эффективный UID изменяется на 0, при этом данная цепочка выполняется на протяжении определенного времени и параллельно с другими процессами. Иными словами, процесс А обращается к процессу В, который модифицирует результат выполнения процесса А, и при этом между процессами протекает определенное время. Конечно, для возникновения скоростных условий требуется выполнение некоторых других условий, однако подробности в данной книге не обсуждаются.

Взглянув на листинг файла /etc/inet.conf, вы можете заметить, что строка, соответствующая службе ftp, активна. Чтобы запретить использование ftp, просто поставьте в начале этой строки символ «#», после этого перезапустите сервер inetd. Таким образом вы закомментируете строку, предписывающую запуск службы ftp. Новички в мире Linux часто путают два файла /etc/inetd.conf и /etc/services. Желая предотвратить запуск некоторой службы, они часто комментируют строки в файле /etc/services, однако на самом деле поступать так не следует. Файл /etc/ services содержит описания всех доступных в системе служб, а файл /etc/inetd.conf содержит информацию о том, какие из этих служб разрешается запускать с использованием демона inetd. Если вы хотите предотвратить загрузку в память некоторой службы, будет лучше, если вы закомментируете соответствующую строку в файле /etc/inetd.conf, оставив при этом файл /etc/services без изменений. Конечно, комментирование или удаление записей в любом из этих файлов приведет к одним и тем же результатам — соответствующая служба перестанет запускаться, однако следует учитывать, что информация из файла /etc/services зачастую используется другими важными программами, поэтому если вы удалите из этого файла информацию о некоторой службе, вы можете нарушить работу таких программ.

СОВЕТ

Перезапустить серверную программу inetd можно несколькими разными способами. Во многих документах рекомендуется посылать серверу сигнал SIGHUP. Это можно сделать следующим образом:

```
kill  -l  $(pidof inetd) kill  -HUP $(pidof inetd)
```

Вы также можете просто перезапустить сервер:

```
/etc/rc.d/init.d/inet stop :  /etc/re.d/init.d/inet start
```

Следует отметить, что при остановке или перезапуске inetd любые серверные приложения, запущенные с использованием inetd до этого, не будут остановлены. Иными словами, если кто-либо подключен к системе с использованием telnet, этот сеанс не будет уничтожен, однако после остановки inetd новые удаленные пользователи не смогут подключиться к системе *с* использованием telnet.

Защита от сетевых атак

Первое, о чем следует сказать, приступая к разговору о сетевых атаках, это то, что для проникновения в вашу систему через сеть взломщик должен воспользоваться одной или несколькими службами, работающими на вашем компьютере и доступными через сетевой интерфейс, к которому имеет доступ взломщик.

СОВЕТ

Любые атаки (то есть взлом системы) осуществляются через службы, которые вы делаете доступными для внешнего мира. Если вы завершаете работу службы или запрещаете доступ к ней, вы предотвращаете доступ к системе извне с использованием этой службы.

Если на сервере не работает ни одна служба, сервер недоступен для атак через сеть. С другой стороны, если на сервере не работает ни одна служба, значит, сервер не в состоянии обслужить ни одного клиента. Что же это за сервер? Такой сервер полностью бесполезен для внешнего мира. Однако это уже другой вопрос. Остаток этой книги будет посвящен обслуживанию и поддержке системы, которая предлагает для внешнего мира набор сетевых служб и при этом остается в достаточной степени защищенной. Обеспечение должного обслуживания клиентов и необходимого уровня безопасности требует определенных усилий с вашей стороны. Если вы не можете или не хотите присматривать за состоянием вашей системы (либо вручную, либо при помощи автоматизированных сценариев, которые пересылают вам необходимые сведения по электронной почте), вы должны либо отказаться от запуска сетевых служб, либо блокировать доступ к ним со стороны всех соединений за исключением некоторых, которым вы доверяете. И даже в этом случае вы не сможете считать себя в достаточной степени защищенным.

Что является объектом атаки? Многие взломщики, действующие на просторах Интернета, являются дилетантами с низким уровнем подготовки и поверхностными знаниями. Как правило, это — достаточно молодые люди с амбициями, иногда настроенные против общества, приступающие к взлому для того, чтобы просто попробовать, получится ли у них это или нет. Однако вы должны учитывать, что в настоящее время это только часть всех злоумышленников Интернета. Во Всемирной сети действует также немалое количество значительно более подготовленных и обученных взломщиков, обладающих более продвинутыми знаниями и навыками. Цели, которые они ставят перед собой, более значительны. Подчас их не интересует конкретно ваша система, однако совершая свои неблаговидные дела, они стараются тщательнее замести следы. Для этой цели они стараются использовать несколько посторонних систем и создать впечатление, что атака на более крупную цель была произведена кем угодно, но не ими. Именно с этой точки зрения они могут заинтересоваться конкретно вашей системой.

Следующая фраза будет повторена несколько раз на протяжении всей оставшейся книги: время является как вашим другом, так и вашим врагом. Некоторые сведения и советы, приведенные в данной книге, позволят вам выиграть время, в течение которого вы сможете обнаружить попытки взлома или дыры в системе безопасности и должным образом прореагировать, чтобы устранить опасность. Однако как только взломщик успешно проникает в вашу систему, время становится вашим врагом. Обладая даже очень небольшим запасом времени, многие взломщики способны быстро замести следы взлома и заменить используемые вами исполняемые файлы на свои собственные версии, которые позволяют им в течение длительного времени работать с вашей системой незамеченными.

Вся сеть работает благодаря тому, что на определенном уровне существует доверие. Кредит доверия может быть предоставлен пользователю и/или системе. Система доверяет пользователю root. Вы доверяете своим исполняемым файлам. Именно поэтому взломщик почти всегда заменяет ключевой, наиболее важный исполняемый файл, которому вы доверяете. Именно поэтому взломщик почти всегда пытается получить доступ к вашей системе в качестве пользователя root. Если взломщик желает устранить абсолютно все следы своего проникновения и деятельности внутри вашей системы, он должен обладать полномочиями root. Если взломщик желает обладать доступом к системе в течение достаточно продолжительного времени, он должен установить в системе один или несколько модифицированных исполняемых файлов или отредактировать файлы /etc/passwd и /etc/shadow.

СОВЕТ

Вы должны сделать все возможное, чтобы защитить вашу систему. Затем вы должны должным образом настроить и проверять систему аудита. Чем сложнее в системе аудита найти следы пребывания в системе

взломщика, тем больше вероятность того, что вы все-таки сможете обнаружить злоумышленника. Кроме того, несколько разных следов в системе аудита лучше, чем только один такой след.

Так что же следует делать? Каким образом вы можете узнать об опасности? Как я сказал ранее, время — это ваш друг. Прежде чем произойдет попытка взлома вашей системы, в большинстве случаев (но все же не всегда) вы можете получить что-то вроде предупреждения. Чтобы узнать об уязвимых местах вашей системы, взломщик должен более тщательно изучить саму вашу систему. Это можно сделать несколькими разными способами однако наиболее простым является способ сканирования систем. При помощи сканирования взломщик может определить, какие службы, работающие на вашем компьютере, являются доступными для внешнего мира (при этом взломщик получает информацию, которая сильно напоминает сведения, отображаемые на экране при помощи команды netstat -a). В большинстве случаев он с точностью до нескольких разрядов версии узнает, какая операционная система установлена на вашем компьютере. Обладая этими сведениями, он может более подробно определить возможные наиболее слабые места защиты вашей ОС и используемых вами служб. При этом злоумышленник может попробовать угадать интересующую его службу. Например, зачастую взломщик, желающий знать больше о некоторой службе, может просто подключиться к ней и попросить сервер предоставить информацию об этой службе (подобная информация может предоставляться по запросу, а может выдаваться в самом начале соединения без какого-либо запроса).

На каждом этапе перечень потенциальных уязвимых мест в защите сужается. Это очень важный момент: чем больше атакующий может узнать о вашей системе, тем более уязвимой она является. Например, в среде Linux можно использовать несколько разных FTP-серверов: wu-ftpd, ProFTP и т. д. Каждый из этих серверов обладает своим набором уязвимых мест. Уязвимые места одного сервера отличаются от уязвимых мест другого сервера. В большинстве случаев попытка получить доступ к системе на уровне привилегий root (для этого часто используется английский термин «to *root a box»)* осуществляется с использованием ошибки переполнения буфера.

СОВЕТ

Проникнув в систему, злоумышленник часто приносит с собой так называемый «комплект root» (root kit), то есть набор средств, позволяющих ему получать привилегированный доступ к вашей системе в течение продолжительного времени и при этом оставаться незамеченным.

Переполнение буфера — это ситуация, когда в переменную записывается число большее, чем может хранить в себе эта переменная. Иными словами, в ячейку записывается количество информации большее, чем может хранить в себе эта ячейка. Например, представьте себе, что некоторая переменная может хранить в себе не более 8 байт. Теперь подумайте, что может произойти, если в нее попытаться записать 16 байт. Если программист забыл включить в программу соответствующую проверку, обработка записываемого в ячейку числа может быть выполнена некорректно. Например, в переменной могут оказаться только младшие 8 байт, остальные 8 байт могут попасть в другую область памяти, в результате чего обработка данных может быть выполнена не так, как это происходит в нормальных условиях.

Приведу небольшой пример. Некоторое время назад взломщики использовали ошибку, присутствующую в программе phf (которая в настоящее время не используется, однако до сих пор иногда встречается на некоторых старых системах) для получения привилегированного доступа к web-серверам. Ошибка была основана на переполнении буфера ввода этой программы. Чтобы взломать сервер, необходимо было послать службе www этого сервера определенную последовательность символов в составе URL-адреса:

http://www.yourserver.org/cgi-bi n/phf?Qname=x%Oa/bin/sh+-s%0a

Обратите внимание на последовательность /bin/sh. Атакующий пытается обмануть phf и запустить командную оболочку с выходом в коммуникационный порт. Сценарий phf может работать от имени пользователя root, поэтому в результате удачного выполнения подобного трюка взломщик получает доступ к системе на уровне привилегий суперпользователя.

Но всегда ли злоумышленник осуществляет сканирование, прежде чем атаковать систему? Нет. Если ваша система делает доступными для внешнего мира какие-либо общеизвестные сетевые службы, например www, будьте уверены, что любой атакующий будет знать об этом без всякого сканирования. Если злоумышленник хочет атаковать web-сервер, ему не надо сканировать систему для того, чтобы узнать о том, что на ней запущена служба www. Он может просто попробовать осуществить атаку, описанную в предыдущем параграфе, в надежде на то, что на атакуемом сервере описанная дыра в системе безопасности не закрыта. Я более чем уверен в том, что если этот злоумышленник не будет пойман и наказан его интернет-провайдером (что очень маловероятно) или одной из атакованных им систем, рано или поздно он получит доступ к какой-либо системе. В действительности данная атака выглядит так, как будто нападение

осуществляется с другого подверженного нападению компьютера (складывается впечатление, что нападение осуществляется с двух разных учебных заведений в одно и то же время и почти наверняка в каждом из этих учреждений для атаки используется взломанная система).

Какими еще средствами может воспользоваться взломщик, чтобы получить доступ к вашей системе? Таких методов множество, однако наиболее популярным является метод, когда взломщик получает доступ к системе, а затем просто «прослушивает» весь трафик, который передается через локальный сетевой кабель, выделяя из него все попадающиеся ему пары «имя пользователя/пароль», передаваемые через сеть. Например, любой дилетант, работающий в сети Road Runner (служба кабельных модемов, доступная в некоторых регионах США), может свободно прослушивать любой трафик, передаваемый через локальный сегмент сети. Если в локальном сегменте насчитывается 200-500 пользователей, за несколько дней злоумышленник может раздобыть более 1000 пар «имя_пользователя/па-роль». Многие пользователи Road Runner используют каналы этой сети для подключения к корпоративной сети своей компании. При этом используются telnet, ftp или программы получения электронной почты. Во многих случаях пароли передаются через сеть в незашифрованном виде. В результате взломщик получает доступ к сотням систем и становится обладателем множества корректных паролей, благодаря чему он получает самые широкие возможности. Все, что ему для этого потребовалось, — это проникновение в систему и взгляд вокруг. Обладая зарегистрированным в системе именем и подходящим корректным паролем, злоумышленник может закачать в систему необходимые файлы и установить комплект доступа на уровне root (root kit). После этого взломщик завладевает системой и может похитить еще больше пользовательских паролей.

Многие операторы связи уже знакомы с таким трюком и используют более защищенные схемы доступа пользователей кабельных модемов к Интернету. Благодаря этому «стянуть» из сети чужие пароли становится не так просто. Как правило, для обеспечения должного уровня защиты каждое соединение превращается в туннель от пользователя к серверу, благодаря этому системы, подключенные к одному и тому же сетевому кабелю, не могут читать чужой трафик, передаваемый по этому кабелю. Однако на данный момент времени такой подход используется далеко не всеми.

Когда взломщик проникает в систему, он может воспользоваться приемами, основанными на переполнении буфера. Находясь в системе, взломщик может попытаться реализовать подобный прием с любыми бинарными файлами, которые выполняются с использованием SUID root. Однако существуют и другие уязвимые места, например, те, которые связаны со скоростными условиями (race condition). Скоростные условия — это набор обстоятельств, которые пользователь может использовать для того, чтобы непривилегированное обращение к системе было выполнено на уровне привилегий root.

В системе Linux некоторые процедуры, такие как создание каталога, связаны с обращением к ядру ОС при помощи системного вызова. Процесс создания каталога состоит из двух этапов: на первом этапе благодаря выполнению высокопривилегированного кода создается каталог, на втором этапе владельцем этого каталога назначается пользователь, инициировавший процедуру создания каталога. Для выполнения каждого из этих этапов требуется некоторое время. Для смены владельца каталога требуется обращение к ядру, поэтому соответствующий процесс работает на уровне привилегий root.

К другим уязвимым местам относится возможность «обмана» ядра таким образом, чтобы оно по запросу обычного пользователя выполнило некоторый процесс на уровне привилегий root. Также злоумышленник может воспользоваться одним из «защищенных» каталогов, таких как /tmp. В этом каталоге процесс, работающий на уровне root, может разместить файл с известным именем. Взломщик, который заранее знает это имя, может заблаговременно разместить там такой файл, и тогда процесс root будет использовать созданный злоумышленником файл вместо своего собственного.

Прочитав несколько предыдущих абзацев, легко можно стать параноиком. Количество разного рода злоумышленников и взломщиков в Интернете со временем все увеличивается (благодаря недорогому круглосуточному выделенному доступу), однако преимущества использования Интернета в большинстве случаев существенно перевешивают риск. Чтобы воспользоваться этими преимуществами, вы должны сделать вашу систему защищенной как от дилетантов, так и от опытных взломщиков, действующих на просторах Интернета в наши дни.

Что такое горшок с медом?

Горшок с медом (honey pot) — это система, предназначенная для приманивания «мух», то есть взломщиков, желающих проникнуть в вашу сеть. Данная система является «ловушкой для дураков» — любые попытки взлома документируются в журналах. Такая система не выполняет каких-либо важных функций, по крайней мере, она не содержит каких-либо важных данных. Единственное предназначение системы — отслеживать попытки доступа и записывать соответствующую информацию в журнал для дальнейшего анализа. Горшок с медом оснащается специальным программным обеспечением, например, на такой системе можно запустить утилиту tcpdump, которая будет записывать весь сетевой трафик,

передаваемый через интерфейс, подключенный к Интернету.

Некоторые компании, занимающиеся разработками технологий в области безопасности, используют горшки с медом, чтобы лучше изучить анатомию взлома систем. Они занимаются этим, отлично зная, на что идут и чем они при этом рискуют. Использование горшка с медом, в особенности если взломщик знает, что это такое, может стать для вас причиной многих горестей и неприятностей. В наше время многие юные взломщики обладают ярко выраженной антиавторитарной направленностью, и если они обнаруживают расставленную вами ловушку, они будут пытаться всеми силами отомстить вам. Можно ожидать, что жестокие нападения будут совершаться на любой из компьютеров, идентифицируемый как часть вашей сети. Если же все эти попытки будут безуспешны, раздраженные до крайности злоумышленники наверняка организуют атаку типа Denial of Service (отказ обслуживания). Таким образом, если вы не являетесь профессиональным сетевым администратором с глубокими знаниями в области безопасности, использование горшков с медом — это не самая лучшая идея.

Заключение

В данной главе рассказывается о том, каким образом Linux обеспечивает работу большого количества служб и при этом не расходуется значительный объем системных ресурсов. Используя метадемон Интернета (inetd), вы можете организовать эффективное функционирование в системе нескольких небольших и быстрых служб. Вы также узнали, каким образом при помощи файла inetd.conf настраивается поведение всех этих служб и метадемона inetd.

Также я рассказал вам о последствиях предоставления служб, работающих в вашей системе, всему остальному миру и о том каким образом взломщики могут попытаться использовать эти службы для проникновения в систему. Вы узнали, что в большинстве случаев перед попыткой взлома вы можете наблюдать некоторого рода предупреждение, которое указывает на то, что кто-то пытается проникнуть в вашу систему. Вы получили также базовые представления о том, как это можно сделать и что должен знать взломщик для того, чтобы проникнуть в систему.

12 Уязвимые службы и протоколы

В данной главе рассматриваются следующие вопросы:
- что такое сетевые службы;
- уязвимые места сетевых служб;
- что такое серверные приложения.

После завершения установки любого комплекта поставки Linux сразу же после самой первой загрузки системы по умолчанию в ней будет функционировать набор служб (см. раздел, посвященный утилите netstat в главе 10). Состав запущенных в системе служб определяется многими факторами, например, набором пакетов, выбранных вами для установки. В большинстве случаев, если вы указываете программе установки установить в системе некоторую службу, скорее всего, после загрузки системы по умолчанию эта служба будет функционировать с использованием конфигурации по умолчанию. Таково стандартное поведение большинства дистрибутивов Linux. Комплект поставки Debian позволяет настраивать некоторые из устанавливаемых служб в процессе установки, однако далеко не все. Таким образом, если вы устанавливаете в системе некоторую службу, будьте готовы к тому, что по умолчанию она будет запущена. Вы должны позаботиться о том, чтобы эта служба была должным образом настроена. Если вы не намерены предоставлять какую-либо службу в распоряжение пользователей, удалите соответствующий пакет из системы. Если в системе функционирует большое количество служб, вы должны учитывать, что каждая их них является потенциальной угрозой безопасности. Любая сетевая служба, включая службы, которые, на первый взгляд, не являются сетевыми, имеет модуль ожидания поступления сообщений через TCP-порт, UDP-порт или RPC-порт. В данной главе я расскажу о некоторых традиционных службах и связанных с ними потенциальных проблемах.

СОВЕТ

Нарушить систему защиты через сеть можно только через доступные службы. Службы, которые не функционируют в системе, не могут быть использованы для взлома. Если вы видите, что в вашей системе работает некоторая служба и вы не можете сказать, зачем она нужна, отключите ее или сделайте доступной только для локального сетевого узла localhost (более подробно об этом рассказывается в главе 16). Этим вы существенно снизите риск.

Службы, связанные с портами, номера которых меньше 1024, считаются привилегированными. Это означает, что связать службу с портом, номер которого меньше 1024, можно только на уровне привилегий root. Иными словами, любая служба, использующая любой порт с номером меньшим 1024, работает от имени пользователя root. Если вы не предпримете должных мер безопасности, защита вашей системы может оказаться нарушенной. Порты с номерами больше 1024 могут использоваться на любом уровне привилегий, однако их можно использовать и на уровне привилегий root. То есть даже если служба связана с портом, номер которого больше 1024, это не означает, что она абсолютно безопасна. В данной главе все же мы будем рассматривать только те службы, которые связаны с привилегированными портами.

FTP, порты 21 и 20

Протокол FTP (File Transfer Protocol) используется для обмена файлами между двумя компьютерами. Сервер FTP не обязательно постоянно находится в активном состоянии. Ожидание поступления запроса через TCP-порт с номером 21 осуществляется демоном inetd, и серверное приложение FTP запускается тогда, когда в этом возникает необходимость (то есть когда к системе пытается подключиться клиент FTP). Если вы используете tcpd, выполняются действия, связанные с tcpd, и если клиент не блокируется, он будет подключен к серверу FTP и ему будет передано приглашение на вход в систему.

Часто когда говорят об FTP, имеют в виду клиентское или серверное приложение, обеспечивающее выполнение связанных с FTP функций на одной или на другой стороне. Но на самом деле FTP — это

протокол. Следует отметить, что любое серверное приложение, будь то ftp, telnet и т. п., использует свой собственный протокол обмена данными.

ПРИМЕЧАНИЕ

Протокол — это набор правил, определяющих, каким именно способом осуществляется обмен данными между двумя системами (протокол означает правила). Таким образом, каждая отдельная служба обладает своим собственным коммуникационным протоколом.

По сравнению с большинством других серверов сервер FTP функционирует необычно. Вдобавок к стандартному соединению через порт 21 этот сервер открывает дополнительное соединение, но только тогда, когда в этом возникает необходимость. Вся процедура выполняется следующим образом.

- Клиент пересылает запрос на подключение, направленный на порт 21 сервера.
- Выполняется последовательность действий SYN-ACK, и (если вы используете tcpd) демон tcpd осуществляет возложенные на него функции.
- Если запрос на подключение не отвергается демоном tcpd, для ответа на этот запрос, поступивший через порт 21, запускается сервер FTP.
- Клиент и сервер обмениваются информацией и начинают сеанс связи.

Активный сеанс FTP (называемый «активным» только для того, чтобы отличить его от «пассивного» сеанса) предусматривает использование порта 21 (с последующим переключением на случайно выбранный порт с номером большим 1024 для продолжения соединения) для пересылки между клиентом и сервером данных, связанных с подключением, а также команд. Но как только клиент отправляет серверу запрос на получение информации (перечень файлов в каталоге или содержимое некоторого файла), открывается еще один канал. Для этой цели используется порт данных FTP (ftp-data) — порт с номером 20. Сервер подключается к клиенту через порт 20 (соединение продолжается через случайно выбранный порт с номером большим 1024) и приступает к пересылке клиенту запрошенных им данных. Состояние передачи пересылается через порт 21.

Связывание с портом

В этой и во многих других книгах, посвященных функционированию сетей, часто употребляется выражение: «служба связана с портом» (service binds a port). Это означает, что программа, осуществляющая обслуживание запросов, создает соединение с этим портом. Для этой цели создается сокет, и программа ожидает поступления через этот сокет данных из сети. Для каждого порта в системе может быть создан только один сокет. Таким образом, две разных программы (или два экземпляра одной и той же программы) не могут быть связаны с одним и тем же портом. Однако одна программа может обслуживать одновременно несколько подключений к одному и тому же сокету. Для этого создаются дочерние процессы, каждому из которых передается очередное соединение. Таким образом, с портом связан только один процесс, который является родительским, а дочерние процессы занимаются обслуживанием запросов. Некоторые из служб связываются одновременно с несколькими портами (например Apache).

Если вы пытаетесь запустить еще один экземпляр уже работающей программы или другую программу, которая связывается с некоторым портом, с которым уже работает одно из функционирующих в системе приложений, система выведет сообщение об ошибке, извещающее вас о том, что порт уже связан. Один и тот же порт можно связать для передачи данных по протоколу TCP, равно как и по протоколу UDP в одно и то же время. Как правило, связывание портов выполняется для того, чтобы обеспечить доступ клиентов к работающим на сервере службам, однако некоторые порты связываются злоумышленниками как метки, указывающие на то, что система взломана. Полный перечень портов, используемых программами типа «троянский конь», а также портов-меток содержится на web-странице http:// www.simovits.com/nyheter9902.html.

Прочитав описание работы FTP, многие читатели, должно быть, скажут: «Ах вот почему при использовании клиента ftp напрямую у меня возникают проблемы и для скачивания файлов по протоколу FTP я вынужден использовать Netscape». Очень многие брандмауэры, и в особенности программное обеспечение, выполняющее маскировку адресного пространства IP (например, система Network Address Translation, NAT), разрешают устанавливать исходящие соединения через порты с номерами меньшими 1024, однако при этом запрещают устанавливать входящие соединения через эти порты. Что же происходит? Находясь под защитой брандмауэра (или в сети, использующей NAT), вы отправляете внешнему серверу FTP запрос на получение некоторого файла. Запрос является исходящим и направлен в порт 21 сервера, поэтому он беспрепятственно проходит через брандмауэр. После этого ваша система (клиент ftp) начинает ждать входящего соединения через порт 20. Сервер FTP пытается вступить с вами в контакт через порт 20, однако брандмауэр блокирует это соединение (оно является входящим и направлено

в один из привилегированных портов). Если же вы работаете в сети NAT, то внешний сервер FTP просто не знает вашего внутреннего IP-адреса и поэтому отправляет запрос на подключение через порт 20 вашему брандмауэру NAT. А брандмауэр NAT просто не знает, кому на самом деле адресован этот запрос. Известно только, что сам брандмауэр не запрашивал соединения, и поэтому поступающие пакеты отбрасываются.

ССЫЛКА

Более подробное обсуждение брандмауэров и маскировки IP содержится в главах 16 и 18 соответственно.

Для передачи файлов по протоколу FTP браузер Netscape использует режим передачи, называемый *пассивным.* При этом, вместо того чтобы открыть дополнительный канал связи с клиентом, сервер пассивно ожидает, пока клиент (в нашем случае Netscape) начнет получать данные через порт 21. Этого же самого можно достичь, воспользовавшись входящей в комплект OpenLinux программой под названием pftp (passive ftp — пассивный ftp). Некоторые клиенты FTP поддерживают пассивный режим работы, который инициируется специальной командой.

Чтобы вы могли лучше представить себе этот процесс, воспользуемся клиентом telnet для подключения к серверу ftp и выполнения нескольких команд. По умолчанию клиент telnet использует порт 23, однако его можно настроить на подключение к порту 21 сервера. Взгляните на листинг 12.1, вы можете попробовать сделать то же самое самостоятельно.

Листинг 12.1. Соединение с сервером FTP при помощи клиента telnet

```
[david@volcan david]$ telnet chiriqui 21
Trying  192.168.0.2...
Connected to chiriqui.pananix.com.
Escape character is '^]'.
220 chiriqui.pananix.com FTP server (Version wu-2.5.0(l) Tue Jul  27 18:42:33 MDT 1999)

ready. user david
331 Password required for david.
pass mypasswd
230 User david logged in.
help
214-The following commands are recognized (* =>'s unimplemented).
USER  PORT  STOR  MSAM*  RNTO  NLST  MKD   COUP
PASS  PASV  APPE  MRSQ*  ABOR  SITE  XMKD  XCUP
ACCT*  TYPE  MLFL*  MRCP*  DELE  SYST  RMD   STOU
SMNT*  STRU  MAIL*  ALLO  CWD  STAT  XRMD  SIZE
REIN*  MODE  MSND*  REST  XCWD  HELP  PWD  MDTM
QUIT  RETR  MSOM*  RNFR  LIST  NOOP  XPWD 214 Direct comments to root@localhost.
pasv
227 Entering Passive Mode (192,168,0,2.214,43) stat 211-chiriqui.pananix.com FTP
server status:
Version wu-2.5.0(l) Tue Jul 27 18:42:33 MDT 1999
Connected to volcan.pananix.com (192.168.0.1)
Logged in as david
TYPE: ASCII, FORM: Nonprint: STRUcture: File; transfer MODE: Stream
in Passive mode (192,168,0,2,214,43) 0 data bytes received in 0 files 0 data bytes
transmitted in 0 files 0 data bytes total in 0 files 45 traffic bytes received in 0 transfers 1113
traffic bytes transmitted in 0 transfers 1208 traffic bytes total in 0 transfers 211 End of status

221-You have transferred 0 bytes in 0 files.
221-Total traffic for this session was 1319 bytes in 0 transfers.
221-Thank you for using the FTP service on chiriqui.pananix.com.
221 Goodbye.
Connection closed by foreign host.
```

Проанализировав данное соединение, вы можете обнаружить, что в его рамках происходит множество интересных вещей. В первой строке инициируется сеанс telnet, однако вместо порта по умолчанию 23 используется порт 21. В самом начале соединения сервер сообщает клиенту некоторую информацию о себе самом. Благодаря этому вы можете получить информацию о типе сервера и номере версии, которая используется на удаленной системе. В нашем случае на удаленном узле с именем chiriqui используется разработанный в университете Вашингтона (Washington University) сервер wu-ftp версии 2.5.0(1), сборка которого была выполнена вечером 27 июля 1999 года в шесть часов сорок две минуты. Эта информация является чрезвычайно важной. Некоторые версии сервера wu-ftp содержат в себе ошибки, о которых давно известно множеству людей. В более поздних версиях этого сервера все эти ошибки устранены, однако если вы не выполнили своевременного обновления вашего сервера, злоумышленник может воспользоваться одной из этих ошибок и тем или иным образом взломать систему. Если вы поищете

соответствующую информацию на разнообразных хакерских сайтах, вы сможете быстро узнать подробнее о данной версии сервера и ее уязвимых местах. Зачастую на подобных сайтах можно обнаружить даже программы, которые позволяют взломать систему, используя для этого описанные уязвимые места.

В следующих нескольких строках упоминается пользователь david. Как правило, клиент ftp просит вас ввести имя пользователя во время процесса подключения. Так как для подключения к FTP-серверу вы используете клиента telnet, а не ftp, вы должны самостоятельно передать серверу регистрационное имя. Далее следует напоминание о том, что вы должны ввести пароль. На следующей строке «pass mypasswd» этот пароль вводится и передается серверу. Обратите внимание, что пароль отображается на экране в том виде, в каком он показан в листинге, то есть в виде обычного текста. В отличие от telnet, клиент ftp блокирует отображение пароля. Для этой цели код клиента выполняет системный вызов, отключающий отображение символов на экране (как будто выполняется команда stty -echo). После того как пользователь введет пароль и нажмет на Enter, пароль пересыла-ется серверу, и выполняется системный вызов, аналогичный команде stty echo, который вновь включает отображение информации на экране.

Текст, приведенный в листинге, в точности соответствует той информации, которая передается через сетевой кабель в процессе обмена данными. Команды, которые вы набираете на клавиатуре, и текст, возвращаемый сервером, передаются через сеть именно так, как это показано в листинге. Иными словами, если во время сеанса FTP вы запустите утилиту tcpdump, вы сможете обнаружить среди сетевого трафика в точности этот же текст. Легко себе представить, насколько уязвима ваша система перед злоумышленником, который запустит утилиту tcpdump внутри вашей сети. Благодаря этому он получит возможность прочитать пароль доступа к FTP-серверу. На самом деле, если маршрутизатор, при помощи которого ваша сеть соединяется с другими сетями, пропускает внутрь вашей сети пакеты из других сетей (такие пакеты, которые на самом деле не предназначены для компьютеров вашей сети), используя tcpdump, вы сможете наблюдать, как через локальный для вас сегмент сети передается достаточно большое количество пар «имя_пользователя/пароль».

Вернемся к изучению листинга. По команде help сервер сообщает клиенту перечень поддерживаемых им команд. Для использования некоторых из них требуются дополнительные аргументы. Возможно, для выполнения той или иной команды нужно некоторое взаимодействие с клиентом, которое вы не в состоянии обеспечить. В конце листинга клиент получает статистическую информацию, а затем завершает сеанс FTP.

ПРИМЕЧАНИЕ

Если вы используете клиент telnet для подключения к серверу, не являющемуся сервером telnet, и при этом обнаруживаете, что вы не знаете что делать дальше — приглашение на ввод команды не появляется, — нажмите комбинацию клавиш Ctrl+] (как рекомендует подсказка «Escape charaster is '^]'»). При этом вы получите доступ к приглашению telnet, в котором можете набрать quit для того, чтобы завершить рабочий сеанс.

По умолчанию анонимный доступ к FTP-серверу обеспечивается в случае, если для подключения используется имя пользователя anonymous или ftp. Пользователь anonymous — это синоним ftp. Если клиент получил возможность анонимного доступа, система делает для него корнем файловой системы домашний каталог ftp. Таким образом, пользователь как бы попадает в тюрьму. Иными словами, он не может перемещаться в какие-либо другие каталоги, отличающиеся от домашнего каталога ftp. Вся система для него выглядит как домашний каталог ftp. Пользователь не может переместиться выше по иерархии каталогов файловой системы. Символические ссылки, указывающие в другие части системы, не работают. Однако для того, чтобы обеспечить корректную работу, данная «тюрьма» должна обладать всем тем, чем обладает корень системы (более подробно об этом рассказывается в главе 14). Такая тюрьма обозначается английским термином *change root jail* (тюрьма с измененным корнем). Некоторые службы используют подобную тюрьму для того, чтобы ограничить возможности подключающихся к ним клиентов и уменьшить повреждения, которые могут быть нанесены взломщиком.

Если в одной системе вы запустите несколько (некоторую комбинацию) доступных для внешних клиентов служб, в системе безопасности могут возникнуть пробелы, которые не свойственны каждой из запускаемых вами служб по отдельности. В частности, если к одной и той же системе вы организуете доступ по протоколам FTP и HTTP, у вас может возникнуть проблема. Один из известных способов взлома предусматривает закачивание в систему некоторого файла, к которому в дальнейшем происходит обращение через HTTP-сервер (предполагается, что он неправильно настроен), такой как Apache. Организация как анонимного, так и пользовательского FTP-доступа также сомнительна с точки зрения безопасноети, если только система не является выделенным FTP-сервером и не используется для работы каких-либо других служб. В главе 3 обсуждается приемлемая конфигурация разрешений на доступ к каталогу входящих файлов FTP. Чтобы решить проблему, лучше всего использовать две различные системы — одна будет предназначена для FTP-доступа, а вторая — для HTTP-доступа. Вы можете также настроить HTTP так, чтобы разрешить клиентам только получать файлы, однако при этом ни один из клиентов не сможет использовать HTTP для передачи данных на сервер.

Чтобы снизить риск взлома, необходимо проверить параметры, заданные с использованием файла/ect/pam.d/ftp. Содержимое файла/ect/pam.d/ftp, используемое в OpenLinux по умолчанию, показано в листинге 12.2.

Листинг 12.2. Содержимое файла /ect/pam.d/ftp по умолчанию

```
auth       required       /lib/security/pam_listfile.so item=user sense=deny file=/etc/
ftpusers onerr=succeed
auth       required       /lib/security/pam_pwdb.so shadow null ok
auth       required       /Iib/security/pam_she11s.so
account          required          /lib/security/pam_pwdb.so  session          required
/lib/security/pam_pwdb.so
```

Первая строка предписывает произвести поиск в файле /etc/ftpusers (на это указывает параметр file=/etc/ftpusers). Файл содержит имена пользователей (item=user), и этим пользователям запрещен доступ к системе (sense=deny). Если файл не существует или пользовательское имя в нем не указано, пользователь сможет получить доступ.

Вторая строка указывает на то, что требуется соответствие пары «имя_поль-зователя/пароль» (допускается использование пустых паролей — null). Третья строка проверяет, является ли командная оболочка пользователя одной из командных оболочек, перечисленных в файле /etc/shells. Последние две строки составлены так, как объяснено в главе 1.

telnet, порт 23

Еще одним широко распространенным протоколом является протокол telnet. В предыдущем разделе я продемонстрировал, каким образом клиент telnet можно использовать для подключения к любому порту. Клиент telnet является чрезвычайно мощным инструментом отладки самых разнообразных служб, в особенности тех, протокол работы которых известен.

Однако насчет telnet не стоит заблуждаться — сервер telnet является наиболее опасным из всех запущенных в системе серверов. Протокол telnet — это широко открытый протокол в том смысле, что любой желающий может использовать его для доступа к системе. Поток данных, передаваемых telnet, прост и незашифрован. Сервер telnet опасен даже в случае, если вы используете его в частной сети. Гораздо более безопасным (и более гибким) является протокол удаленного входа в систему (remote login), на котором основана работа службы ssh (Secure Shell).

Если в вашей системе работает сервер telnet, я настойчиво рекомендую вам остановить его. Если вы желаете понять, кто пытается обнаружить в вашей систем открытые порты telnet (как правило, этим занимаются малоопытные взломщики и дилетанты), вы можете устроить ловушку. Откройте порт 23 с использованием оболочки TCP (о программах категории TCP Wrapper рассказывается в главе 15) и при этом запретите доступ к службе или используйте сетевой фильтр, который позволит вам следить за содержимым пакетов, но при этом отбрасывать их (см. главу 16). Выбор за вами. Как и во многих других системах, в Unix существует несколько способов решить какую-либо проблему. Вы должны подобрать такой, который лучше всего соответствует вашей политике.

СОВЕТ

Утилита dig не описывается в данной книге, однако эта утилита оказывается весьма полезной в случае, если вы хотите следить за IP-адресами, являющимися источниками сообщений. Отдайте команду dig -x <неизвестный_IP-арес> и вы можете узнать владельца этого IP-адреса. Обладая этими сведениями, вы можете составить жалобу, адресованную интернет-провайдеру или владельцу сети. В состав жалобы можно включить содержимое журналов. Возможно, это подействует. Если нет, то вы можете блокировать данный IP-адрес или соответствующую ему подсеть.

smtp, порт 25

В рамках Caldera OpenLinux по умолчанию устанавливается серверная программа sendmail — несомненный лидер в области серверных приложений электронной почты. Эта программа — чрезвычайно мощная и гибкая. Она может обслуживать одновременно сотни подключений, через которые осуществляются как прием, так и передача почты. Обсуждению корректной конфигурации этого приложения можно посвятить отдельную книгу. Конфигурация sendmail, используемая в Caldera OpenLinux по умолчанию, вполне подходит для большинства ситуаций. Для нормальной работы вам наверняка потребуется почтовый сервер, однако существует мнение, что другие серверы более безопасны и проще настраиваются. К таким серверам относится, например, qmail. Такие серверы не входят в состав Caldera OpenLinux, поэтому если вы заинтересованы в их использовании, вам потребуется выполнить

поиск в Интернете.

ПРИМЕЧАНИЕ

Все почтовые программы делятся на три категории. К первой из них относятся серверы SMTP, которые используются для передачи почты между системами. Такие программы называют агентами передачи почты (Mail Transfer Agent, MTA). Ко второй категории относятся почтовые клиенты, которые осуществляют чтение и передачу почты MTA. Такие программы называются почтовыми пользовательскими агентами (Mail User Agent, MUA). Часто для обмена почты между MTA и MUA используются агенты локальной доставки почты (Mail Delivery Agent, MDA). К этой категории относятся такие программы, как procmail и т. п. Приложения MDA получают почту от sendmail и направляют ее в локальный почтовый ящик.

Сервер sendmail — это большая, мощная и популярная программа, поэтому за все время ее существования взломщики обнаружили множество способов ее взлома. К счастью, sendmail позволяет использовать некоторые приемы, повышающие защиту. Например, вы можете создать отдельный процесс для некоторых пользователей, которые, как правило, либо отправляют, либо принимают почту, чтобы ограничить объем повреждений в случае, если сообщение содержит исполняемый код (иначе говоря, «троянского коня» — подробнее о «троянских конях» рассказывается в главе 14).

Многие участники электронных дискуссий в Usenet рекомендуют заменить sendmail другой аналогичной программой. В данной книге я не рекомендую вам делать этого. Несмотря на мощь и сложность, программу sendmail вполне можно использовать с соблюдением должного уровня безопасности. В листинге 12.3 показан протокол почтового соединения.

Листинг 12.3. Почтовое соединение

```
[david@volcan david]$ mail   -v rootPchiriqui
Subject:  test
this is a test.
.
EOT
root@chiriqui...  Connecting to chiriqui.pananix.com.  via esmtp...
220 chiriqui.pananix.com     ESMTP     Sendmail     8.9.3/8.9.3;     Fri,     8     Oct     1999     09:52:50     -0500
>>> EHLO volcan.pananix.com
250-chiriqui.pananix.com Hello volcan.pananix.com [192.168.0.1],  pleased to meet you
250-EXPN
250-VERB
250-8BITMIME
250-SIZE
250-DSN
250-ONEX
250-ETRN
250-XUSR
250 HELP
>>> MAIL From:<david@volcan.pananix.com> SIZE=49
250 <david@volcan.pananix.com>... Sender ok
>>> RCPT To:<root@chiriqui.pananix.com>
250 <root@chiriqui.pananix.com>... Recipient ok
>>> DATA
354 Enter mail, end with "." on a line by itself
>>> .
250 JAA06968 Message accepted for delivery
root@chiriqui... Sent (JAA06968 Message accepted for delivery)
Closing connection to chiriqui.pananix.com.
>>> QUIT
221 chiriqui.pananix.com closing connection
```

Чтобы увидеть, что именно происходит в процессе обычной процедуры передачи почты с использованием sendmail, вы можете запустить программу mail в режиме вывода подробной информации (mail -v). Эту программу можно использовать для решения проблем удаленной передачи почты. Обратите внимание, что, воспользовавшись таким приемом, вы можете получить отказ сервера в случае, если вы пытаетесь соединиться с sendmail, используя компьютер, который не является разрешенным для связи узлом sendmail. Подобным способом вы можете проверить, осуществляет ли сервер sendmail обмен почтой со всеми узлами или он настроен на соединение только с определенным набором разрешенных сетевых узлов.

Теперь попробуйте сделать то же самое при помощи telnet. Данное соединение осуществляется с сетевого узла volcan (192.168.0.1). Клиент подключается к узлу chiriqui (192.168.0.2). Сервер sendmail требует, чтобы было возможным определить имя подключающегося узла. Однако в остальном сервер sendmail чрезвычайно доверчив. И не только потому, что узел volcan принадлежит той же самой подсети

(вы можете попробовать выполнить то же самое в отношении любого узла в Интернете). Узлы, не требующие обратного сопоставления имени, являются уязвимыми. (Данный сервер sendmail работает в режиме отключенной ретрансляции — anti-relaying — однако в данном случае сервер chiriqui настроен так, что для узла volcan разрешена ретрансляция.)

Листинг 12.4. Подключение к sendmail с использованием telnet

```
[david@volcan david]$ telnet chiriqui 25
Trying  192.168.0.2...
Connected to chiriqui.pananix.com.
Escape character is '^]'.
220 chiriqui.pananix.com ESMTP Sendmail 8.9.3/8.9.3; Fri, 8 Oct 1999 09:59:39 -0500
EHLO local host
250-chiriqui.pananix.com Hello volcan.pananix.com [192.168.0.1], pleased to meet you
250-EXPN
250-VERB
250-8BITMIME
250-SIZE
250-DSN
250-ONEX
250-ETRN
250-XUSR
250 HELP
MAIL From:<root@localhost>
250 <root@localhost>... Sender ok
RCPT To:<root@localhost>
250 <root@localhost>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
I see you.

You can't hide.


250 KAA07061 Message accepted for delivery QUIT
221 chiriqui.pananix.com closing connection
Connection closed by foreign host.
```

СОВЕТ

Для того чтобы узнать перечень команд, поддерживаемых MTA, вы всегда можете использовать команду help. Однако имейте в виду, что MTA может отказать в выполнении некоторых команд, таких как vrfy или expn (эти две команды используются для проверки, а также расширения почтового адреса).

Первым отличием, которое бросается в глаза при сравнении первого и второго листингов, — это отсутствие приглашения >>, отмечающего, какой из сетевых узлов выполняет вывод текста. Сообщение подписано почтовым адресом root@localhost, однако на самом деле мы знаем, что оно отослано пользователем david@volcan. В действительности вы вообще можете не вводить обратный адрес. Можете попробовать поступить так, как показано в листинге 12.5.

Листинг 12.5. Подключение с сетевого узла null

```
MAIL From:<>
250 <>... Sender ok
```

Вторая строка указывает на то, что для sendmail данный ввод в порядке вещей. Однако взглянув на почтовое сообщение, включая все его заголовки, можно увидеть нечто подобное тексту листинга 12.6.

Листинг 12.6. Заголовок письма, сформированного в листинге 12.5

```
Received:  from localhost  (volcan.pananix.com [192.168.0.1])  by chiriqui.pananix.com
(8.9.3/8.9.3) with ESMTP id KM07263 for root<root@localhost>: Fri, 8 Oct 1999 10:15:07 -0500
Date: Fri, 8 Oct 1999 10:15:07 -0500 Message-ID:
<199910081515.KAA07263@chiriqui.pananix.com>
Status:
X-Mozilla-Status: 0000 X-Mozilla-Status2: 00000000
X-UIDL: 37fd305900000706

I see you. You can't hide.
```

Письмо в листинге 12.6 просматривается с использованием Netscape, поэтому в заголовке присутствуют строки, характерные для почтовой системы Netscape. Поле From: заголовка пусто, поэтому оно не отображается. В рамках стандарта заголовок должен включать в себя поля Subject:, To: и From:.

Первая строка заголовка начинается с метки Received: и занимает в листинге несколько строк. В ней сообщается, что сетевой узел volcan.pananix.com с IP-адресом 192.168.0.1, называющий себя localhost, переслал сообщение почтовому серверу *c* именем chiriqui, на котором работает MTA sendmail версии 8.9.3, использующий расширенную разновидность SMTP. Сообщение обладает идентификатором id KAA07263, адресовано root и датировано 8 октября 1999 года 10 часов 15 минут утра локального времени, которое на пять часов опережает время по Гринвичу (GMT). Идентификатор сообщения будет добавлен в журнал, в котором можно обнаружить сведения, приведенные в листинге 12.7.

Листинг 12.7. Запись в журнале о транзакции из листинга 12.4

```
Oct  8 10:15:45 chiriqui sendmail[7263]: KAA07263: from=o, size=29, class-0, pri=30029,
nrcpts=l, msgid=<199910081515.KAA07263@chiriqui.pananix.com>, proto=ESMTP,
relay=volcan.pananix.com [192.168.0.1]
Oct  8 10:15:46 chiriqui sendmail[7264]: KAA07263: to=david, delay=00:00:39,
xdelay=00:00:00, mailer=local, relay=loca1, stat=Sent
```

Каждая отдельная запись журнала начинается с даты. Таким образом, в журнал заносятся две записи. Первая запись указывает на прием сообщения для дальнейшей доставки. Вторая запись показывает, кому доставлено сообщение. Здесь вы можете видеть отметку to=david. Данная отметка добавляется в журнал потому, что вся почта, адресованная учетной записи root, передается определенному пользователю для прочтения.

Прием с подменой отправителя на root используется дилетантами для того, чтобы напугать малоопытных сетевых администраторов. Однако теперь вы знаете, каким образом это происходит (данный трюк часто применяется при рассылке «грязной» почты, то есть *spam),* вы можете убедиться в том, что это всего лишь пустой трюк и пользователь, отправивший почту, не обладает никакими специальными привилегиями. Вы также видите полноценный след, оставленный в результате выполнения этой транзакции в журналах аудита.

В подавляющем числе случаев атаки, нацеленные на sendmail, основаны на неправильно назначенных разрешениях на доступ. В процессе функционирования сервер sendmail обращается к множеству каталогов и файлов. Если разрешения на доступ к этим файлам обеспечивают слишком широкую свободу действий, нападающий может воспользоваться этим. Большинство каталогов и файлов, принадлежащих sendmail, должны быть настроены только для чтения для всех пользователей за исключением пользователя, от лица которого работает sendmail. Пользовательские файлы .forward должны быть размещены в каталогах, для которых запрещена запись для группы и для всех остальных пользователей. Сам по себе файл .forward должен быть доступен для чтения-записи только для соответствующего пользователя.

Очень важно обращать внимание на то, от лица какого пользователя вы выступаете, когда читаете почту. Никогда не следует читать почту от лица пользователя root. Вся почта, адресованная пользователю root, должна перенаправляться одному из непривилегированных пользователей. Для этого лучше всего воспользоваться файлом псевдонимов сервера sendmail. Некоторых из почтовых клиентов можно обмануть таким образом, чтобы они запускали исполняемый код, содержащийся внутри почтового сообщения. Вместо того чтобы выяснять, является ли используемый вами почтовый клиент уязвимым для такого рода атак, лучше просто использовать для чтения почты непривилегированную учетную запись. В среде Linux не возникает таких серьезных проблем с вирусами, которые присущи другим платформам, однако фактически все системы могут оказаться жертвами «троянских коней».

domain порт 53

Порт 53 используется системой доменных имен DNS (Domain Name System). В нормальном режиме — то есть при поступлении запроса на определение IP-адреса некоторого узла с известным доменным именем — информация передается с использованием UDP через IP. Однако в некоторых случаях DNS использует TCP — например, для выполнения трансфера зоны. Это происходит потому, что в ходе трансфера зоны через сеть требуется передать объем данных, который, как правило, значительно превышает размер пакета UDP.

Демон DNS (также известный как BIND, Berkeley Internet Nameserver Daemon), запускаемый под именем named (name server daemon), часто подвергается атакам с целью получения доступа к системе на уровне привилегий root. В главе 14 мы подробнее рассмотрим демон сервера имен. В этой главе я расскажу вам о том, как следует установить и запустить этот демон в рамках тюрьмы с измененным корнем файловой системы (change root jail). Другие демоны также могут работать подобным образом. Ранее я уже упоминал о том, что подобная тюрьма применяется также для обеспечения доступа к системе по протоколу FTP.

Некоторые демоны, в особенности web-сервер Apache и FTP-сервер с анонимным доступом, создают свои собственные тюрьмы с измененным корнем, поэтому описываемая процедура может применяться не

только в отношении какого-то конкретного демона. Однако эти демоны могут работать и без создания подобной тюрьмы. При необходимости пользователей системы также можно размещать в тюрьме с измененным корнем, однако при этом администрирование системы существенно усложняется. Как правило, традиционная политика безопасности, основанная на частных пользовательских группах и ограничениях прав доступа внутри файловой системы, является более эффективной и предусматривает более простое администрирование.

tftp, порт 69

Упрощенный протокол передачи файлов TFTP (Trivial File Transfer Protocol) может стать причиной сильной головной боли сетевого администратора, так как именно благодаря TFTP в вашей системе защиты могут открыться очень серьезные дыры. Именно по этой причине я упоминаю этот протокол в данной книге, ведь на практике этот протокол используется относительно редко. Протокол TFTP основан на UDP, однако он создает псевдосоединение. Изначально этот протокол был предназначен для совместного использования с ARP (Address Resolution Protocol) и BOOTP (Bootstrap Ptotocol) и использовался для передачи через сеть файловой системы на рабочую станцию, не оснащенную жестким диском. Если вы хотите обеспечить функционирование бездисковой рабочей станции, используйте поддерживаемый на уровне ядра протокол BOOTP или RARP (Reverse Address Resolution Protocol) и монтируйте файловую систему с использованием NFS. Это значительно более надежный способ, чем использование TFTP. Если только в вашей сети отсутствуют очень старые системы, которые не могут работать без TFTP, вы должны отключить tftp в файле /etc/inetd.conf, так как приемлемых способов защиты этого протокола не существует.

finger, порт 79

Демон finger является постоянным источником информации о вашей системе для внешних взломщиков. В состав OpenLinux входит демон под названием safe_finger. Этот демон можно использовать на локальном узле, не открывая при этом порта. Данная утилита может использоваться также для подключения к демону finger, работающему на удаленном компьютере. По умолчанию в OpenLinux даже эта утилита доступна только для пользователя root. Как я уже отмечал несколько раз, передача подобной информации удаленным узлам не оправданна с точки зрения безопасности.

www, порт 80

Более подробно о web-сервере Apache и его механизмах безопасности будет рассказано в главе 20. Если в системе нет места для столь мощного сервера, как Apache, вы можете воспользоваться небольшим web-сервером, встроенным в ядро версии 2.4.x. Этот web-сервер может быть встроен прямо в ядро или может загружаться в память в виде модуля ядра. Благодаря этому механизму небольшие системы получают возможность использовать небольшой компактный web-сервер для обслуживания простых web-страниц. Помимо сервера Apache в операционной среде Linux можно использовать также другие web-серверы. Все зависит от того, каким образом и для какой цели вы намерены обеспечивать доступ к системе через HTTP. Встроенный в ядро Linux демон http, конечно же, не может полностью заменить собою полноценный сервер httpd, однако этот маленький демон может оказаться весьма полезным. В главе 20 эта новая возможность Linux будет рассмотрена подробнее. Также будет рассказано о том, как этот механизм можно использовать для повышения уровня зашиты в ходе совместного использования с сервером Apache.

pop2, порт 109 и pop3, порт 110

Протокол для чтения почты POP (Post Office Protocol) — это достаточно защищенный протокол, который, как правило, не является источником проблем с точки зрения безопасности. Этот протокол активно используется провайдерами Интернета для обеспечения доступа к электронной почте для своих клиентов. Вместо устаревшего pop2 рекомендуется использовать более новый pop3, однако ни один из этих протоколов не содержит каких-либо бросающихся в глаза уязвимых мест. Следует, однако, учитывать, что в рамках POP пароли передаются через сеть в незашифрованном виде. Это еще одна хорошая причина не использовать учетную запись root для чтения почты.

Служба POP обладает модулем pat, в котором по умолчанию (для OpenLinux) содержится информация, показанная в листинге 12.8.

Листинг 12.8. Модуль pat, по умолчанию используемый службой POP

```
auth     required      /1ib/security/pam_pwdb.so shadow null ok
auth     required      /lib/security/pam_nologin.so
account required      /1ib/security/pam_pwdb.so
password              required    /1ib/security/pam_pwdb.so shadow null ok use_authtok
session  required      /1ib/security/pam_pwdb.so
```

Содержимое этого файла вполне понятно: четыре модуля, которые требуют аутентификацию с использованием пары «имя_пользователя/пароль», и один модуль, который проверяет наличие или отсутствие файла /etc/nologin, запрещающего или разрешающего доступ (соответственно).

sunrpc, порт 11

Технология RFC (Remote Procedure Call), разработанная компанией Sun Microsystems, предназначена для монтирования файловых систем через сеть, однако может использоваться для решения многих других задач. Служба sunrpc используется системой для регистрации других программ RPC. Через этот же порт клиент nfs впервые подключается к серверу nfs через сеть. В зависимости от конкретного приложения это может быть обмен данными UDP или TCP.

Чтобы посмотреть на sunrpc в действии, воспользуйтесь командой rpcinfo -p имя_узла. Вы увидите по крайней мере процесс portmapper, связанный с портом 111 по протоколам TCP и UDP. Однако в отображенном на экране листинге могут быть перечислены также и другие службы RPC, как показано в листинге 12.9.

Листинг 12.9. Вывод утилиты rpcinfo

```
[david@chiriqui david]$rpcinfo -p volcan
program vers proto   port
100000  2  tcp    111  portmapper
100000  2  udp    111  portmapper
300019  1  tcp    743  amq
300019  1  udp    744  amq
100021  1  udp   1024  nlockmgr
100021  3  udp   1024  nlockmgr
100021  1  tcp   1024  nlockmgr
100021  3  tcp   1024  nlockmgr
100001  5  udp    781  rstatd
100001  3  udp    781  rstatd
100001  2  udp    781  rstatd
100001  1  udp    781  rstatd
```

Службы RPC регистрируются при помощи номера программы. Версия необходима только для того, чтобы клиент смог узнать о возможностях сервера. В третьей колонке указываются доступные протоколы. В большинстве случаев доступными являются оба протокола: как UDP, так и TCP. Сведения о портах являются полезными в том смысле, что вы можете определить, какие из портов можно заблокировать с использованием netfilter (или ipchains) чтобы блокировать несанкционированный доступ. Обратите внимание, что за исключением процесса portmapper для большей части программ RPC какие-либо записи в файле /etc/ services отсутствуют.

Службы RPC являются более безопасными, чем такие службы, как tftp, однако их можно обмануть с использованием трюка, когда злоумышленник выдает себя за того, кем он на самом деле не является. Службы RPC не осуществляют подобных проверок и верят клиенту «на слово». Для всех систем, которые позволяют доступ чтения/записи, сервер должен быть настроен на выполнение процедуры *root_squash* (подавление корня) — иными словами, он должен отображать любые запросы на доступ для root (UID 0) на пользователя nobody. Доступ к службам RPC необходимо тщательно контролировать.

ПРИМЕЧАНИЕ

Технология RPC позволяет удаленным системам обращаться к системным вызовам локальной системы через сеть. В общем и целом набор доступных вызовов определяется локальной системой, однако все системы знают, как осуществляются чтение и запись данных файловой системы. Таким образом, эти вызовы на удаленной системе преобразуются в эквивалентные вызовы локальной системы, и если это разрешено, выполняется соответствующая операция.

auth, порт 113

Служба аутентификации запускает демон identd. В настоящее время этот демон почти не используется, так как в Интернете подавляющее число клиентов являют-

ся клиентами Microsoft и не поддерживают службу аутентификациию В Интеренте

почти не осталось серверов, которые запрещают соединение с клиентом в случае, если клиент не отвечает на соответствующий запрос аутентификации, однако многие службы по-прежнему нуждаются в этом.

Некоторые версии Caldera OpenLinux включают в себя версию identd, которая не может корректно завершить работу. В результате система заполняется сотнями работающих процессов inetd. Учитывая тот факт, что данная служба в большинстве мест не используется, вы можете просто отключить ее.

netbios, порты 137-139

Любая система, использующая протокол LanMan (включая системы Linux, на которых работает сервер Samba), использует порты от 137 до 139 для поиска других систем. Именно эти порты используются клиентами Microsoft для поиска узлов в Network Neighbornhood (Сетевое окружение). Если вы используете Samba, вы должны самостоятельно решить, разрешаете ли вы отвечать на запросы на подключение, поступающие через эти порты, и если да, то разрешаете ли вы устанавливать соединения только в рамках внутренней сети или вы планируете обслуживать также внешние соединения.

В системах, не являющихся системами Microsoft (равно как и в системах Microsoft, в которых используется только стек TCP, а прямая поддержка NetBIOS не загружена), любые обращения к NetBIOS обслуживаются с использованием TCP. Если передача данных NetBIOS осуществляется с использованием TCP, эти данные могут передаваться значительно дальше, чем в случае использования протокола NetBIOS в чистом виде. Протокол NetBIOS основан на широковещательной передаче данных, а широковещательная передача данных осуществляется только в рамках локальной сети. Это означает, что широковещательные пакеты не передаются из сети в сеть через шлюзы и маршрутизаторы, и передача данных системам, располагающимся в других сетевых сегментах, не может быть осуществлена. Однако если для передачи пакетов NetBIOS используется TCP, это ограничение может быть снято. Это означает, что сообщения могут быть адресованы системам, располагающимся в других сетевых сегментах, а значит, взломщики получают еще одну потенциальную возможность проникнуть в вашу систему. Таким образом, эти порты необходимо заблокировать для доступа со стороны любых систем, кроме локальных.

imap2, порт 143 и imap3, порт 220

Протокол доступа к электронной почте IMAP (Internet Mail Access Protocol или Interim Mail Access Protocol или Interactive Mail Access Protocol) позволяет работать с электронной почтой в интерактивном режиме. Обеспечивающий его работу сервер является еще одной чрезвычайно слабо защищенной серверной программой, работающей в системе. Многочисленные свидетельства указывают на то, что соответствующий демон неоднократно становился объектом успешных атак взломщиков. Протокол IMAP значительно более мощный, чем протокол POP (Post Office Protocol), однако за его мощь приходится платить высоким риском. Именно поэтому многие провайдеры Интернета не используют его.

IMAP устанавливается как RPM и включает в себя как imapd, так и popd. Если вы удаляете RPM этого протокола, вы также удаляете и popd — возможно, это не то, что вам нужно. Большинство почтовых клиентов могут использовать как ШАР, так и POP, поэтому окончательный выбор в большинстве случаев остается за вами. Но если вы запускаете imapd (в состав OpenLinux входит версия 2, которая использует порт 143), будьте готовы к тому, что злоумышленники попытаются атаковать вашу систему через этот порт. Вы можете блокировать большую часть Интернета в соответствии с тем, откуда к вашей системе должны подключаться пользователи.

Службе IMAP соответствует файл pam.d/imap. По умолчанию в OpenLinux этот файл содержит данные, показанные в листинге 12.10.

Листинге 12.10. Содержимое /etc/pam.d/imap

```
auth      required      /lib/security/pam_pwdb.so shadow nullok
auth      required      /lib/security/pam_nologin.so
account required      /lib/security/pam_pwdb.so
password              required    /lib/security/pam_pwdb.so shadow nullok use_authtok
session  required      /lib/security/pam_pwdb.so
```

Этот PAM не содержит ничего необычного. Все строки, за исключением второй, указывают на необходимость проверки пары «имя_пользователя/пароль». Вторая строка либо разрешает, либо запрещает подключение в зависимости от отсутствия или присутствия (соответственно) файла /etc/nologin. Вы можете повысить уровень защиты ШАР, вставив в файл еще одну строку auth, использующую модуль securetty.so и удалив две метки nullok напротив модуля pam_pwdb.so (пользователи не должны использовать пустых паролей). Модуль securetty.so запрещает пользователю root использовать данную службу (в любом случае почта, адресованная пользователю root, должна быть перенаправлена другому пользователю). То же самое можно сделать и для службы POP.

xdmcp, порт 177 (UDP)

Если в вашей системе начинает работу X, система открывает один или несколько портов в диапазоне от 6000 до 6010. Это обычные порты, через которые удаленные пользователи могут подключиться к X. Однако если вы запускаете xdm, данная служба связывается с портом 177. Для этого используется сокет UDP. Когда сервер X через xdm начинает поиск серверов для управления экраном, по умолчанию для этой цели используется порт 177. Если вы не хотите, чтобы X-серверы использовали сервер, на котором работает xdm для управления экраном, вы должны блокировать порт 177 UDP.

printer, порт 515

Демон печати lpd содержится в RPM-пакете LPRNG. Этот демон устанавливается и настраивается в процессе установки. Вместе с ним устанавливается файл /etc/lpd.perm, который определяет набор разрешений на доступ к демону печати. Файл lpd.perms, устанавливаемый в рамках Caldera OpenLinux по умолчанию, разрешает относительно широкий доступ к принтеру. Иными словами, по умолчанию выполнить распечатку на вашем принтере может любой желающий (листинг 12.11).

Листинг 12.11. Разрешения на доступ к принтеру, определяемые в файле /etc/lpd.perms по умолчанию

```
ACCEPT SERVICE=C SERVER REMOTEUSER=root
ACCEPT SERVICE=S
REJECT  SERVICE=CSU
ACCEPT SERVICE=M SAMEHOST SAMEUSER
ACCEPT SERVICE=M SERVER REMOTEUSER=root
REJECT  SERVICE=M
DEFAULT ACCEPT
```

Переменной SERVICE можно присвоить одно из следующих значений: P — печать (printing); R — размещение в очереди (spooling); C — управление (control); S — состояние (status); U — пользователю разрешается операция lpc; M — удаление из очереди; Q — информация об очереди (queue). Если посмотреть на листинг 12.11, можно заметить, что первая строка разрешает пользователю root управлять работой службы lpd с локального сервера. Вторая строка разрешает любому желающему (включая тех, кто живет на другом конце земного шара) получить сведения о состоянии демона lpd. Никаких проверок IP не осуществляется. Третья строка запрещает управление или использование команд lpc для всех, кому не предоставлен доступ.

В строках 4-7 определяются правила удаления заданий печати из очереди. Эти строки относительно разумны: только пользователь root (UID 0) с локального сервера или тот же самый пользователь с того же самого сетевого узла обладают правом удалять задание печати, всем остальным делать это запрещено. Однако в последней строке стоит DEFAULT ACCEPT, это означает, что то, что не запрещено, считается разрешенным. Таким образом, любой желающий в любой точке земного шара, имеющий возможность вступить в контакт с вашим IP-адресом, сможет распечатать на вашем принтере все, что ему взбредет в голову. Вы уверены, что вам это нужно?

Чтобы заблокировать нежелательные запросы на распечатку, достаточно просто заблокировать порт 515 для всех, за исключением тех, кто работает в составе локальной сети. На момент написания данной книги мне не известно ни об одном риске, связанном с LPRNG, у большинства пользователей Интернета вряд ли возникнет желание печатать что-либо на вашем принтере.

Существует альтернативный метод блокирования нежелательных пользователей принтера. Для этого необходимо изучить синтаксис файла lpd.perms и добавить в него новые правила. Программа LPRNG поддерживает предельно гибкий и мощный набор правил, позволяющий контролировать доступ к принтеру, однако для новичков редактирование файла lpd.perms может оказаться несколько затруднительным. Для изучения этого вопроса следует обратиться к хорошей книге, посвященной администрированию.

СОВЕТ

Для подключения к любой службе TCP вы можете использовать клиента telnet. Используя telnet, вы можете обмениваться данными с любой службой, достаточно знать, что именно ожидает от вас служба. Использовать telnet совместно со службами, основанными на UDP, несколько сложнее, однако при этом работают те же самые принципы.

Команды «r» (rsh, rexec, rlogin), порты 512, 513, 514

Латинская буква «r» в данном случае означает *remote,* то есть «удаленный». Три этих утилиты были разработаны в самом начале развития сетей для того, чтобы использоваться совместно или заменить собой telnet, rsh означает *remote shell* -удаленная командная оболочка, rexec означает *remote execution* — удаленное выполнение заданий, наконец, rlogin означает *remote login* — удаленное подключение. Эти утилиты появились в обиходе в то далекое время, когда люди в значительно большей степени доверяли компьютерным сетям. Благодаря этим трем командам пользователи сети получают возможность доступа к удаленным системам через сеть, и при этом у них даже не спрашивают пароля. Нетрудно себе представить, какую опасность представляют собой эти команды.

Все три команды безоговорочно доверяют сети и сетевым пользователям. В наши дни воспользоваться этим доверием слишком просто. Действие команд не ограничивается обычными пользователями, они действуют и в отношении привилегированных пользователей. Два файла, расположенных либо в каталоге /etc, либо в домашнем каталоге пользователя, могут стать причиной крупных неприятностей для администратора системы. Альтернативой командам «r» является команда ssh — *Secure Shell* (защищенная командная оболочка).

ССЫЛКА

Более подробно о ssh рассказывается в главе 21.

Программа ssh куда более защищена и обеспечивает куда более высокий уровень безопасности, однако несмотря на это на некоторых предприятиях до сих пор предпочитают использовать команды «r». Если вы имеете дело с такой сетью, вы должны хорошенько продумать защиту этих служб. Помните, что любые связанные с ними данные передаются через сеть в незашифрованном виде — никакое кодирование не применяется, и поэтому любые пароли могут быть похищены злоумышленниками при помощи tcpdump или любой другой программы, «прослушивающей» сеть.

ПРИМЕЧАНИЕ

Утилиты или программные средства, которые позволяют получать все пакеты, передаваемые через сеть, вне зависимости от того, кому они адресованы, обозначают английским термином sniffer. Подобные инструменты разработаны для того, чтобы помочь системным администраторам решать проблемы, связанные с передачей данных через сеть. Однако в наши дни подобные средства часто используются злоумышленниками для того, чтобы похищать передаваемые через сеть пары «имя_ пользователя/пароль».

В файле /etc/services все эти службы обозначены без префикса «r», то есть в этом файле они обозначены как shell, exec, login. Точно так же ссылается на них и файл inetd.conf, однако в конце каждой соответствующей строки можно увидеть rshd, rexecd и rlogind (соответственно). Если все эти строки закомментированы (что я вам настоятельно рекомендую сделать) и демон inetd послал сигнал SIGHUP, значит, ваша система не обеспечивает доступ с использованием этих служб.

Если же вы вынуждены обеспечить доступ клиентов с использованием этих служб, вы должны обеспечить должную аутентификацию. Начать следует с изучения подкаталога /ect/pam.d. Здесь можно обнаружить три раздельных файла: rsh, rexec и rlogin, которые, как нетрудно догадаться, соответствуют рассматриваемым службам. Содержимое этих файлов показано в листингах 12.12,12.13 и 12.14.

Листинг 12.12. Содержимое файла /etc/pam.d/rsh по умолчанию

```
auth required /lib/security/pam_rhosts_auth.so
auth required /lib/security/pam_nologin.so
account required /lib/security/pam_pwdb.so session required
/lib/security/pam_pwdb.so
```

Первая строка файла /etc/pam.d/rsh может содержать несколько параметров, о которых рассказывалось в главе 1. Эта строка разрешает использование файлов rhosts в домашних каталогах пользователей.

Листинг 12.13. Содержимое файла /etc/pam.d/rexec по умолчанию

```
auth      required        /lib/security/pam_pwdb.so shadow nullok
auth      required        /Iib/security/pam_nologin.so
auth      required        /lib/security/pam_listfile.so file=/etc/ftpusers item=user
sense=deny onerr=succeed account      required        /lib/security/pam_pwdb.so
```

Файл rexec предусматривает несколько более строгую защиту, так как использование файлов rhosts запрещено. Однако вы можете разрешить это, просто добавив в файл еще одну строку. Обратите внимание, что rexec использует файл /etc/ftpusers для ограничения доступа через сеть точно так же, как делает это ftp. Конечно же, подобную строку можно добавить и в файл rsh. Еще одна возможность демонстрируется в файле /etc/pam.d/rlogin, содержимое которого представлено в листинге 12.14.

Листинг 12.14. Содержимое файла /etc/pam.d/rlogin no умолчанию

```
auth    required        /lib/security/pam_securetty.so
auth    sufficient      /lib/security/pam_rhosts_auth.so
auth    required        /lib/security/pam_pwdb.so shadow nullok
auth    required        /lib/security/pam_nologin.so
account                 required    /lib/security/pam_pwdb.so
#password     required       /lib/security/pam_cracklib.so
password                required    /lib/security/pam_pwdb.so shadow nullok use_authtok
sessionrequired         /lib/security/pam_pwdb.so
```

В файле /etc/pam.d/rlogin первая строка запрещает пользователю root подключаться к системе через любое устройство, за исключением локального tty. Вторая строка разрешает подключение без ввода пароля в случае, если подключающийся пользователь обладает файлом rhosts, содержащим подходящую запись (или соответствующая запись содержится в файле /etc/hosts.equiv).

Если строка 2 не выполняется, происходит тестирование строк 3-8 (за исключением закомментированных строк, таких как строка 6). Обо всех этих модулях уже подробно рассказывалось в главе 1.

Файл hosts.equiv является глобальным файлом, который устанавливается системным администратором. В общем и целом, этот файл используется для того, чтобы сообщить локальному узлу о том, что перечисленные в этом файле пользователи удаленных систем (идентифицируемых при помощи имени или IP-адреса) эквивалентны пользователям, зарегистрированным под таким же именем на локальной системе.

Пользовательский файл rhosts служит для той же самой цели, что и глобальный файл hosts.equiv, однако разным пользователям можно поставить в соответствие разные файлы rhosts. Записи файла rhosts показаны в листинге 12.15.

Листинг 12.15. Некоторые из записей файла rhosts

```
<узел>                  Разрешается доступ того же пользователя
(hosts.equiv: все пользователи) с узла <узел> <узел> <пользователь>     Разрешается доступ для
<пользователь> с узла <узел>
-<узел>                 Пользователям с узла <узел> доступ запрещен
<узел> -<пользователь>  Запрещается доступ для
<пользователь> с узла <узел>
Следующие записи применяются только в режиме "promiscuous" (прослушивание):
+                       Подключение разрешается для кого угодно с любого узла
+ +                     То же самое, что и предыдущая запись
+ <пользователь>        Указанному пользователю разрешается подключаться откуда угодно
<узел> +                Разрешается подключаться всем пользователям указанного узла
+ -<пользователь>       Запрещается подключаться указанному пользователю с любого узла
```

За исключением последней записи я не рекомендую вам использовать модификатор «плюс» (+), так как в этом случае ваша система становится широко открытой для постороннего доступа.

Другие службы

На вашей системе вы можете запустить множество других служб. Запуская службу, вы должны обратить внимание на следующие детали.

- Работает ли служба от лица привилегированного пользователя? Сокеты с номерами ниже 1024 могут быть связаны только пользователем root, поэтому соответствующие службы должны быть запущены от лица пользователя root. Сокеты с номерами выше 1024 могут использоваться любыми пользователями, включая root.

- Может ли служба работать от лица непривилегированного пользователя? Если да, то запустите ее именно в таком режиме.

- Можно ли разместить службу в тюрьме с измененным корнем (change root jail)? (Более подробно об этом рассказывается в главе 14.)

- Можете ли вы блокировать или как-либо ограничить доступ к порту?

- Существует ли надобность в этой службе? Если нет или вы не уверены, отключите ее.

Заключение

В данной главе мы рассмотрели некоторые традиционные службы и связанные с ними уязвимые места. В главе также продемонстрирована работа двух коммуникационных протоколов (все коммуникационные протоколы функционируют сходным образом).

13 Атаки DoS и как они работают

В данной главе рассматриваются следующие вопросы:
- что такое атака DoS (Denial of Service — отказ обслуживания);
- что такое ping-затопление (ping flooding);
- что такое атака TCP ACK;
- что такое ping-атаки;
- смягчение последствий атак DoS.

DoS означает Denial of Service — отказ в обслуживании. Атаки этой категории могут иметь множество форм, однако суть одна и та же: вы не можете получить доступ к ресурсу, за который вы заплатили. Все ресурсы, имеющие отношение к компьютерам, оплачиваются, не существует ничего бесплатного. Возможно, оплата осуществляется неявно и не вами, однако так или иначе кто-то должен платить. Атаки категории DoS, как правило, совершаются озлобленными или разочарованными людьми, которые не находят иного способа нанести ущерб. Такие атаки могут быть как низкотехнологичными, так и более совершенными в техническом смысле — все зависит от типа атаки и способов, применяемых для того, чтобы скрыть, кто именно является автором нападения. Существует одна чрезвычайно сложная в техническом смысле атака DoS, однако количество инженеров, способных организовать доменное нападение (doinam hijacking), не так велико.

Атаки DoS могут принимать множество форм, однако вне зависимости от формы их эффективность сложно подвергать сомнению. Некоторые интернет-провайдеры были вынуждены закрыть свой бизнес из-за того, что один очень озлобленный бывший клиент решил отомстить, используя атаку DoS. Атака DoS делает провайдера неспособным обеспечивать доступ к Интернету для своих клиентов. А когда провайдер не способен обслужить своих клиентов, рано или поздно его клиенты разбегутся кто куда. Если провайдер является небольшой только начинающей свое развитие компанией, он может лишиться всех своих клиентов в течение буквально пары дней. Очень многие небольшие и даже средние по размеру предприятия не в состоянии бороться с атаками этой категории. В подобной ситуации очень важно знать, что именно надо делать, так как если вы не знаете этого, значит, ваш сервер становится недоступным для клиентов, а вы теряете доступ к сети.

Что такое ping flooding

Несколько форм атаки DoS обозначаются общим названием ping flooding (ping-затопление). В нормальных условиях утилита ping посылает на целевой узел один пакет ICMP в секунду. Однако если заставить ping посылать пакеты ICMP настолько быстро, насколько позволяет самая медленная линия связи, входящая в состав маршрута от localhost до целевого узла, то происходит ping-затопление. Это означает, что если линия связи достаточно быстрая, то через канал передается настолько много пакетов ICMP, что передача всего остального трафика существенно замедляется. Еще одним способом создания подобной нагрузки на линию связи является программа spray, которая разработана для измерения пропускной способности канала путем максимального заполнения этого канала передаваемыми данными. Однако для работы spray необходимы клиентская и серверная части, поэтому spray можно использовать только между двумя специально подготовленными для этой цели компьютерами.

Осуществить ping-затопление чрезвычайно просто. В Интернете можно найти множество специально предназначенных для этой цели инструментов, ориентированных на использование в отношении самых разных операционных систем. В операционной среде Linux выполнить ping-затопление позволяет сама утилита ping, для этого используется специальный ключ -f. Однако воспользоваться этим ключом может только пользователь с UID 0. По очевидным причинам обычным пользователям запрещается использовать этот режим работы ping.

Затопление ping воздействует не только на цель, но и на клиента. Многие модифицированные программы специально переписывают заголовок ICMP-пакета таким образом, чтобы изменить адрес-источник. Благодаря этому злоумышленник подвергает нападению две системы одновременно, однако при этом он обязан продолжать посылать пакеты ping. В процессе ping-затопления на экране могут

отображаться сведения, показанные в листинге 13.1.

Листинг 13.1. ping-затопление смежного узла

```
volcan# ping -f chiriqui
PING chiriqui.pananix.com (192.168.0.2):  56 data bytes
.............................

--- chiriqui.pananix.com ping statistics  ---
7321 packets transmitted,  7299 packets received,  0% packet loss round-trip min/avg/max =
0.4/3.3/21.6 ms
```

Данные результаты получены на никем другим не используемой сети между двумя системами, соединенными 10-мегабитным каналом связи. Несколько иные результаты можно получить, если направить ping на локальный узел localhost. Вывод подобной команды показан в листинге 13.2. Обратите внимание на скорость пакетов и количество потерянных пакетов. Атаки типа Denial of Service основаны на потере пакетов.

Листинг 13.2. ping-затопление локального узла localhost

```
volcan# ping -f localhost
PING localhost (127.0.0.1):  56 data bytes
.....................................................................
.....................................................................


--- localhost ping statistics  - --
5448 packets transmitted,  1022 packets received. 81% packet loss
round-trip min/avg/max = 0.3/122.3/342.1 ms
```

Еще один прием, который может как сработать, так и не сработать, это использование ping в отношении широковещательного адреса. Этот трюк может не сработать, так как большинство современных маршрутизаторов не передают пакеты ping, направленные на широковещательный адрес. Успешность подобной атаки зависит также от того, соответствует ли IP-стек целевой системы всем требованиям стандарта, определенного в RFC.

ПРИМЕЧАНИЕ

RFC (Request For Comments) — это документы, в которых содержится информация о том, каким образом функционирует Интернет. Иными словами, в документах RFC определяется стандарт — несоответствие документам RFC означает, что используемое вами программное обеспечение ведет себя не так, как все остальные программы Интернета, при этом возможно, что вы не сможете взаимодействовать с другими системами. Следует отметить, что Linux является одной из немногих систем, поддерживающих IP-стек, на все 100 % совместимый с требованиями RFC.

Чтобы пояснить сказанное, приведу пример. Представьте, что у вас есть сеть, в которой работает несколько систем Linux (или других систем, совместимых с Unix, таких как SCO OpenServer, HP-UX, AIX или Solaris, — обо всех этих системах известно, что они в достаточной степени совместимы с RFC), а также несколько систем Microsoft (Windows 9x или NT 4). В подобной среде вы можете легко протестировать различие в поведении утилиты ping.

Для этого в сети выполните следующую команду:

```
ping -c 2 <широковещательный_адрес_сети>
```

На экране появится информация, подобная показанной в листинге 13.3.

Листинг 13.3. Сведения о выполнении ping в отношении широковещательного адреса

```
64 bytes from 192.168.0.1:   icmp_seq=0 ttl=255 time=0.3 ms
64 bytes from 192.168.0.2:  icmp_seq=0 ttl=255 time=0.6 ms (DUP!)
64 bytes from 192.168.0.1:   icmp_seq=l ttl=255 time=0.2 ms
```

Если вы осуществляете однократный ping-тест, вы увидите только один адрес, так как локальная система всегда отвечает на запрос быстрее, чем любая другая. Утилита ping ожидает только один ответ, поэтому она завершит работу до того, как получит ответ от остальных систем. Однако если вы будете ожидать несколько ping-ответов, вы увидите ответы от всех сетевых узлов, не являющихся узлами Microsoft.

ПРИМЕЧАНИЕ

Если запустить ping с одного из узлов Microsoft, вы получите ответы только от сетевых узлов Microsoft. Дело в том, что только IP-стек, разработанный Microsoft, отвечает на ping-запросы подобным образом (ответ на широковещательные сообщения не осуществляется). Это означает, что IP-стек, разработанный Microsoft, не

соответствует стандарту, определенному в RFC. Это наиболее ярко выраженная несовместимость между стеками Microsoft и стандартом RFC, однако она в значительной степени влияет на сетевой обмен данными по протоколу IP. Таким образом, не следует ожидать, что сетевомузел Microsoft будет корректно вести себя в рамках сети.

Только что описанный тест не сработает в сетях категории «точка-точка», а также его нельзя осуществить через маршрутизатор. Например, если вы попытаетесь использовать данный трюк для того, чтобы определить, какие еще системы, не являющиеся системами Microsoft, работают в такой сети, как @home (американская кабельная сеть), вы увидите только одну вашу систему. Дело в том, что кабельный модем является маршрутизатором и не пропускает широковещательные ping-запросы. Однако если маршрутизатор настроен неправильно или на нем отключен механизм блокирования широковещательных ping-запросов, описанный прием сработает.

Атаки типа ping flood не являются эффективными. Подобные атаки нельзя поддерживать в течение длительного времени. Злоумышленник вынужден запускать ping на системе, которая в последующем не сможет использоваться в сети. Дело в том, что подобную систему легко идентифицировать, и одного телефонного звонка провайдеру, как правило, достаточно для того, чтобы отключить эту систему от Интернета. Даже если адрес-источник модифицирован, пересылку таких модифицированных пакетов требуется продолжать с изначального адреса-источника. Нападающему сложно определить, была ли атака успешной, кроме того, к системе, с которой осуществляется атака, привлекается слишком много внимания.

Существует одно исключение: атака, которая, как правило, срабатывает. Эта атака обозначается английским термином *smurf*. Атака выполняется при помощи программы с таким же именем, которую легко найти в Интернете. Если в распоряжении злоумышленника находится несколько систем (как правило, злоумышленник использует для этой цели несколько чужих взломанных им систем), нападающий может использовать их все для совместного нападения на цель. Программа smurf поддерживает широкие возможности по модификации заголовка IP, благодаря чему нападающий может организовать весьма эффективную атаку, к тому же его будет очень сложно выследить.

Что такое атаки SYN DoS и как с ними бороться

Значительно более эффективными являются атаки, основанные на том факте, что основанные на IP системы используют последовательность SYN-ACK, для того чтобы установить соединение и покинуть очередь ожидания. Это достаточно старая атака, однако она до сих пор эффективна против некоторых систем. Если атака этой категории будет предпринята в отношении стандартной системы OpenLinux, для администратора этой системы это, скорее всего, будет не более чем легким раздражающим фактором.

Атака этой категории основана на том факте, что в прошлом (а также и в наши дни на некоторых системах) очередь ожидания обладала небольшим размером и ее можно было быстро заполнить. Подразумевается, что при создании каждого нового соединения серверный процесс некоторое время ждет в очереди, а затем покидает ее, однако это происходит только в случае, если завершается последовательность SYN-ACK. Только после завершения этой последовательности соединение TCP считается установленным. Нападающий может отправить запрос на установку соединения, а затем игнорировать адресованный ему пакет с установленным битом SYN (или сообщить серверу, что источник запроса вообще является совершенно другой системой, тогда пакет SYN будет направлен системе, которая не будет на него отвечать). В результате сервер будет ожидать в течение длительного времени, а запрос на подключение будет по-прежнему находиться в очереди ожидания. Пока сервер ждет, никто другой не сможет к нему подключиться. Нападающий может продолжать посылать серверу подобные запросы, создавая тем самым непреодолимое заграждение, связывая очередь ожидания и блокируя остальные попытки подключения к системе.

ПРИМЕЧАНИЕ

Модификация заголовка пакета с целью изменения адреса-источника — это одна из форм маскировки злоумышленников в Интернете. Маскировка в Интернете обозначается английским термином spoofing.

Когда Интернет впервые подвергся подобным нападениям, в первую очередь пострадали многие интернет-провайдеры. Единственным способом восстановить обслуживание был останов и перезапуск атакованной службы. Для некоторых операционных систем это означало полную перезагрузку системы. Однако если атака продолжалась, доступ к службе снова блокировался.

Чтобы снизить эффективность подобных атак, хакеры ядра Linux разработали несколько специально предназначенных для этой цели механизмов. Первый из способов противодействия предусматривает использование SYN cookies. Этот механизм позволяет обнаружить атаку SYN-ACK и блокировать попытку связать очередь ожидания. Данный метод по-прежнему может использоваться теми, кто подозревает, что

его постоянно атакуют с использованием SYN DoS, однако в рамках данного метода расходуется достаточно большое количество системных ресурсов, к тому же благодаря появлению новых методов в настоящее время метод SYN cookies используется редко. Данный механизм, даже если он включен в состав ядра в процессе компиляции, по умолчанию выключен, поэтому вы должны включить его при помощи следующей команды:

```
echo 1 >/proc/sys/net/ipv4/tcp_syncookies
```

Механизм криптографически проверяет каждое из соединений. Если возникает подозрение, что с некоторых IP-адресов в отношении вашей системы осуществляется атака SYN DoS, механизм SYN cookies сообщает вам эти адреса. Однако вы должны учитывать, что эти IP-адреса извлекаются из поступающих к вам пакетов. Таким образом, это могут оказаться ложные подделанные IP-адреса или адреса систем, которые злоумышленник использует для атаки без ведома их владельцев. Общаясь с людьми, которые несут ответственность за данные IP-адреса, вы должны понимать, что это могут оказаться совершенно невинные люди.

Со времени появления SYN cookies размер очереди ожидания существенно увеличился, благодаря чему количество ожидающих в очереди запросов возросло. Само по себе это обстоятельство снизило эффективность небольших по масштабу атак или атак с использованием медленных соединений. Непреклонный и решительный взломщик все же может попробовать заполнить очередь ожидания и блокировать большое количество соединений от остальных клиентов, однако длинная очередь ожидания является более эффективным методом борьбы с подобного рода нападениями, чем SYN cookies. Для успешного нападения на систему, обладающую длинной очередью ожидания, злоумышленник может использовать одновременно несколько взломанных им систем (как в случае smurf).

Третьим средством борьбы с атаками SYN DoS является изменение времени в течение которого система ожидает завершения последовательности SYN-ACK прежде чем удалить незавершенный запрос из очереди ожидания. Также в кол ядра включен алгоритм, автоматически удаляющий из очереди «подвисшие» запросы на подключение в случае, если очередь заполнена. В подобной ситуации удаление запросов происходит даже раньше, чем истечет обычный таймаут, однако подобное происходит достаточно редко. Только чрезвычайно перегруженный сервер вынужден освобождать свою очередь от запросов подобным образом.

Если ваш сервер сильно загружен и подвергается атаке SYN DoS, при необходимости вы можете изменить значения некоторых параметров в самом ядре. Этс непростая задача. Значения всех настраиваемых параметров определяются в исходных файлах ядра, а именно в файле tcp.h (этот файл, как правило, можно обнаружить в /usr/src/linux/indude/net/tcp.h). Здесь определяется допустимое количество открытых соединений TCP, количество повторных попыток SYN, время ожидания между попытками и т. п.

ВНИМАНИЕ

Изменение значений, по умолчанию заданных в файле, может привести к нежелательным результатам. Например, количество повторных попыток (TCP_SYN_RETRIES) в настоящее время устанавливается равным 10. Если вы уменьшите это значение слишком сильно, некоторые клиенты не смогут подключиться к вашей системе. Это может произойти из-за отброшенных пакетов (по причине коллизий, слишком крупных фрагментов и установленного бита DNF — do not fragment— и т. п.) или из-за того, что выбранный верхний порт оказался занятым и его следует определить заново. Могут возникнуть также и другие причины. Если значение выбирается слишком большим, вы можете столкнуться с другими сложностями.

Если вы заглянете внутрь файла tcp.h, вы увидите множество ссылок на состояния TCP-соединения, например Fin Wait, Time Wait, Established, Listening, Closing и т. п. Вы также обнаружите, что код хорошо комментирован.

ПРИМЕЧАНИЕ

В коде, написанном на языке С, комментарии содержатся между комбинациями символов / и */ и могут включать в себя строки, начинающиеся с символа звездочки (*). Не следует путать этот символ со знаком решетки (#), который используется для обозначения комментариев в сценариях оболочки. В языке С символ решетки (#) стоит в начале таких директив, как include, define и т. п. Начинающиеся с символа решетки строки не являются комментариями, и компилятор осуществляет их обработку.*

Более старые атаки

Еще одна достаточно старая атака, от которой до сих пор могут пострадать некоторые системы, называется big ping (большой ping) или ping of death (смертельный ping). Вспомним материал более ранних

глав, где речь шла о битах и байтах. Байт, как правило[7], состоит из 8 бит. Для хранения значения длины пакета отводится два байта, то есть 16 бит. Любое 16-битное число лежит в диапазоне от 0 до 65 535. Таким образом, размер пакета не может превышать 65 535 байт. Сообщения ICMP по определению не могут быть крупнее, поэтому многие системы не проверяют (до сих пор) размер ICMP-пакета. На практике единственным ICMP-пакетом, размер которого с легкостью можно изменить, является ping-пакет.

Злоумышленники обнаружили способ модифицировать код, создающий ping-пакеты, таким образом, чтобы стало возможным создание очень больших пакетов, размер которых превышает установленный предел. ICMP-заголовок ping-пакета в нормальных условиях состоит из 8 байт. Стандартный пакет добавляет к этому 56 байт дополнительно, чтобы размер пакета стал равным 64 байт.

В Linux размер пакета (который по умолчанию составляет 64 байт) можно изменить. Для того чтобы указать количество дополнительных байтов, можно воспользоваться ключом -s. Однако стандартная утилита ping не позволяет использовать сколь угодно большое значение. Допустимыми являются значения от 1 до 65 468. При использовании значений от 1 до 65 454 утилита ping работает в нормальном режиме. Если вы укажете одно из значений от 65 465 до 65 468, утилита ping может вообще никак не сообщить о ходе операции либо на экране появятся сведения о множестве пакетов. Если же совместно с ключом -s указать значение большее, чем 65 469, утилита ping отобразит на экране сообщение о том, что пакет слишком большой. Благодаря этому отправить в сеть ping-сообщение, размер которого превышает максимально допустимый предел, невозможно. Таким образом, у систем, беззащитных перед атакой big ping, никаких проблем не возникает. Однако в Linux исходный код утилиты ping доступен для всех желающих, благодаря чему кто угодно может модифицировать его таким образом, чтобы стало возможным сгенерировать и отослать в сеть пакет любого размера.

В сети до сих пор используются системы, работа которых нарушается в случае приема крупных пакетов ping. Очевидно, что такие системы требуют защиты. Если в сети работает брандмауэр Linux и в составе ядра версии *2.2.x* скомпилирован параметр CONFIG_IP_ALWAYS_DEFRAG=y (в ядре версии 2.4.x от этого параметра решили отказаться — данные функции выполняет netfilter), в этом случае фрагменты не будут передаваться и системы будут защищены от внешнего нападения.

Смягчение последствий атак DoS

Прежде чем вы доведете себя до крайности и вам начнут мерещится атаки DoS там, где их на самом деле нет, вы должны почувствовать, каким образом и по каким законам изменяется ваш сетевой трафик. При этом вы должны принимать во внимание также другие системы, которые подключены к тому же самому сетевому кабелю, что и вы. Если вы подозреваете, что подверглись атаке типа DoS, прежде всего необходимо рассмотреть другие причины, по которым производительность вашей службы могла понизиться. Возможно, причиной этого является повышенный интерес к вашей системе или к системе, которая работает по соседству с вами и использует тот же самый канал связи (например, причиной может стать ссылка на популярный сайт). В прошлом многие предприятия уже не однократно убеждались в том, насколько эффективной может оказаться рекламная кампания. Достаточно вспомнить демонстрацию последних моделей женского нижнего белья Victoria's Secret Show через Интернет в 1999 году. Реклама этого шоу была настолько эффективной, что количество желающих посмотреть на шоу через Интернет в несколько раз превысило ожидания организаторов. При этом сам сервер был в состоянии обслужить необходимое количество запросов, однако емкости канала связи, по которому он обменивался данными с Интернетом, оказалось недостаточно. Соответственно пострадали также и другие системы, подключенные к этому же каналу. В некотором отношении это можно назвать отказом в обслуживании (Denial of Service), однако подобная ситуация возникла ненамеренно.

ПРИМЕЧАНИЕ

Чтобы получить представление о нормальном сетевом трафике, вы можете воспользоваться программным средством Multi-Router Traffic Grapher (MRTG). Если, используя данную программу, вы проследите за маршрутизатором, который связывает вас с Интернетом (а возможно, и за маршрутизатором, который стоит следующим по цепочке), вы сможете получить неплохое представление о том, как меняется трафик. Программа MRTG отображает сведения о нагрузке на каналы в наглядном графическом виде. Прежде чем приступать к наблюдению за маршрутизаторами, следует спросить разрешения у тех, кому эти маршрутизаторы принадлежат (если эти маршрутизаторы принадлежат не вам). Программу MRTG можно получить по адресу: http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html

Также следует убедиться в том, что на вашей системе не работает никаких программ, которые нуждаются в дополнительной пропускной способности. Иными словами, возможно, что причиной отказа в

[7] В общем случае размер байта в битах зависит от архитектуры компьютера. — *Примеч. ред.*

обслуживании являетесь вы сами. К счастью, это очень легко исправить. Вы также должны проверить возможность аппаратных проблем, например неисправные сетевые карты, которые требуется заменить.

После того как вы отмели все возможные ненамеренные причины, перед вами встает проблема: что делать дальше? После изучения содержимого файлов журналов у вас, возможно, возникнут какие-нибудь идеи, однако сведения, полученные от SYN cookies, нельзя считать достоверными. Если сеть перегружена, механизм SYN cookies может выдавать неправильную информацию. Кроме того, если злоумышленник модифицирует заголовки IP-пакетов, то с помощью SYN cookies вы не сможете определить IP-адрес этого злоумышленника. В случае атаки механизм SYN cookies всего лишь позволяет обеспечить большее количество подключений к серверу, однако за это приходится платить перерасходом серверных ресурсов.

Выбор варианта действий во многом определяется тем, как именно вы соединены с Интернетом. Те, кто использует для подключения к Интернету кабельный модем, скорее всего, вообще не столкнутся с подобными атаками. Компании, предоставляющие доступ к Интернету с использованием кабельных модемов, делают все от них зависящее, чтобы обеспечить своим клиентам достаточную пропускную способность канала связи. Если же возникает атака типа Denial of Service, ее можно будет решить на уровне вашего провайдера.

Если вы используете линию DSL, значит, вы используете соединение типа «точка-точка», которое соединяет ваш сервер с вашим провайдером. При помощи traceroute вы сможете определить, подвержена ли атаке сама линия DSL или нагрузка создается только на более дальний от вас канал связи (на который вы вряд ли сможете повлиять). Однако лучше всего продолжать функционирование так, как будто никакой атаки нет или она оказалась неэффективной. Почему? Во-первых, рано или поздно атака будет остановлена. Во-вторых, если нападающий подумает, что атака неэффективна, он откажется от нее и попробует заняться чем-нибудь другим.

Если атака направлена только на одну систему и использует только один маршрут, вы можете попытаться переместить исходящий трафик на другую систему и использовать другой шлюз по умолчанию. Благодаря этому вы снизите нагрузку на основной маршрут (если, конечно, вы обладаете подобным резервным маршрутом).

Если атаке подвержена только одна служба, остановка и перезапуск этой службы временно очистят очередь запросов. Однако она начнет заполняться вновь, таким образом, используя этот метод, вы сможете решить проблему только временно. Если пакеты поступают от одного адреса или из одной сети, вы можете попробовать просто отбрасывать все пакеты, поступающие из этой сети (см. главу 16).

Заключение

В данной короткой главе я коротко рассказал о наиболее сложных проблемах сегодняшнего Интернета. Мы с вами рассмотрели несколько разновидностей атак DoS и то, как они осуществляются. Вы получили представление о том, насколько просто организовать подобную атаку, однако при этом вы также узнали о том, насколько малоэффективными могут быть такие атаки. Я также рассказал о некоторых мерах, которые можно предпринять для того, чтобы смягчить последствия подобных атак.

14 Устранение уязвимых мест

В данной главе рассматриваются следующие вопросы:
- как понизить уязвимость;
- уменьшение количества служб;
- построение DNS-сервера с измененным корнем;
- восстановление после нападения;
- обнаружение нападения;
- восстановление файловой системы.

В начале данной книги я рассказал вам о том, как устроена ваша система и как работает сеть, к которой она подключена. Теперь, когда вы получили базовые сведения, необходимые для принятия важных решений в области защиты, вы можете приступить к использованию этих сведений на практике. В данной главе мы вначале изучим, каким образом можно предотвратить нападение, а затем я расскажу о том, как можно определить, что в вашу систему проник злоумышленник.

Вновь хочу напомнить, что полностью безопасной можно считать систему, которая тщательно запакована в картонную коробку, в которой вам ее продали. Однако если вы хотите получить от системы хоть какую-то пользу, вы должны запустить ее. Пока система не соединена с какими-либо другими системами, опасность исходит только с консоли и программ, которые могут быть запущены при помощи консоли. Если вы боитесь всего на свете, вы можете раздобыть исходный код всего необходимого вам программного обеспечения, прочесать весь этот код на наличие чего-либо, напоминающего «троянских коней», и только убедившись в том, что код полностью безопасен, откомпилировать его и приступить к его использованию. Однако, я надеюсь, что таких параноиков среди читателей данной книги не много.

На самом деле достаточно получить хорошую копию хорошо известного и достаточно распространенного комплекта поставки Linux (например, такого, который записан на прилагаемом к данной книге компакт-диске) и полностью довериться компании-поставщику. Будьте уверены, что компания-поставщик сделала все от нее зависящее для того, чтобы избавить поставляемый ею исходный и исполняемый код от «троянских коней». Любая компания-поставщик начинает работу над собственным комплектом с исходного кода и формирует бинарные пакеты самостоятельно. Пакеты, входящие в состав OpenLinux, собраны компанией Caldera. Иными словами, на базе предыдущей версии формируется новый набор пакетов, все это устанавливается на компьютере, заново пакуется и записывается на компакт-диск, который вы в дальнейшем и получаете. Это относится ко всем пакетам, за исключением тех, которые записаны в подкаталоге contrib. В этом каталоге содержатся пакеты, собранные третьими лицами и предоставленные компании Caldera для включения в комплект поставки OpenLinux. Также компания Caldera не имеет отношения к содержимому подкаталога security. За содержимое этого каталога несет ответственность автор, который собрал пакеты этого каталога с использованием Caldera OpenLinux 2.3.

ПРИМЕЧАНИЕ

На компакт-диске, прилагаемом к данной книге, содержится полный комплект бинарных файлов, входящих в состав OpenLinux 2.3 (исходные файлы можно скачать с FTP-узла компании Caldera Systems). Также на компакт-диске содержится каталог col/security, который сформирован автором для решения задач безопасности, рассматриваемых в данной книге.

СОВЕТ

Заниматься формированием защиты вашей системы следует до того, как вы подключите ее к окружающему миру. Если вы вначале обеспечите связь с внешним миром, а затем займетесь обеспечением защиты, вы не сможете полностью доверять своей системе.

После того как у вас появится выбранный вами комплект поставки Linux, вы должны установить его на компьютере. На этом этапе вы не должны быть подключены к какой-либо сети. Имейте в виду, что сразу после завершения установки ваша система является наиболее уязвимой. Начните с настройки сетевых карт, однако не подключайте к ним сетевой кабель (по крайней мере, тот, который связывает систему с Интернетом). Имейте в виду, что в некоторых конфигурациях (система DNS загружена и в файле /ect/resolve указывает на узел 127.0.0.1 как на первый сервер имен, к которому следует обращаться)

некоторые демоны, в частности amd — *automount daemon* — демон автоматического монтирования, могут подвиснуть до того момента, пока не истечет таймаут ожидания ответа от DNS, который может никогда не поступить. Если подобное двухминутное ожидание раздражает вас, отключите amd до тех пор, пока вы не подключитесь к Интернету. То же самое относится и к sendmail

После того как ваша система заработает, используйте информацию из данной книги и других ресурсов для того, чтобы настроить вашу систему на наивысшем возможном уровне защиты. После этого можно подключить ее к Интернету.

Ограничение повреждений

На данном этапе вы установили вашу систему и проверили отсутствие наиболее явных дыр в системе безопасности. Вы узнали конфигурацию сети, вы проверили каждую из предлагаемых вами служб и остановили работу тех служб, которые вы не намерены использовать. Теперь следует тщательней проанализировать службы, которые вы намерены сделать доступными для внешнего мира.

Каким образом можно убедиться в том, что служба не может стать причиной взлома системы? Как уже отмечалось ранее в данной книге, цель любого взломщика — стать пользователем root. Только пользователь root обладает правом связывать привилегированные порты. Именно через эти порты взломщик пытается получить доступ к командной оболочке и таким образом взломать систему.

У вас есть две возможности. Во-первых, вы можете перенаправлять любые соединения с привилегированного порта на непривилегированный порт, который используется от лица непривилегированного пользователя. Во-вторых, вы можете запустить привилегированный процесс в тюрьме с измененным корнем (change root jail). Следует учитывать, что далеко не каждую службу можно запустить как непривилегированную. Например, демон протокола сетевого времени должен обладать возможностью перенастраивать системные часы, а правом на это обладает только пользователь root. Таким образом, принимая решение, вы должны хорошо понимать, что именно делает тот или иной демон и какими возможностями он должен обладать. Однако если служба всего лишь читает статические документы, то если вы сделаете ее непривилегированной и будете перенаправлять ей любые запросы, поступающие на стандартный порт этой службы, никаких проблем не возникнет. Таким образом, если вы имеете дело с демоном, который не осуществляет какую-либо запись (равно как и не вносит в систему каких-либо других изменений), вы можете запустить этот демон от имени непривилегированного пользователя, которому разрешается только читать документы, с которыми работает данная служба (документы по-прежнему принадлежат пользователю root, однако могут быть прочитаны пользователем, от имени которого запущен данный демон). Этот демон должен быть связан с портом, номер которого больше 1024, однако вы можете использовать специальную программу, которая будет перехватывать запросы, поступающие через стандартный порт, и перенаправлять их в порт, с которым связан данный демон. О том, как организовать подобное перенаправление, рассказывается в главе 18.

ПРИМЕЧАНИЕ

Некоторые серверные приложения обладают встроенной системой защиты: они запускаются от лица пользователя root, однако в дальнейшем меняют уровень привилегий и начинают работать от лица непривилегированного пользователя. Однако подобное переключение не делает их абсолютно защищенными, так как некоторые фрагменты кода подобных программ зачастую все равно выполняются на уровне привилегий root — именно их, как правило, и пытаются взломать злоумышленники.

В некоторых случаях вам сопутствует удача, так как привилегированный процесс сам по себе умеет работать в условиях тюрьмы с измененным корнем. Такой возможностью обладает, например, демон ftp, работающий в режиме анонимного доступа. Если клиент подключается к ftp как пользователь ftp или anonymous, демон начинает работать от имени пользователя ftp. Web-сервер Apache обычно работает от имени пользователя nobody, и его деятельность ограничена корневым каталогом иерархии, в которой хранятся web-документы, таким путем он также формирует нечто вроде тюрьмы *с* измененным корнем. Однако сервер Apache может быть уязвимым для внешних атак — все зависит от его конфигурации.

ССЫЛКА

Врезка типа «Ссылка» указывает на другие места книги, в которых излагается материал, связанный с рассматриваемым.

Создание тюрьмы с измененным корнем будет проиллюстрировано на примере использования привилегированного сервера DNS. К счастью, начиная с версии BIND 8.1.2 этот демон умеет работать в границах тюрьмы с измененным корнем. К сожалению, до определенного момента он не обладал такой

возможностью. Именно тогда некоторые системы подверглись успешным атакам, использующим некоторые ошибки, обнаруженные в бинарном файле named.

Здесь я не буду подробно описывать характер уязвимых мест BIND, скажу лишь, что версия BIND, входящая в состав OpenLinux 2.3, содержит в себе эти ошибки. Если сервер содержит в себе такие ошибки, вы можете обнаружить внутри файла /var/named строку AMDROCKS. В этом случае следует получить обновленную версию пакета, которую можно обнаружить по адресу ftp://ftp.calderasystems.com/pub/openLinux/updates/2.3/current/RPMS/. Три пакета, которые вас интересуют, называются bind-8.2.2p4-l.i386.rpm, bind-doc-8.2.2p4-l.i386.rpm и bind-utils-8.2.2p4-l. i'386.rpm. Если доступна более свежая версия сервера, вы должны использовать ее.

Номера версий пакетов RPM

Чтобы проверить, обладаете ли вы самыми последними версиями пакетов, воспользуйтесь следующей командой:

rpm -qa | grep bind

Эта команда подскажет вам, какие из установленных в системе бинарных пакетов содержат в себе bind. Сравните вывод этой команды с файловыми именами пакетов, размещенных на web-узле компании Caldera Systems (если вы используете комплект Linux другого поставщика, обратитесь к web-узлу этого поставщика). Имя пакета RPM состоит из имени пакета, номера версии программного обеспечения, номера ревизии пакета и идентификатора архитектуры:

<имя_пакета>-<нонер.версии.програмны>-<нонер_версии_пакета_rpm>.<архитектура>.rpm

Например, взгляните на файловое имя пакета BIND: bind-8.2.2p4-l.i386.rpm. Можно заметить, что в данном случае:

Имя пакета: bind
Версия программы: 8.2.2p4
Номер версии пакета RPM: 1
Архитектура: i386

Архитектура может обозначаться одной из следующих меток: i386, spare, alpha, arm, m68k, mips, ppc, s390, sparc64 и noarch. Последняя метка, noarch, обозначает, что пакет является архитектурно-независимым. В таких пакетах могут содержатся, например, программы Perl или сценарии оболочки. Вместо архитектуры может быть также использоваться метка src.

Если в конце имени RPM-пакета стоит сочетание src.rpm, это означает, что в пакете содержится исходный код. Такой пакет можно установить в системе, однако в дальнейшем его необходимо преобразовать в бинарный пакет RPM. В системе Red Hat вы можете установить rpm, затем от лица root выполнить команду cd /usr/src/redhat/SPECS (в системе Caldera OpenLinux путь может быть /usr/src/OpenLinux/SPECS — в других комплектах поставки путь может быть другим). Находясь в данном каталоге, выполните команду:

rpm -bb <имя_програнны>.spec
или rpm -ba <имя_программы>.spec

Подставьте соответствующее имя программы. Первая команда формирует бинарный RPM-файл. Вторая команда создает как бинарный RPM, так и новый файл src.rpm. Вторая команда используется тогда, когда вы вносите изменения в некоторый spec-файл и в связи с этим желаете создать свежий файл src.rpm. Бинарные rpm-файлы размещаются в подкаталогах ../RPMS/i386 или ../RPMS/noarch/. Многие файлы src.rpm преобразуются в несколько бинарных rpm-файлов. Пакет src.rpm, содержащий в себе BIND, преобразуется в три бинарных rpm-фай-ла, о которых говорится в тексте. Файлы src.rpm размещаются в подкаталоге ../SRPMS.

BIND обозначает Berkley Internet Nameserver Daemon. Главный исполняемый файл, то есть собственно демон сервера имен, называется named (от английского *nameserver daemon* — демон сервера имен). На примере BIND я продемонстрирую вам процесс создания тюрьмы с измененным корнем (change root jail). Описанный подход вы сможете применить в отношении какой-либо другой службы. Чтобы получить представление о том, как именно должна выглядеть эта тюрьма, взгляните на домашний каталог ftp с анонимным доступом (/home/ftp).

Прежде чем начать, следует выполнить небольшое предварительное планирование. Для начала необходимо подобрать файловую систему с необходимым количеством свободного пространства. Для примера предположим, что каталог /home соответствует отдельному дисковому разделу, на котором достаточно свободного места. Если в вашей системе это не так, вы можете использовать либо /usr/local, либо /usr. Во-вторых, вы должны придумать имя пользователя, от лица которого будет запущен соответствующий демон. Имя пользователя рекомендуется сделать связанным с именем службы. Также следует решить, какие идентификаторы UID/GID будут использоваться для запуска демона. В операционной среде Caldera OpenLinux идентификаторы с номерами от 0 до 499 используются системой, а идентификаторы с номерами выше 500 используются для обычных пользователей. Если взглянуть в файл /etc/passwd, можно обнаружить, что номера всех используемых системных идентификаторов находятся в диапазоне от О до 100. Стало быть, идентификаторы с номерами от 101 до 499 вполне можно использовать для таких целей, как размещение службы в тюрьме с измененным корнем. Таким образом, вы можете

зарезервировать номера от 200 до 299, что будет вполне достаточно.

Далее, используя любой удобный для вас инструмент администрирования, вы должны выполнить следующие действия.

1. Зарегистрируйте UID/GID 200 для пользователя dns с домашним каталогом /home/dns:

groupadd -g 200 dns useradd -g dns -u 200 -m dns

Для регистрации достаточно выполнить эти две команды. Убедитесь в том, что учетная запись заблокирована, так как ни один реальный пользователь не должен обладать возможностью подключиться к системе с использованием учетной записи dns. Чтобы заблокировать учетную запись, можно воспользоваться COAS (System Administration/Account Administration). Команда useradd блокирует учетную запись в момент ее создания. Если вы не желаете создавать подкаталог dns в каталоге /home, не добавляйте в команду ключ -т. Вместо этого следует указать необходимый путь в файле /etc/passwd (например /usr/dns или любой другой удобный для вас), а затем создать соответствующий каталог вручную (убедитесь в том, что этот каталог принадлежит пользователю dns).

2. Теперь переместитесь в каталог /home/dns. Вы обнаружите, что в данном ката логе содержатся некоторые файлы, большинство из них являются скрытыми (то есть с именами, начинающимися с символа точки). Все эти файлы следует удалить. После того как вы сделаете это, начинается реальная работа по созда нию тюрьмы с измененным корнем.

3. В процессе создания тюрьмы с измененным корнем вы должны разместить в создаваемой вами иерархии каталогов все файлы, необходимые для запуска размещаемых в этой тюрьме программ. К таким файлам относятся библиоте ки, устройства, исполняемые файлы, а также конфигурационные файлы и фай лы данных. Все эти файлы необходимо разместить относительно измененного корня приблизительно так же, как они размещаются относительно основного корневого каталога основной файловой системы. Таким образом, начать сле дует с создания базовой структуры необходимых вам каталогов:

mkdir dev ; mkdir etc ; mkdir usr ; mkdir usr/sbin ; mkdir var ; mkdir van/named ; mkdir var/run ; mkdir var/lock : mkdir var/lock/subsys ; mkdir lib

4. Теперь необходимо заполнить эти подкаталоги файлами, необходимыми для запуска named. Можно предположить, что к этим файлам относится бинарный файл named, а также все имеющие к нему отношение конфигурационные файлы и файлы данных. Таким образом, необходимо скопировать следующие файлы:

/usr/sbin/named /usr/sbin/named-xfer /etc/named.conf /var/named/*

ПРИМЕЧАНИЕ

Если вы используете поставку, отличающуюся от Caldera OpenLinux, перечисленные файлы могут размещаться в других местах файловой системы.

Перечисленные файлы необходимо скопировать в соответствующие подкаталоги каталога /home/dns (например, файл /usr/sbin/named копируется в /home/dns/ usr/sbin/named).

5. После этого перейдите в каталог /usr/sbin (cd /usr/sbin) и выполните коман ду ldd named. По этой команде на экране появится перечень библиотек, ко торые необходимы для запуска named. Вывод команды ldd named показан в лис тинге 14.1.

Листинг 14.1. Результаты выполнения команды Idd named libc.so.6 => /lib/libc.so.6 (0x4001a000) /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)

Эти библиотеки используются в процессе функционирования named. Эти же библиотеки необходимы также для функционирования named-xfer. Поэтому их нужно скопировать в соответствующие места иерархии /home/dns.

6. После этого скопируйте /etc/named.conf и /var/named/* в соответствующие каталоги в иерархии /home/dns. Взгляните внутрь /etc/named.conf и убедитесь в том, что вы скопировали все необходимые файлы зон, упомянутые в этом файле. Возможно, в каталоге /var/named отсутствуют какие-либо файлы зон, возможно, у вас в системе вообще отсутствует каталог с таким именем, однако именно в этом файле рекомендуется создавать новые файлы зон.

Если вы хотите организовать протоколирование сведений о работе службы в журналах (неплохая идея), вы должны скопировать также файл часового пояса, размещенный в etc. Если ваша система настроена должным образом, файл /etc/ localtime является ссылкой, указывающей на /usr/share/zoneinfo/.../<Ваш_часовой _пояс>. Однако если вы просто скопируете файл /etc/localtime в каталог /home/ dns/etc, скопирован будет файл, на который указывает ссылка, стало быть, все будет в порядке: cp /etc/localtime /home/dns/etc/localtime

После этого перейдите в /home/dns/dev и создайте устройство null:

```
cd /home/dns/dev
mknod -m 0666 null c 1 3
```

Вам потребуется также urandom: mknod -m 0644 urandom c 1 9

Теперь создание базовой структуры завершено. Однако прежде чем запускать named, необходимо выполнить еще несколько немаловажных действий. Прежде всего, необходимо должным образом настроить syslog. По умолчанию syslog следит только за /dev/log. Чтобы приказать syslog следить за другими местами системы, необходимо добавить параметр, указывающий, за чем именно должен следить syslog. В системе OpenLinux данный параметр добавляется в файл /etc/sysconfig/ daemons/syslog. Для этого в строку OPTIONS_SYSLOG необходимо добавить запись -a /home/dns/dev/log. После этого необходимо остановить, а затем заново запустить syslog, в результате будет осуществляться слежение еще за одним устройством документирования, которое создается демоном named в подкаталоге /home/dns/ dev/. После того как вы перезапустите syslog, вы увидите новый сокет с именем log в подкаталоге /home/dns/dev. Если этого не произойдет, вновь остановите и заново запустите syslog. Иногда при перезагрузке syslog не может правильно прочитать параметры.

Перейдите в каталог /etc/sysconfig/daemons и отредактируйте файл named таким образом, чтобы строка OPTIONS выглядела следующим образом: OPTIONS=-t /home/dns/ -u dns -g dns.

СОВЕТ

Если у вас возникли проблемы и при работе демона, запущенного в тюрьме с измененным корнем, возникают сбои, запустите в его отношении strace. Взгляните на имена нескольких последних процедур, которые этот демон пытается запустить. Возможно, вы сможете обнаружить решение проблемы.

Теперь все, что осталось сделать, — это отредактировать сценарии, которые контролируют запуск и завершение работы named. Необходимо модифицировать два сценария: /etc/re.d/init.d/named и /usr/sbin/ndc. На всякий случай скопируйте инициализационный сценарий под именем named.org, на случай, если вам потребуется изначальное содержимое этого сценария. Измените строку DAEMON таким образом, чтобы вместо /usr/sbin в ней значилось home/dns/usr/sbin. После этого найдите любые ссылки на /var/<что-либо> и добавьте в начало каждой из них имя вашего каталога /home/dns. После этого найдите раздел, начинающийся со строки # Start daemons (запуск демонов). Строку, расположенную сразу же после этой строки, необходимо изменить таким образом, чтобы вместо start-stop-daemon -S -n $NAME -x $DAEMON в ней стояло start-stop-daemon -S -n $NAME -x $DAEMON -- $OPTIONS (иными словами, в нее необходимо добавить - $OPTIONS).

В сценарий ndc следует внести несколько более сложные изменения. Выполните следующие действия.

- На всякий случай сделайте резервную копию файла ndc (назовите ее, напри мер, ndc.orig).
- В начало выражения PATH добавьте строку /home/dns/usr/sbin.
- После этого измените выражение PIDFILE так, чтобы оно указывало на /home/ dns/var/run/named.pid. Потом найдите все упоминания /usr/sbin/named (таких строк должно быть две) и замените их на:

```
/home/dns/usr/sbin/named -t /home/dns -u dns -g dns  ...
```

ПРИМЕЧАНИЕ

Перед тем как редактировать ndc, следует выполнить проверку. Более свежие версии вместо сценария оболочки используют откомпилированный бинарный файл. Если в результате выполнения сценария /usr/sbin/ndc на экране появляется сообщение ELF 32-bit LSB executable, Intel 80386, version 1, dynamically linked (uses snared libs) not stripped или нечто подобное, пропустите действия, связанные с модификацией ndc, — вы не сможете его использовать.

Сейчас вы готовы запустить демон. Чтобы запустить или остановить named, можно использовать либо ndc, либо /etc/rc.d/init.d/named. На самом деле будет неплохо, если вы проверите корректность работы каждого из этих сценариев. Если вы взглянете в таблицу процессов, вы должны увидеть, что демон named работает от лица пользователя dns, а при его запуске использовался аргумент -t /home/dns.

Теперь необходимо проверить сформированную вами конфигурацию. Для этого вы можете использовать либо nslookup, либо dig, либо dnsquery. Если разрешение имен узлов осуществляется корректно, можете считать, что вы завершили процесс запуска сервера DNS в тюрьме с измененным

корнем.

Аналогичную процедуру можно использовать также в отношении других служб, однако перед запуском сервера потребуется выполнить команду chroot (то есть *change root — изменить корень)*. Конечно же, перед этим вы должны выполнить все описанные действия, сформировав необходимую структуру каталогов и разместив в ней все файлы, необходимые для запуска службы в рабочей среде с измененным корнем.

Другие меры

Запуск служб в рабочей среде с измененным корнем — это лишь один из способов, которыми вы можете воспользоваться для того, чтобы защитить вашу систему. В качестве дополнительных средств защиты можно использовать IP Chains (ядро версии 2.2.x) или NetFilter (ядро версии 2.4.x) для наблюдения за портами и/или блокирования портов. Вы также можете использовать tripwire для того, чтобы связать службу с ловушкой. В остальных частях книги мы рассмотрим программные средства, которые можно использовать для защиты.

Однако вы не должны забывать о том, что лучшая защита системы — это ваша собственная бдительность. Конечно, вы не обязаны круглосуточно сидеть рядом с системой, подкарауливая вредителей, вполне достаточно уделить контролю безопасности хотя бы пару минут в день. В последующих главах я подробнее расскажу об этом.

Как уже говорилось ранее, помимо прочего для контроля безопасности вы можете воспользоваться командой lsattr -v. Очевидно, этот метод не является на 100% надежным. Говоря точнее, если версия inod файла изменилась с 3302778664 на 1, вы можете быть уверенными в том, что файл модифицирован. Однако если номер версии не изменился, вы не можете сказать с полной уверенностью, осталось ли содержимое файла неизменным или файл был модифицирован. Чтобы проверить это, вы должны воспользоваться каким-либо иным способом.

Причина подобной неуверенности лежит в том, каким образом работают команды копирования (cp) и перемещения (mv) файлов. Если вас это интересует, я расскажу об этом подробнее.

В Linux программы копирования и перемещения файлов работают по-разному. Их работа также зависит от того, размещаются два файла на одном и том же разделе или на разных. Сначала рассмотрим ситуацию, когда два разных файла размещены на одном и том же разделе.

- Команда cp копирует данные из блока данных одного файла в блок данных другого файла и обновляет atime и mtime (время доступа и время модификации).

- Команда mv изменяет указатель каталога и перенаправляет имя файла от одного inode на другой inode. После этого имя файла указывает на другой mode. Значения atime и mtime не обновляются. Если же файлы размещаются на двух разных разделах:

- Команда cp выполняется так же, как и в первом случае.

- Команда mv не только выполняет копирование блоков данных, но и меняет некоторую информацию inode (только не всю). Номер версии inode не меняется, однако меняются значения atime и mtime. Таким образом, в этом случае указатель каталога не изменяется, inode используется повторно, благодаря чему номер версии inode остается прежним.

Исходя из только что сказанного можно сделать вывод, что изменение номера версии inode осуществляется только в одном из четырех рассмотренных случаев: в ситуации, когда выполняется команда mv и два файла размещаются на одном и том же разделе.

Восстановление после атаки

Если вы обнаружили, что кто-либо обладает доступом к вашей системе на уровне привилегий root, вы можете действовать в соответствии с одним из трех вариантов. Первый вариант предусматривает полную переустановку всей системы и всех программ — сделайте резервную копию рабочих данных, отформатируйте каждый из разделов и начните формирование системы с нуля. Это гарантированный и самый лучший метод (самый простой, дающий полную уверенность в решении проблемы и самый практичный), однако данный вариант далеко не всегда является приемлемым. Если вы отформатируете все разделы и выполните установку заново, вы уничтожите все следы взлома. При этом вы также потеряете все конфигурационные файлы, данные и т. п. Иногда подобное недопустимо.

ПРИМЕЧАНИЕ

Если вы видите некоторую запись в журнале, которая выглядит как искаженная передача данных, попробуйте

найти строку /bin/sh. Зачастую эта команда располагается в глубине строки, окруженная множеством специальных символов (каждый из которых выглядит как буква верхнего регистра, перед которым стоит символ ^). Если вы обнаружите подобную запись, имейте в виду, что это не случайный набор символов, а преднамеренная попытка переполнения буфера.

Второй вариант — ничего не делать. На самом деле это не вариант. Если в системе есть хотя бы что-либо ценное для вас, наверняка вы не можете себе позволить, чтобы совершенно посторонние люди делали все что им заблагорассудится с ценными для вас данными. Если в системе нет ничего ценного и у вас есть свежие резервные копии всех конфигурационных файлов и файлов данных, значит, переустановка системы — это самый лучший для вас выход.

Третий вариант требует значительного времени. Попытайтесь идентифицировать любые измененные файлы и замените их файлами, про которые вы знаете, что они корректны. Сразу же после того как вы восстановите защиту системы, следует попытаться определить, каким образом взломщик проник в систему. Если вы не сможете определить, каким образом злоумышленник реализовал свой замысел, возможно, после запуска системы вы вновь обнаружите его внутри системы. Таким образом, прежде чем предпринимать какие-либо другие действия, следует скопировать конфигурационные файлы на другую систему или на ленту.

ССЫЛКА

О конфигурации syslog и чтении файлов syslog рассказывается в главах 22 и 23 соответственно.

Корректность каких файлов следует проверить? Лучше всего осуществлять проверку последовательно — каталог за каталогом. Прежде чем начать, убедитесь в том, что компьютер не связан с сетью. Для этого достаточно просто отсоединить сетевой кабель от сетевой карты. Теперь вы готовы начать.

Прежде всего следует убедиться в том, что никаких изменений не внесено в любой из подкаталогов etc/. В вашей системе может быть несколько таких подкаталогов. Зачастую программы, устанавливаемые в системе, создают конфигурационные файлы в подкаталогах /usr/local/etc, var/etc, /use/etc и проч. В подкаталогах /opt также могут находиться подкаталоги etc/. Однако корневой каталог /etc является наиболее важным.

В главном каталоге /etc содержатся файлы passwd, shadow, group и (возможно) gshadow. Необходимо убедиться в том, что ни один из этих файлов не модифицирован. Фактически невозможно (а на самом деле и бессмысленно) проверить, завладел ли взломщик копией файла shadow или нет. Поэтому всегда следует считать, что этот файл уже известен взломщику. Исходя из этого следует сменить пароль пользователя root.

Изучая содержимое упомянутых файлов, следует обращать внимание на пользователей, обладающих UID 0 и не являющихся пользователем root, появление пустых паролей (паролей null) для любой из учетных записей, разблокирование ранее заблокированных учетных записей, пользователей, добавленных в любую из системных групп, и т. п. Подобные проверки можно выполнить с использованием grep (если файлы очень крупные, например, как в университетских компьютерных сетях) или вручную (для небольших офисных и домашних сетей).

В каталоге /etc располагается также подкаталог pam.d. Расположенные в нем файлы определяют работу служб с ограниченным доступом, и если изменить любой из них, можно открыть доступ к соответствующей службе через вход, который ранее был закрытым. Также можно просто отключить некоторые из проверок. Более подробно о содержимом этих файлов рассказывается в главе 1. Однако следует помнить, что некоторые файлы каталога /etc/pam.d указывают на другие файлы, расположенные в каталоге /etc, поэтому они могут быть также модифицированы.

В каталоге /etc также содержится подкаталог rc.d и файл inittab. Все программы, упоминаемые в файле inittab и подкаталоге rc.d, запускаются от имени пользователя root, и злоумышленник может воспользоваться этим. Например, он может добавить строку, которая будет от имени пользователя root запускать командную оболочку, и связывать ее с любым удобным для него портом. Так как оболочка будет запущена от имени пользователя root, ее вовсе не обязательно связывать с портом ниже 1024.

В каталоге /etc/X11 содержится набор подкаталогов, содержащих файлы, запускаемые от имени пользователя root. Поэтому данный каталог тоже может заинтересовать взломщика, впрочем, как и каталог /etc/cron.d. На самом деле весь каталог /etc (и его подкаталоги) является идеальным местом для взломщика, желающего добавить в какой-нибудь файл строку, запускающую какую-либо службу от имени root.

Проверяя, модифицировал ли взломщик какие-либо файлы, следует уделить внимание подкаталогам sbin/ и bin/. Такие подкаталоги размещаются в каждом из крупных каталогов, начиная с корневого каталога файловой системы (в этих подкаталогах размещаются наиболее важные исполняемые файлы) и заканчивая /usr, /usr/local и даже в подкаталогах каталога /opt.

Наиболее часто модификациям подвергается исполняемый файл /bin/login. Для любой системы этот

исполняемый файл является воротами, которые ведут внутрь системы. Без всякого сомнения, этот исполняемый файл является наиболее уязвимым местом системы. Даже не пытайтесь проверить его неизменность — просто замените его на заведомо корректную версию. Корректная (изначальная) версия этого файла содержится в пакете util-linux-x.xx-x.i386.rpm. Всегда используйте самую последнюю версию, кроме того, сделайте соответствующий более свежий пакет RPM пакетом по умолчанию. Система администрирования пакетов RPM (kpackage) позволяет *форсировать* (force) использование более свежих версий пакетов вместо пакетов, которые уже установлены в системе. Благодаря этому вы сможете заменить более свежими версиями значительное количество исполняемых файлов, расположенных в подкаталогах /bin и /usr/bin.

Далее следует уделить внимание каталогам lib/. В подкаталоге /lib корневого каталога системы располагается подкаталог modules, в котором размещаются модули ядра, добавляемые пользователем root. В этих подкаталогах также содержится весь код общего доступа, то есть код, используемый разнообразными утилитами и программами. Если в вашей системе произошла замена службы login, новая версия login могла быть сформирована одним из трех способов: сборка могла быть выполнена на вашей системе с использованием ваших библиотек (в этом случае новая версия абсолютно точно будет работать в вашей рабочей среде); сборка могла быть выполнена на аналогичной системе с использованием библиотек, аналогичных вашим (таким образом, новая версия опять же должна работать); сборка могла быть выполнена с использованием произвольной библиотеки, в этом случае в вашей системе должны быть установлены как бинарный файл login, так и используемая им библиотека.

Наконец, следует выполнить проверку подкаталогов tmp/. Здесь следует искать не столько модифицированные файлы, сколько файлы неизвестного назначения. Если вы сомневаетесь относительно некоторого файла, лучше всего просто уничтожьте его. Временные каталоги часто используются в качестве места хранения комплектов доступа к системе на уровне root (root kit). Иными словами, злоумышленник размещает во временном каталоге файлы, необходимые для несанкционированного доступа в систему. Если во временном каталоге обнаруживаются файлы, к которым никто не обращался на протяжении всей минувшей недели, скорее всего, такие файлы следует удалить.

ПРИМЕЧАНИЕ

Входящая в состав комплекта Caldera OpenLinux 2.3 программа cleandir упрощает процесс очистки каталога tmp.

Не следует доверять любой базе данных, которая может храниться в вашей системе и указывать на то, что все в порядке. Это относится также и к базе данных RPM. Однако установленную в системе базу данных можно сравнить с заведомо корректной базой данных, доступ к которой из вашей системы невозможен.

Хороший метод проверки показан в листинге 14.2.

Листинг 14.2. Процедура проверки установленных в системе пакетов RPM #!/bin/bash

```
PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH
VFILE=/root/rpm.verify.'date +%d%m%y'

for i in 'rpm -qa'
do
rpm -V $i >> $VFILE
done
```

Сценарий, подобный этому, можно ежедневно запускать при помощи cron. Полученный таким образом файл, содержащий сведения о корректности установленных пакетов, можно сравнивать с аналогичным эталонным файлом, который хранится вне системы. Для этого ежедневно формируемый файл следует копировать на дискету и сравнивать с эталоном при помощи утилиты diff. Обнаруженные различия будут указывать на возможную модификацию файлов.

ПРИМЕЧАНИЕ

Среди файлов dailyscript есть программа под названием check-packages, которая выполняет проверку установленных в системе пакетов, а затем сравнивает их с результатами, полученными за день до этого.

Сценарий просматривает каждый из RPM-файлов, упоминаемых в базе данных пакетов RPM, установленных в системе, а затем выполняет сравнение так, как это показано в листинге 14.3.

Листинг 14.3. Сокращенный вывод команды rpm -V, выполненной в отношении всех установленных в системе пакетов

```
.M...UG.        /dev/vcsa1
.M....G.        /dev/vcsa2
....L...        /dev/video
S.5....T    c  /etc/printcap
.......T    c   /etc/sysconfig/daemons/bigfs
SM5....T  c  /usr/XllR6/lib/Xll/app-defaults/Bitmap
missing       /usr/src/linux-2.2.10/vmlinux
```

В этом листинге латинские буквы обозначают один из тестов, в результате которого обнаружено несоответствие. Каждая точка обозначает, что тест пройден.

Листинг 14.4. Обозначения, используемые в листинге 14.3

```
S   размер файла
M  режим доступа (включая разрешения и тип файла)
5   MD5 сумма
D   устройство (Device)
L   символическая ссылка (Symlink)
U   пользователь (User)
G   группа (Group)
T   время модификации (Mtime)
c   обозначает конфигурационный файл
```

Обратите внимание, что в результирующий файл вносятся только записи об обнаруженных несоответствиях. Если для некоторого файла все тесты выполнены, соответствующая запись не вносится в результирующий файл. Например, если вы проверяете пакет util-linux и в ходе проверки не обнаружено ни одного несоответствия (все восемь тестов пройдены), запись об этом не будет добавлена в результирующий файл проверки.

Также следует убедиться в том, что ни в одном из пользовательских домашних подкаталогов не содержится программ, которые способны работать в режиме SUID root или SGID root. Поиск подобных программ обсуждался в начале книги.

Подготовка к неизбежному

Самый лучший способ подготовки к нападению на вашу систему очень прост, и вы о нем знаете: это резервное копирование. Вовсе не обязательно делать полную резервную копию всей системы. На самом деле для того, чтобы в случае необходимости полностью восстановить работоспособность системы, достаточно сделать архив разнообразных подкаталогов etc/, а также некоторых подразделов каталога /var.

Конечно, вам потребуется сделать резервное копирование рабочих данных (в частности, каталога /home, который, как правило, соответствует отдельному разделу системы). Для хранения резервной копии рабочих данных может потребоваться значительное место, однако в остальном резервная копия конфигурации системы может быть очень небольшой по размеру. Благодаря этому для того, чтобы быстро восстановить работоспособность, вам потребуется заново установить систему, а затем скопировать в нее конфигурационные файлы из резервной копии.

Для реализации подобного подхода следует определить те небольшие участки системы, которые нуждаются в резервном копировании. Например, на многих серверах в состав резервной копии системы достаточно включить содержимое подкаталогов /etc, /var/named (или /home/dns в случае, если вы запускаете DNS-сервер в рабочей среде с измененным корнем), а также /home/httpd и копию содержимого базы данных.

За исключением базы данных все остальные перечисленные данные являются относительно статическими. Страницы сервера Web, как правило, изменяются чаще, чем все остальное. В частности, система может работать многие месяцы, в течение которых вы можете вообще не заглядывать в подкаталоги /etc или /var/ named.

СОВЕТ

Резервное копирование является ключевым механизмом восстановления работоспособности. Хорошая резервная копия позволит вам восстановить работоспособность системы всего за несколько минут. Если у вас нет возможности регулярно создавать резервные копии на магнитных лентах, как можно чаще копируйте наиболее важные файлы на носитель информации, не входящий в состав системы. Если для этой цели вы используете гибкие диски, храните две копии каждого из них (на случай, если на одном из них возникнут дефектные секторы).

Таким образом, прежде чем подключиться к сетевому кабелю Ethernet, найдите время и выполните следующие команды:

```
cd / ; tar czvf etc.tar.gz /etc
tar czvf misc.tar.gz /var/named /home/httpd
```

Благодаря этому вы получите резервную копию большей части вашей изначальной конфигурации. Кроме того, каждый из этих файлов умещается на одном гибком диске. На многих системах вы можете записать на одну дискету оба этих файла. Скопируйте каждый из этих файлов на две дискеты, пометьте их и спрячьте на черный день.

Заведите привычку регулярно создавать резервные копии важных файлов данных (к ним относятся файлы, расположенные в каталоге $HOME, а также любые файлы баз данных, размещенные где-либо в системе). «Регулярно» может означать раз в неделю, раз в день или еще более часто, в зависимости от того, какой объем рабочих данных хранится у вас в системе. Важные системные файлы можно архивировать сразу же после того, как вы вносите в них изменения. Можно хранить не одну, а несколько последних резервных копий. Если вы выполняете резервное копирование один раз в неделю, возможно, для вас будет лучше хранить четыре последних резервных копии (за весь минувший месяц), а также, возможно, две копии двух предыдущих месяцев дополнительно. Старая резервная копия всегда лучше, чем ничего.

Теперь вы окончательно подготовились к запуску. Подключайте сетевой кабель, и пусть взломщики приступают к своим грязным делам.

Заключение

В данной главе рассмотрены разнообразные вопросы, связанные с защитой вашей системы. Вы узнали, что основной целью любого злоумышленника является доступ к системе на уровне привилегий root. Вы также узнали, что взломщики могут достичь этой цели несколькими разными способами, однако в основном это осуществляется с использованием предлагаемых вами служб. Вы также поняли, что некоторые службы настолько сложны и мощны, что их полную защиту фактически невозможно гарантировать.

Я рассказал вам о некоторых способах снизить вероятность проникновения в систему и ограничить объем возможных повреждений. В частности, я рассказал о том, как можно запустить некоторую службу в рабочей среде с измененным корнем. Рассмотренный в данной главе пример использует для этого специальную возможность, встроенную в свежие версии DNS-сервера. Однако помимо DNS подобным образом можно настроить и другие службы. Запуск в рабочей среде с измененным корнем необходим для тех программ, которые не могут работать на уровне привилегий ниже, чем root. Но таких программ становится все меньше, так как если программу (или, по крайней мере, программный поток, обслуживающий внешние подключения) можно запустить от лица непривилегированного пользователя, она обладает рядом преимуществ.

Однако если злоумышленнику все же удалось проникнуть в систему, вы должны предпринять комплекс мер для того, чтобы восстановить защиту и дальнейшее корректное функционирование системы. Я рассказал вам о некоторых действиях, которые следует выполнить в подобной ситуации, например, о проверке номеров версий inod (chattr -v/lsattr -v) и пакетов RPM (rpm -V). Я также рассказал вам о том, за какими службами следует присматривать особенно.

Наконец, в последнем разделе главы я рассказал о резервном копировании.

15 Использование оболочек TCP (TCP Wrappers)

В данной главе рассматриваются следующие вопросы:
- что такое оболочка TCP (TCP Wrappers);
- как работает оболочка TCP;
- настройка оболочки TCP;
- тестирование конфигурации оболочки TCP;
- использование tcpdchk;
- использование tcpdmatch.

В данной главе речь пойдет о программах из категории TCP Wrappers (оболочка TCP). Оболочка TCP — это программное средство, предназначенное для обнаружения и предотвращения атак через сеть. Я подробно расскажу о том, что такое оболочка TCP, как она работает и как осуществляется ее настройка. Чтобы хорошо усвоить этот материал, вам потребуются знание файла /etc/inetd.conf — какие данные в нем содержатся и в каком формате. В данной главе коротко рассматривается файл inetd.conf, а также файлы /etc/hosts, allow и /etc/hosts, deny.

Оболочка TCP (TCP Wrappers) — это программа, специально предназначенная для обнаружения и предотвращения попыток взлома системы. Как правило, это программное средство работает совместно с inetd, однако многие другие серверные программы, которые не запускаются с использованием inetd, могут быть скомпилированы с использованием библиотеки TCP Wrappers, благодаря чему они станут поддерживать эту функциональность. Чтобы понять, как настраивается TCP Wrappers, необходимо овладеть приемами конфигурации файлов /etc/ hosts.allow и /etc/hosts.deny.

ССЫЛКА
Подробнее о файле inetd.conf рассказывается в главе 11.

Что такое TCP Wrappers?

В переводе с английского *wrapper* означает *«оболочка»* — то есть нечто, что располагается вокруг чего-либо для того, чтобы защитить его от внешнего влияния. TCP Wrappers — это программа, которая защищает модуль ожидания поступления запросов через tcp (tcp listener) для некоторого серверного приложения. Оболочка TCP позволяет предотвратить взлом сетевого узла через соответствующую службу, также ее можно использовать для протоколирования взаимодействия клиента и сервера в журнале. TCP Wrappers не является абсолютной защитой, однако с ее помощью можно обнаружить попытку взлома и сохранить в журнале сведения об источнике подобных попыток. Благодаря этому вы сможете сделать так, чтобы эти попытки были безуспешными. Однако сама по себе программа TCP Wrappers не делает абсолютно всего. Для активной борьбы с последующими атаками следует использовать другие программы, например IP Chains или Netfilter.

ССЫЛКА
Подробнее об IP Chains и netfilter рассказывается в главе 16.

Говоря о TCP Wrappers, легче объяснить, чем не является это средство, чем дать определение, что же это такое. TCP Wrappers не является программой, с помощью которой можно наблюдать за любым серверным процессом. Эффективно использовать TCP Wrappers можно только в отношении серверных приложений, основанных на TCP и запускаемых с использованием inetd (интернет-суперсервер) или

откомпилированных с использованием библиотеки TCP Wrappers. Однако далеко не все серверные приложения могут эффективно использовать TCP Wrappers. В частности, команды «r», такие как rlogin, rsh, rexec, плохо подходят для совместного использования с TCP Wrappers, а службы, основанные на UDP, вообще не могут использовать TCP Wrappers.

ПРИМЕЧАНИЕ

Не существует универсального способа обеспечения внешней и внутренней безопасности. Чтобы защитить систему, приходится использовать разнообразные средства в комплексе. Однако ваша собственная бдительность является лучшим инструментом, который ничем не заменить.

Как работает TCP Wrappers

Программное средство TCP Wrappers можно использовать несколькими разными способами в зависимости от конкретной ситуации. Во-первых, вы можете заменить бинарный файл программы файлом tcpd (исполняемым файлом программы TCP Wrappers). Однако подобная замена не является достаточно гибким методом. В результате любого обновления изначального программного пакета вам придется заново настраивать в его отношении TCP Wrappers. Так как этот метод не является гибким и достаточно удобным, мы не будем его обсуждать.

Второй метод предусматривает запуск tcpd при помощи inetd. Демон inetd запускает tcpd вместо изначальной программы. Для этого необходимо должным поэтому обратное сопоставление имнени не сработает. Не сработает оно также и в случае, если первичный и вторичный серверы DNS недоступны. Таким образом, если только у вас нет серьезных причин, по которым вы обязаны использовать режим PARANOID ,будет лучше, если вы отключите этот режим и будете проверять подобные подозрительные соединения каким-либо иным способом.

Еще один связанный с этим механизм имеет отношение к маршрутизации источника (source routing). Чтобы предотвратить подделку адреса-источника (такая подделка обозначается английским термином spoofing), программа TCP Wrappers может быть скомпилирована в режиме игнорирования предоставляемых клиентом сведений о маршрутизации источника. Этот режим неэффективен в отношении служб UDP. Если вы используете этот метод в отношении служб UDP, у вас может появиться ложное чувство безопасности.

ПРИМЕЧАНИЕ

Маршрутизация источника (source routing) — это метод передачи приемнику информации о маршруте, позволяющем передать данные обратно к источнику. При этом источник самостоятельно передает эту информацию приемнику, а приемник использует эти сведения вместо того, чтобы маршрутизировать пакеты своими силами. Похоже на то, как если бы, отправляя письмо своему другу, вы приложили бы к нему не только свой обратный адрес, но также и карту, сообщающую, как надо идти, чтобы прийти к вам домой. Обратный маршрут может оказаться неправильным, и тогда, если следовать этому маршруту, можно оказаться в совершенно другом месте.

Еще один режим компиляции tcpd предусматривает подключение в соответствии с требованиями RFC 931. Этот RFC описывает взаимодействие с демоном идентификации identd (ident daemon). Если вы компилируете tcpd в этом режиме, tcpd пытается соединиться с демоном идентификации, работающим на удаленной системе. Однако в настоящее время в Интернете работает огромное количество клиентов Microsoft Windows, а эта операционная система не поддерживает демон идентификации. Для подобных клиентов попытка соединиться с демоном идентификации оканчивается неудачей. При этом демон tcpd не прекращает работу, однако при подключении подобных клиентов возникает задержка длительностью приблизительно 10 секунд (10 секунд — это таймаут по умолчанию для демона идентификации). Так как клиенты Microsoft не могут ответить на запросы, адресованные демону идентификации, на многих системах Unix этот демон также отключен. Кроме того, подобный способ работы не эффективен для служб, основанных на UDP.

Версия tcpd, входящая в комплект OpenLinux, не использует большинства режимов, которые можно добавить в этот демон во время компиляции. Конечно, в будущем ситуация может измениться, однако на текущий момент этот пакет скомпилирован так, чтобы быть полезным для большинства пользователей, то есть без режима PARANOID, без поддержки RFC 931 (identd) и без игнорирования маршрутизации источника. Если вы хотите воспользоваться подобными возможностями, вы можете использовать для этой цели другие механизмы, например netfilter или специальные возможности, добавленные в состав ядер 2.2.x/2.4.x и предназначенные для обнаружения подделки адресов IP и игнорирования маршрутизации источника. Если все же вы желаете добавить соответствующие возможности в tcpd, вы должны заново

откомпилировать tcpd, используя исходные файлы. Если вы используете комплект поставки Linux, основанный на RPM, установите соответствующий пакет с исходным кодом и изучите файл <имя_файла>.spec. В комплект поставки Debian входит аналогичный инструмент, облегчающий компиляцию пакетов и сообщающий вам, как тот или иной пакет настроен для компиляции. Если вы не знаете, каким образом осуществляется компиляция и инспектирование файлов в вашей системе, обратитесь к хорошей книге по администрированию.

ССЫЛКА

Подробнее о netfilter рассказывается в главе 16.

ПРИМЕЧАНИЕ

Исходный код демона tcpd содержится на прилагаемом к книге компакт-диске.

Чтобы документировать сведения о соединениях «клиент:демон» в журнале, программа TCP Wrappers использует механизм syslog. Подробнее о конфигурации TCP Wrappers рассказывается в последующих разделах.

Реализация TCP Wrappers

Чтобы приступить к использованию TCP Wrappers, достаточно выполнить очень простую процедуру. Загляните в файл /etc/inetd.conf и найдите там строку:

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

Замените ее на строку: telnet stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.telnetd

В результате такой замены при поступлении запроса к службе telnet метаде-мон inetd запустит демон tcpd, который запишет в журнал сведения о соединении «клиент:демон» и, в случае если все необходимые для этого условия выполнены, запустит демон telnetd.

ПРИМЕЧАНИЕ

Чтобы оповестить демон inetd о том, что файл /etc/inetd.conf был модифицирован, необходимо отослать этому демону сигнал SIGHUP. Этот сигнал необходимо посылать после каждой модификации файла /etc/inetd.conf. По этому сигналу метадемон inetd заново читает в память и обработает содержимое файла /etc/inetd.conf.

Не удивляйтесь, если увидите, что упомянутая строка в файле /etc/inetd.conf уже содержит в себе tcpd. Во многих комплектах Linux программа TCP Wrappers используется по умолчанию. Однако в этом случае конфигурация tcpd, как правило, является наиболее универсальной. О том, как настроить tcpd в соответствии с вашими собственными предпочтениями, рассказывается далее.

После того как вы внесете в файл /etc/inetd.conf предложенные ранее изменения, суперсервер inetd вместо сервера telnet будет запускать демон tcpd, который будет документировать сведения о любых попытках подключения к порту telnet в журнале. Вы можете настроить таким способом любую службу, упоминаемую в файле /etc/inetd.conf. Для этого в качестве целевой запускаемой программы для данной службы следует указать tcpd, а в качестве аргумента указать имя демона данной службы, как показано ранее. Естественно, данная строка не должна быть закомментирована. Следует также помнить, что tcpd не поддерживает надежного функционирования совместно со службами, основанными на UDP, а также совместно со службами «r» (rlogin, rsh, rexec).

Однако если помимо документирования сведений о попытках подключений вы желаете также либо вообще блокировать доступ к службе, либо разрешить доступ только для определенных клиентов, вы должны использовать для этой цели два файла: /etc/hosts.allow (разрешенные узлы) и /etc/hosts.deny (запрещенные узлы). Вне зависимости от того, разрешены или запрещены соединения при помощи файлов /etc/hosts.allow и /etc/hosts.deny, сведения о любых подключениях и попытках подключения по-прежнему будут документироваться в журнале, благодаря чему вы по-прежнему будете получать информацию об источниках этих попыток, если, конечно, вы не используете исполняемый файл, откомпилированный в режиме PARANOID.

СОВЕТ

Если для блокирования доступа к порту вы используете IP Chains или netfilter (об этом рассказывается в главе 16), программа TCP Wrappers не будет протоколировать сведений о подключениях через этот порт, так как подключение запрещено правилами брандмауэра. Таким образом, если вы хотите установить для некоторого порта ловушку (TCP Wrappers запускает специализированный сценарий оболочки) или желаете при помощи TCP Wrappers отправить обратно клиенту некоторое сообщение, не следует блокировать данный порт с использованием правил брандмауэра.

Изначально для функционирования tcpd требуются оба файла: /etc/hosts.allow и /etc/hosts.deny. Однако при желании вы можете объединить всю необходимую информацию о разрешенных и запрещенных соединениях в одном файле /etc/ hosts.allow и в дальнейшем работать только с одним этим файлом. Такая конфигурация с использованием только одного файла будет рассматриваться позже в данной главе. Чтобы обеспечить работу в таком режиме, необходимо убедиться, что демон tcpd откомпилирован с использованием параметра PROCESS_OPTIONS (в OpenLinux, равно как и в некоторых других комплектах Linux, этот параметр используется по умолчанию). Формат некоторых записей (в особенности команд оболочки) в файле /etc/hosts.allow различается в зависимости от того, был ли использован во время компиляции параметр PROCESS_ OPTIONS, поэтому вы не должны объединять два файла в один, не выполнив предварительной проверки. В противном случае результаты могут оказаться нежелательными.

ПРИМЕЧАНИЕ

Добавляя записи в файлы /etc/hosts.allow и /etc/hosts.deny, помните, что при обнаружении первой подходящей строки поиск подходящих записей прекращается. Таким образом, все остальные подходящие записи не обрабатываются.

Записи файла /etc/hosts.allow обладают следующим форматом:

демон(ы) : клиент(ы) : параметр : параметр ...

Каждая строка — это разделенный двоеточиями список, состоящий из двух или большего количества колонок. Если вы используете только файл /etc/hosts.allow (как подразумевается в данном тексте), первая колонка параметров обязательна, а все остальные могут отсутствовать. В первой колонке указывается демон или демоны, доступ к которым вы желаете обезопасить. Во второй колонке перечисляются клиенты, в отношении которых действует эта запись. В третьей и последующих колонках перечисляются параметры. Это могут быть программы, которые требуется запустить, или действия, которые требуется выполнить. Если в составе какого-либо параметра используется символ двоеточия (например, в выражении PATH), который не является признаком разделения колонок, перед таким символом двоеточия следует поставить символ обратной косой (\). Кроме того, каждая строка, включая последнюю строку файла, должна завершаться символом новой строки. В противном случае строка не будет обработана.

СОВЕТ

Обратная косая (\) используется для того, чтобы интерпретировать специальные символы как обычные символы. Если некоторый символ имеет специальное значение, а вы хотите, чтобы Ьн был обработан как обычный символ, поставьте перед ним символ обратной косой (\ ).

Использование демонов и символьных шаблонов

В первом поле записи файла/etc/hosts.allow содержится список демонов, разделенных пробелами и/или запятыми (можно использовать любой из этих символов). Имя демона должно быть таким, под каким он известен в системе, например, in.telnetd в случае OpenLinux. Если сетевой узел обладает несколькими IP-адресами, для идентификации демона следует использовать формат демон@узел, благодаря чему вы сможете различать различные сетевые карты, установленные на сетевом узле. Вы также можете использовать символьные шаблоны. К ним относятся:

- ALL: универсальное соответствие (все демоны/узлы);

- LOCAL: соответствует узлам, чьи имена не содержат в себе символ точки, например foo или baz;

- UNKNOWN: соответствует любому пользователю, чье имя неизвестно, или любому узлу, чье имя или адрес неизвестны;

- KNOWN: соответствует любому известному пользователю, а также любому узлу, для которого известен как адрес, так и имя;

- PARANOID: соответствует любому узлу, для которого имя не соответствует адресу.

Обратите внимание, что корректное использование шаблонов KNOWN и UNKNOWN связано с DNS. Если система DNS недоступна, эти шаблоны могут работать неправильно. То же самое относится и к PARANOID: если система DNS недоступна, имя сетевого узла не будет соответствовать его адресу и соединение будет разорвано.

ПРИМЕЧАНИЕ

Если DNS недоступна и вы используете UNKNOWN, вы получите соответствие с сетевым узлом, в то время как этого соответствия быть не должно. При этом остальные записи также будут обработаны некорректно: программа просто проигнорирует их, так как первое встреченное соответствие завершает поиск. Таким образом, формируя набор правил, будьте осторожны, старайтесь формировать правила так, чтобы совпадение не возникало тогда, когда вы этого не хотите. Обратитесь также к разделам tcpdchk и tcpdmatch далее в данной главе.

Клиенты, образцы и имена узлов

Во второй колонке записи содержится список имен сетевых узлов, IP-адресов, образцов (см. далее) или шаблонов.

Образец — это часть имени или IP-адреса сетевого узла. Образцы могут быть двух видов: начинающиеся на точку и заканчивающиеся на точку.

Представьте, что ваша система (foo) использует следующую таблицу сетевых узлов:

192.168.0.1 foo.void.org foo 192.168.0.2 bar.void.org
bar 192.168.0.3 baz.void.org baz

В этом случае вы можете использовать образец .void.org, который будет соответствовать трем системам: foo, bar и baz в домене void.org. Если же вы уберете из образца самый первый символ (точку), тогда ему будет соответствовать только один узел с именем void.org, однако такого узла не существует. Подобным же образом вы можете указать образец 192.168.0. (обратите внимание, что в конце стоит символ точки). Этому образцу будут соответствовать все сетевые узлы с IP-адресами, начинающимися на 192.168.0.

Символьные шаблоны, которые можно использовать в колонке демонов, могут использоваться также и в колонке клиентов.

Формы и операторы

Если вы хотите обозначить диапазон IP-адресов, вы можете использовать формат сеть/сетевая_маска, например, если вы укажете 192.168.0.0/255.255.255.128, это будет соответствовать адресам от 192.168.0.0 до 192.168.0.127. Чтобы идентифицировать единственный узел, вы можете использовать тот же самый формат: запись 192.168.0.64/255.255.255.255 соответствует единственному узлу с адресом 192.168.0.64.

Наконец, в качестве образца можно использовать комбинацию символа «@» и имени сетевой группы (используется только для клиентов). Такой формат образца применяется только совместно с NIS. Имя сетевой группы чувствительно к регистру символов.

ПРИМЕЧАНИЕ

Сетевая группа — это механизм, используемый совместно с NIS, поэтому вы можете использовать его только в случае, если вы используете NIS. Сетевая группа выполняет такие же функции, как и обычная группа Unix, однако используется в отношении домена NIS.

В составе списков демонов и клиентов можно использовать оператор EXCEPT (за исключением). Операторы EXCEPT можно вкладывать друг в друга, однако при этом вы должны быть особенно внимательны. Выражение a EXCEPT b EXCEPT c означает (a EXCEPT (b EXCEPT c)). Таким образом, прежде чем использовать подобные операторы, необходимо освежить и проверить ваши знания в области булевой арифметики. Для того чтобы быть полностью уверенным в правильности выражений, я рекомендую вам использовать скобки.

Два основных параметра, которые можно использовать в составе записи файла /etc/hosts.allow, это ALLOW (разрешить) и DENY (запретить). Другие допустимые параметры рассматриваются далее в данной главе.

Правила

Теперь вы обладаете знаниями, достаточными для того, чтобы создать несколько базовых правил:

ALL : LOCAL, .void.org EXCEPT david@bar.void.org : ALLOW

ALL EXCEPT in.telnetd : david@bar.void.org : ALLOW

in.ftpd : ALL : ALLOW ALL :

ALL : DENY

Рассмотрим эти правила по порядку одно за другим, начиная с самого верхнего. Именно в таком порядке они обрабатываются программой TCP Wrappers. Первое правило относится ко всем демонам (в первой колонке стоит ALL). Таким образом, оно действует в отношении всех портов, прослушиваемых метадемоном inetd, для которых этот метадемон запускает tcpd. Так как метка ALL в данном случае обозначает все демоны, первая часть данного правила всегда будет соответствовать всем демонам, совместно с которыми используется tcpd. Однако раздел правила, в котором определяются клиенты, соответствует только локальным адресам, а также всем узлам из домена void.org за исключением пользователя david, который использует для подключения узел bar из домена void.org. Получается не так уж и много адресов. Для всех этих адресов доступ разрешен, так как в первой колонке параметров указан параметр ALLOW (разрешить). В соответствии с этим правилом локальные пользователи, а также пользователи сетевых узлов домена void.org (за исключением пользователя david, работающего с компьютером bar из домена void.org) смогут подключиться к любому из работающих в данной системе демонов, запуск которых осуществляется с использованием tcpd. Исключение составляет пользователь david@bar.void.org. Для него это правило не действует, и если он попытается подключиться к данной системе, для него программа tcpd продолжит обработку правил файла /etc/hosts.allow.

ПРИМЕЧАНИЕ

В последнем абзаце мы обсуждали только демоны, запускаемые с использованием tcpd из файла / etc/inetd.conf. Однако другие демоны, откомпилированные с использованием библиотеки libwrap (библиотека TCP Wrappers), также будут обращаться к данному списку правил, как если бы их вызов осуществлялся бы при помощи tcpd с использованием inetd.

Если при попытке подключения клиента обнаружено совпадение с первым правилом, доступ разрешается, просмотр файла /etc/hosts.allow прекращается и обработка остальных правил не производится. Если же правило не соответствует либо демону, либо клиенту (например, если клиент не является локальным или не принадлежит домену void.org, или является пользователем david, пытающимся подключиться к системе с узла bar в домене void.org), система переходит к обработке следующего правила. Второе правило относится ко всем демонам за исключением демона telnet. В этом правиле определяется совсем короткий список клиентов — лишь один печально известный пользователь david компьютера bar в домене void.org. Если этот пользователь пытается подключиться к любому демону, за исключением telnet, доступ для него открыт (ALLOW). Если же нет (он намерен соединиться именно с telnet), данное правило для него не действует. В этом случае система переходит к обработке следующего правила.

Третье правило относится только к демону FTP. Если подключающийся клиент пытается соединиться с демоном FTP, значит, доступ для него разрешен, кем бы он ни был (в колонке клиентов указана метка ALL). Иными словами, абсолютно всем клиентам разрешается доступ к вашему FTP-узлу.

Последнее правило соответствует всем демонам и всем клиентам и запрещает доступ (DENY). Обработка этого правила выполняется только для пользователя david, работающего с компьютером bar в домене void и пытающегося подключиться к службе telnet, а также для всех клиентов, не входящих в домены LOCAL и void.org, пытающихся подключиться к любому демону, не являющемуся демоном FTP.

Надеюсь, вы овладели основами настройки TCP Wrappers с использованием файла /etc/hosts.allow. Теперь посмотрим, какими дополнительными возможностями позволяет воспользоваться этот файл.

В качестве параметров в любом из правил допускается указывать команды оболочки, сетевые конфигурационные параметры, параметры поиска и разнообразные другие параметры. Директивы spawn и twist позволяют выполнять в составе правила команду оболочки.

Например, директива spawn позволяет выполнить команду оболочки в качестве одного из параметров правила. Эта директива никак не влияет на взаимодействие между клиентом и сервером, так как потоки stdin, stdout и stderr направлены в устройство /dev/null. Наиболее общее использование этой директивы описывается в электронной документации man:

```
spawn (/path/to/safe_finger -l @%h | /usr/bin/mail root) &
```

Эта команда посылает пользователю root электронное сообщение, в котором содержатся результаты выполнения команды safe_finger в отношении подключающейся системы (символьные расширения будут рассмотрены позднее). Эту команду можно использовать совместно с DENY — таким путем вы создадите ловушку, которая будет срабатывать в отношении некоторой службы, доступ к которой вы запретили для внешних клиентов. Не следует использовать такой прием совместно с демоном finger. Дело в том, при попытке получить сведения об удаленном узле с использованием finger этот удаленный узел может точно так же попытаться запустить finger, чтобы получить сведения о вашем узле, а ваш узел опять же попытается вновь запустить finger, и так до бесконечности, а точнее, до тех пор, пока у одного из вас не закончатся ресурсы.

ПРИМЕЧАНИЕ

Информацию, полученную с использованием finger, нельзя считать достоверной. Помните о том, что злоумышленник, пытающийся получить доступ к службам, работающим на вашей системе, может делать это с использованием другой взломанной им системы или применяя подделку IP-адреса (spoofing).

В рассмотренном ранее примере мы могли бы модифицировать третье по счету правило следующим образом:

```
in.telnetd : david@bar.void.org : spawn (/usr/sbin/safe_finger -l @%h \| /usr/bin/mail root) & : DENY
```

В результате пользователь root будет получать сообщение по электронной почте каждый раз, когда пользователь david@bar.void.org будет пытаться подключиться к telnet. Эту строку необходимо добавить сразу же после второй строки в файл /etc/hosts.allow.

Обратите внимание, что длинные правила, такие как рассмотренное, можно разместить на двух строках. Для этого в конце первой строки следует разместить символ обратной косой (\), как это показано в примере. Символ обратной косой является признаком продолжения строки.

Второй способ выполнения команды оболочки основан на использовании директивы twist. Отличие между spawn и twist состоит в том, что spawn перенаправляет все потоки данных в /dev/null, в то время как twist перенаправляет все потоки данных обратно клиенту. Можно использовать, например, следующую директиву:

```
in.ftpd : ... : twist /bin/echo 608 Message to client
```

Данная строка вместо того, чтобы запустить демон ftpd, пересылает клиентам, удовлетворяющим правилу in.ftpd, текст «608 Message to client» (этот текст, естественно, можно заменить на любой другой). Директива twist должна быть самой последней в строке.

В качестве одного из параметров записи файла /etc/hosts.allow можно использовать также следующие директивы: keepalive, linger, rfc931, banner, nice, setenv, umask и user. Вот их предназначение:

- keepalive (без аргументов) — сервер периодически пересылает клиенту пакет подтверждения соединения (keepalive packet). Если клиент не отвечает, сервер разрывает соединение. Такой режим полезен в случае, если вы имеете дело с пользователями, которые выключают свои компьютеры, не удосуживаясь корректно разорвать соединение с сервером;

- linger <*количество_секунд*> — длительность времени, в течение которого ядро продолжает попытки переслать данные клиенту, после того как соединение закрыто;

- rfc931 [таймаут в секундах] — осуществляет запрос пользовательского имени в соответствии с RFC 931. Используется только совместно с TCP. Если клиент не использует IDENT или аналогичную службу идентификации RFC 931 (большинство PC не обладают такой службой), при подключении такого клиента к службе возникает заметное подвисание. Количество секунд в директиве можно не указывать, в этом случае будет использовано значение, по умолчанию заданное внутри исходного кода;

- banners </*путь*/> — содержимое файла, расположенного в каталоге </путь/> и обладающего таким же именем, как процесс демона (например in.telnetd для службы telnet) копируется клиенту. Внутри файла допускается использование символьных расширений (см. далее). Этот механизм применяется только совместно с соединениями TCP;

- nice [число] — изменяет значение nice (по умолчанию 10);

- setenv <*имя значение*> — используется для настройки переменных окружения для тех демонов, которые не переустанавливают свое окружение в момент запуска. При указании значения переменной окружения допускается использовать символьные расширения;

- umask < *число* > — похоже на переменную umask командной оболочки;

- user <*пользователь[.группа]*> — назначает пользователя и (если она указана) группу, от лица которых запускается демон.

Символьные расширения

Символьные расширения — это комбинации специальных символов, которые в процессе обработки текстовой информации заменяются некоторыми данными. Допускается использовать следующие символьные расширения:

- %a (%A) — адрес сетевого узла клиента (сервера);

- %c — какая-либо информация о клиенте (в зависимости от того, что известно), это может быть запись в формате пользователь@узел и IP-адрес или просто IP-адрес; - %d — имя процесса демона (например, in.teLnetd);

- %h (%H) — имя сетевого узла или IP-адрес клиента (сервера);

- %n (%N) — имя сетевого узла клиента (сервера) или «unknown» или «paranoid» если имя недоступно;

- %p — идентификатор процесса (PID) демона;

- %s — информация о сервере: демон@узел или IP-адрес или просто имя демона, в зависимости от того, какая информация доступна;

- %u — имя пользователя (или unknown);

- %% — в результирующем тексте заменяется на обычный символ процента (%).

'

Разнообразные особенности

Несмотря на свою гибкость и богатый набор возможностей, программа TCP Wrappers обладает рядом недостатков. Во-первых, TCP Wrappers не может корректно работать с UDP. Если бы эта программа могла бы обслуживать UDP, ее автор назвал бы ее IP Wrappers или TCP/UDP Wrappers. Одна из причин этого кроется в том, что метадемон inetd вызывает службы UDP с ключом wait (ожидание). Демоны, вызываемые с ключом wait, не покидают очередь ожидания сразу же после приема пакета UDP. Таким образом, информация о первом подключившемся к службе клиенте будет внесена в журнал, однако если второй клиент подключится к этой же службе раньше, чем демон покинет очередь ожидания, сведения о нем не будут внесены в журнал. Помимо этого TCP Wrappers не будет корректно работать также в отношении служб, основанных на RPC, то есть таких, которые в файле /etc/services помечены как rpc/tcp.

Программу TCP Wrappers не следует использовать совместно с web-сервером Apache. Дело в том, что этот сервер обладает встроенным механизмом TCP Wrappers и использование совместно с ним программы tcpd является излишеством. Конфигурирование Apache выполняется приблизительно так же, как и конфигурирование tcpd, поэтому если вы освоили работу с tcpd, вы сможете выполнить также настройку Apache.

tcpdchk

Утилита tcpdchk позволяет проверить наличие синтаксических ошибок в файле hosts.allow. Утилита поддерживает множество ключей командной строки и режимов работы. Например, ключ -d предписывает утилите использовать файл hosts.allow (и hosts.deny, если он используется), расположенный в текущем каталоге, а не в каталоге /etc. Конечно, если текущий каталог совпадает с каталогом /etc, этот режим будет бесполезным для вас, однако воспользовавшись ключом -d, вы сможете протестировать разрабатываемый вами набор правил, прежде чем использовать его на практике.

При использовании ключа -v утилита tcpdchk будет отображать на экране каждую обрабатываемую строку, а также порядок ее обработки. Пример вывода этой утилиты представлен в листинге 15.1. В этом листинге осуществляется обработка набора правил, рассмотренного в самом начале, с внесенными в него добавлениями из дальнейшего текста.

Листинг 15,1. Вывод утилиты tcpdchk

```
# tcpdchk _v
Using network configuration file:  /etc/inetd.conf

>>> Rule /etc/hosts.allow line 1:
daemons:    ALL
clients:  .void.org EXCEPT david@bar.void.org
option:  ALLOW
access:  granted
>>> Rule /etc/hosts.allow line 2:
daemons: ALL EXCEPT in.telnetd
clients: david@bar.void.org
option:  ALLOW
access:  granted
>>> Rule /etc/hosts.allow line 4:
daemons: in.telnetd
clients: david@bar.void.org
option:  spawn (/usr/sbin/safe_finger _1 @client_hostname | /usr/bin/mail root) &
option:  DENY
access:  denied
>>> Rule /etc/hosts.allow line 5:
```

```
daemons: in.ftpd
clients: ALL
option:  ALLOW
access:  granted
>>> Rule /etc/hosts.allow line 6:
daemons: ALL
clients:    ALL
option:  DENY
access:  denied
```

Если утилита tcpdchk не может обнаружить используемый вами файл inetd.conf, вы можете воспользоваться ключом -i, после которого следует указать полный путь к файлу inetd.conf. Этот же ключ может оказаться полезным в процессе тестирования пробной конфигурации inetd (когда тестовый файл inetd.conf располагается в каталоге, отличающемся от /etc).

Наконец, ключ -a проверяет наличие возможности доступа, который явно не разрешен в составе конфигурации tcpd, иными словами, какие демоны могут быть запущены клиентами благодаря тому, что в файле hosts.allow отсутствуют правила, явно разрешающие доступ к этим демонам.

tcpdmatch

Утилита tcpdmatch позволяет выполнять конкретные тесты в отношении ваших конфигурационных файлов. Ключ -d позволяет тестировать файл hosts.allow, расположенный в вашем текущем каталоге (как и в случае с утилитой tcpdchk). Ключ -i позволяет указать полный путь к файлу inetd.conf.

При запуске tcpdmatch необходимо использовать следующий формат:

tcpdmatch демон[@сервер] [пользователь@]клиент

Параметр *сервер* можно использовать для сетевых узлов, оснащенных несколькими сетевыми картами. Некоторые примеры показаны в листинге 15.2.

Листинг 15.2. Пример вывода утилиты tcpdmatch

```
# tcpdmatch in.telnetd bar
warning: bar: hostname alias
warning:      (official  name:  bar.void.org)
client:       hostname bar.void.org
client:       address    192.168.0.2
server:       process    in.telnetd
matched:      /etc/hosts.allow line 1
option:       ALLOW
access:       granted
# tcpdmatch in.telnetd david@bar
warning:      bar:  hostname alias
warning:      (official  name:  bar.void.org)
client:       hostname bar.void.org
client:       address    192.168.0.2
client:       username david
server:       process    in.telnetd
matched:      /etc/hosts.allow line 4
option:       spawn (/usr/sbin/safe_finger _1 Pbar.void.org  |  /usr/bin/mail  root) &
option:       DENY
access:       denied

# tcpdmatch in.ftpd locutus2.calderasystems.com
client:       hostname locutus2.calderasystems.com
client:       address 207.179.39.2
server:       process in.ftpd
matched:      /etc/hosts.allow line 5
option:       ALLOW
access:       granted

# tcpdmatch in.telnetd locutus2.calderasystems.com
client:       hostname locutus2.calderasystems.com
client:       address 207.179.39.2
server:       process in.telnetd
matched:      /etc/hosts.allow line 6
option:       DENY
access:       denied
```

Заключение

В данной главе я рассказал вам о том, что такое TCP Wrappers и как эта программа используется на практике. Вы узнали о гибкости ее конфигурации, а также о том, что частично эта гибкость достигается за счет использования параметров времени компиляции. Вы также узнали о том, какие из параметров были использованы при компиляции версии tcpd, которая вошла в состав OpenLinux. Вы узнали о том, как можно поставить ловушку и как можно отправить сообщение обратно клиенту.

Я также рассказал вам о том, как выполняется тестирование набора правил с использованием утилит tcpdchk и tcpdmatch, которые прилагаются к программе tcpd.

Часть III
Брандмауэры и специальное программное обеспечение

16 Применение брандмауэров, фильтрующих пакеты

В данной главе рассматриваются следующие вопросы:
- построение брандмауэров, фильтрующих пакеты, в рабочей среде Linux;
- настройка ядра для брандмауэра;
- построение простого брандмауэра с использованием ipchains;
- что будет внутри ядер 2.4.x;
а введение в iptables.

В данной главе я расскажу немного о брандмауэрах в среде Linux — что это такое, как они работают и как можно сформировать один из них. Материал данной главы не сделает вас экспертом в области брандмауэров, однако вы познакомитесь с важными концепциями, имеющими отношение к брандмауэрам, кроме того, вы подробнее узнаете о том, как реализовать брандмауэр, фильтрующий пакеты, при помощи программного средства ipchains. В следующей главе будет подробно рассмотрен proxy-брандмауэр на основе squid.

Термин *«брандмауэр»* (английским эквивалентом является термин *firewall)* пришел в компьютерную индустрию из автомобилестроения. Этим термином обозначается огнеупорный щит, используемый в автомобилях (а также в некоторых других транспортных средствах), для защиты пассажиров от огня, возникающего в случае аварии в моторном отделении. В компьютерной индустрии брандмауэром называют аппаратное устройство или программное средство, которое защищает пользователей и данные в локальной сети от атак, исходящих из Интернета (или из сети «экстранет»). Брандмауэр может использоваться также для предотвращения подключения пользователей локальной сети к «запрещенным» узлам Интернета. Кроме того, вы можете использовать брандмауэр для разделения внутренней сети на несколько взаимоизолированных областей.

Некоторые люди слишком поздно узнают о том, что универсальный брандмауэр не может защитить локальную сеть от злоумышленников, действующих внутри этой сети. Однако для изоляции различных отделов организации можно с успехом использовать внутренние брандмауэры. Благодаря этому вы сможете ограничить распространение повреждений и защитить, например, бухгалтерию от конструкторского отдела, а конструкторский отдел от отдела продаж. Вряд ли сотрудники этих отделов должны обладать правом просмотра файлов, принадлежащих сотрудникам других отделов.

Введение в технологию брандмауэров

В среде Linux можно использовать две базовых разновидности брандмауэров. Брандмауэры в среде Linux делятся на *пакетные фильтры* (packet filters) и *брандмауэры proxy.* Каждая из этих двух базовых разновидностей обладает двумя под-разновидностями. Пакетные фильтры могут быть *пересылающими* (forwarding), то есть принимающими решение о том, передавать ли пакет из сети в сеть или нет, *и маскирующими* (masquerading), то есть модифицирующими адрес-источник и адрес-приемник. Брандмауэры proxy могут быть *стандартными* (standard), которые иногда называют *непроницаемыми* (opaque), в которых клиент подключается через специальный порт, а передаваемые данные перенаправляются в другой порт, или *прозрачными* (transparent), в которых клиент не использует специальный порт, а программное обеспечение брандмауэра перенаправляет пакеты из сети в сеть прозрачно для пользователей.

В данной главе в основном речь пойдет о пакетных фильтрах. В следующей главе рассказывается о брандмауэрах proxy, в особенности тех, которые основаны на использовании Squid. Далее я познакомлю вас с технологией маскировки IP (IP Masquerading), которую часто называют трансляцией сетевых адресов (Network Address Translation, NAT).

Пакетные фильтры

Функционирование брандмауэров фильтрации пакетов основано на принципе, что вся информация, которая необходима для принятия решения о том, что делать с пакетом, содержится в заголовке этого пакета. В заголовке пакета содержатся сведения об адресе-источнике, адресе-приемнике, TTL (time to live — время жизни пакета) и многие другие. Здесь же содержится контрольная сумма заголовка, благодаря которой можно определить, был ли заголовок поврежден или нет, а также размер полезной нагрузки. Всего в заголовке IP-пакета содержатся 13 отдельных полей информации, некоторые из этих полей содержат несколько разных кусков информации.

ПРИМЕЧАНИЕ

Подробная информация о заголовке IP-пакета содержится в документе RFC 791.

ПРИМЕЧАНИЕ

Протокол IP не выполняет каких-либо проверок полезной нагрузки, за исключением проверки корректности размера полезной нагрузки. За целостность полезной нагрузки пакета несет ответственность протокол TCP (Transport Control Protocol).

Для реализации фильтрации пакетов в OpenLinux используется программа ipchains. Если вы намерены создать брандмауэр на основе фильтрации пакетов, вы должны принять решения относительно того, в отношении пакетов каких типов ваш брандмауэр будет действовать и что он должен делать с этими пакетами.

Программа ipchains позволяет применять в отношении пакетов набор различных критериев. Эта программа может осуществлять анализ входящих пакетов, исходящих пакетов, а также пакетов, передаваемых через брандмауэр (forwarded packets). Решения могут приниматься на основании адреса, откуда исходят пакеты, адреса, куда они направляются, а также на основании номера порта, для которого они предназначены. Различные правила могут применяться в зависимости от того, является ли пакет пакетом TCP, UDP или ICMP. Наконец, если некоторый тип пакетов не определен явно в рамках набора правил брандмауэра, в его отношении будет применяться определяемая вами политика по умолчанию.

Любой маршрутизатор, шлюз или сетевой узел, который передает пакет из одной сети в другую, осуществляет модификацию заголовка пакета. Однако в процессе этой модификации адрес-источник и адрес-приемник не меняются. Модифицируется контрольная сумма, TTL и, в некоторых случаях, размер пакета, а также смещение фрагмента (если пакет нуждается в фрагментации). Если в ходе обсуждения я говорю о модификации заголовка, я подразумеваю модификацию адресов, содержащихся в заголовке.

Модель OSI (Open Source Interconnect) является популярной (хотя и не совсем точной, однако вполне удобной теоретической парадигмой) моделью, используемой для описания, каким образом пакеты перемещаются *с* прикладного уровня (Application layer) на физический уровень (Physical layer). В рамках OSI для объяснения, каким образом работает сетевое программное обеспечение, используется семь уровней. Для целей данной книги важно лишь отметить, что разные программы работают на разных уровнях модели OSI и что это одна из отличительных характеристик между брандмауэрами proxy и пакетными фильтрами. Однако эта разница чрезвычайно важна. Брандмауэры proxy создают большую нагрузку на систему, так как тщательно анализируют полное содержимое пакетов (иначе говоря, они выполняют *инспекцию в зависимости от состояния* — stateful inspection). Кроме того, брандмауэры proxy сложнее настроить.

ПРИМЕЧАНИЕ

Инспекция в зависимости от состояния (stateful inspection) означает, что программное обеспечение proxy выполняет анализ всего пакета в некотором контексте, то есть в рамках определенного состояния, в то время как пакетные фильтры осуществляют анализ лишь заголовка каждого пакета, не оглядываясь на какой-либо контекст. Инспекция в зависимости от состояния — это более сложный процесс, который требует большего расхода ресурсов.

Какой тип использовать?

Брандмауэры, фильтрующие пакеты, и брандмауэры proxy выполняют аналогичные функции, однако действуют разными способами. Они работают как щиты, защищающие одни сетевые сегменты от других. В этом отношении брандмауэры обеих разновидностей работают с одинаковой эффективностью. Как те, так и другие требуют присмотра и должного конфигурирования. Если конфигурирование выполнено неправильно, как те, так и другие обеспечат вас ложным ощущением безопасности. Как те, так и другие могут обеспечить некоторый уровень защиты, однако и те и другие одинаково уязвимы перед тщательно спланированной высокотехнологичной атакой.

Я еще раз напомню вам, что единственной полностью безопасной системой является система, не извлеченная из продажной упаковки и не включенная в сеть. Такая система вряд ли будет полезной для вас. Вы должны более реалистично смотреть на то, чем наделяют вас брандмауэры. Благодаря брандмауэру вы

получаете запас времени на то, чтобы успеть прореагировать на атаку. В ваши обязанности входит перенастроить брандмауэр таким образом, чтобы обеспечить защиту от атаки. Позвольте мне еще раз подчеркнуть это: брандмауэр дает вам время, благодаря чему вы получаете возможность прореагировать на атаку.

Как правило, решение о том, какую разновидность брандмауэров использовать, формируется на основании индивидуального выбора и опыта, которым вы обладаете. Если ранее вы уже работали с proxy и обладаете необходимым опытом, для вас наверняка будет привычнее использовать брандмауэр на основе proxy. Если вы используете брандмауэр одной разновидности, это не запрещает вам использовать брандмауэр другой разновидности. Некоторые proxy, предназначенные для работы с web-трафиком (http), могут успешно дополнять собой механизм фильтрации пакетов. В частности, Squid очень удобно использовать для блокирования конкретных web-узлов и рекламных баннеров, это намного проще, чем заниматься формированием наборов правил для фильтрации пакетов, исходящих из бан-нерных web-узлов. Напротив, правила ipchains могут быть созданы в основном для документирования сведений о трафике. Наборы правил ipchains могут использоваться совместно с proxy для того, чтобы следить за специальными разновидностями трафика. Таким образом, для вас лучшим вариантов может быть использование комбинации различных брандмауэров — все зависит от целей, которые вы ставите перед собой.

С точки зрения безопасности ни один из типов брандмауэров нельзя назвать лучшим. Однако существует ситуация, в которой, возможно, удобнее использовать ipchains. Эта ситуация, в которой вы хотите модифицировать поле Type of Service (TOS) для того, чтобы оптимизировать передачу трафика. Например, если у вас есть несколько систем, использующих брандмауэр, а также несколько пользователей, обращающихся к нему в одно и то же время, вы можете использовать поле TOS в качестве флага очередности, чтобы назначить пакетам HTTP более высокий приоритет по сравнению с пакетами FTP. Дело в том, что при просмотре Web пользователям, как правило, удобнее, если страницы загружаются через сеть как можно быстрее, в то же время при загрузке FTP-файлов время реакции сети менее важно.

СОВЕТ

При помощи ipchains вы можете модифицировать TOS таким образом, чтобы указать один из следующих параметров: минимальная задержка (minimum delay), максимальная устойчивость (maximum reliability), максимальный объем данных, переданный через канал (maximum throughput), или минимальные затраты (minimum cost).

Физические конфигурации

Обсуждая физическую конфигурацию брандмауэра, вы должны принимать во внимание как оборудование, так и программное обеспечение, используемое для защиты сети. Помните, что основное предназначение брандмауэра — это защита сети, которой вы доверяете, от сети, которой вы не доверяете. Таким образом, сетевой узел, выполняющий функции брандмауэра, — это одновременно и ворота, через которые ваша сеть обменивается данными с другой сетью, и объект повышенного внимания со стороны тех, кто намерен обмануть вашу систему защиты. Если они хотят проникнуть внутрь, прежде всего они должны миновать брандмауэр.

Сетевой узел, выполняющий функции брандмауэра

Для начала необходимо решить, каким должен быть сетевой узел, который вы будете использовать в качестве брандмауэра. Вы можете установить брандмауэр, оснащенный всего одним сетевым интерфейсом, и использовать этот интерфейс как для доверенных подключений, так и для подключений, которым вы не доверяете. Однако данная конфигурация предлагает меньший уровень защиты, поэтому, учитывая относительно низкую стоимость современных сетевых карт, всегда рекомендуется оснастить брандмауэр двумя сетевыми интерфейсами. Благодаря этому вы полностью изолируете одну сеть от другой. В подобной конфигурации миновать брандмауэр будет сложнее для злоумышленников.

ПРИМЕЧАНИЕ

Сетевой узел, который изолирует сеть, которой вы доверяете, от сети, которой вы не доверяете, при помощи двух интерфейсов (по одному для каждой сети), называется бастионным узлом (bastion host). В свое время бастионы использовались для защиты стен замков, теперь бастионы используются для защиты компьютерных сетей от посягательств злоумышленников.

Насколько мощным должен быть компьютер, выполняющий функции брандмауэра? Ответ на этот вопрос зависит от множества факторов. Если вы планируете использовать брандмауэр для соединения двух

сетей со скоростью 10 Мбит/с, при этом вы планируете использовать фильтрацию пакетов и не использовать proxy, будет вполне достаточно использовать для этой цели компьютер с процессором 80486-33 и памятью объемом 16 Мбайт. Однако если вы намерены обеспечить обмен данными с большей скоростью (например, 100 Мбит/с или быстрее), мощности данного процессора будет недостаточно и вы столкнетесь со значительными потерями пакетов.

Настройка ядра для брандмауэра

В данном разделе речь пойдет о конфигурации ядра Linux. Для конфигурирования ядра можно использовать один из трех вариантов (make xconfig, make menuconfig и make config).

Чтобы построить брандмауэр, вы должны переконфигурировать ядро. Чтобы включить фильтрацию пакетов, для ядра необходимо установить несколько параметров. Некоторые из них определяются исходя из используемого вами аппаратного обеспечения, некоторые должны быть отключены, а другие обязательно следует включить. Более подробно о процессе построения своего собственного ядра рассказывается в документации, прилагаемой к вашему комплекту Linux. Первый параметр имеет отношение к завершенности кода и расположен в разделе Code maturity level:

```
CONFIG_EXPERIMENTAL=y
```

Если вы используете в составе ядра так называемые экспериментальные драйверы, это вовсе не означает, что вы чем-то сильно рискуете. Для использования Linux совместно с некоторыми аппаратными конфигурациями, а также для применения в составе ядра некоторых важных механизмов без экспериментальных драйверов просто не обойтись. Таким образом, использовать экспериментальные драйверы необходимо разумно. Экспериментальными называют драйверы, в отношении которых не выполнено столь же тщательное тестирование, которое применялось в отношении других составных частей ядра, поэтому подразумевается, что подобные драйверы могут стать причиной нестабильности и риска с точки зрения безопасности. Экспериментальный код соответствующим образом помечен, и если вы не нуждаетесь в соответствующих возможностях, вы не обязаны добавлять его в ядро. Однако если вы не разрешите использование экспериментального кода, вы не сможете воспользоваться ни одним экспериментальным модулем.

Следующий раздел, на который вам следует обратить внимание, называется Loadable module support (поддержка загружаемых модулей) и имеет отношение к загружаемым модулям. В наше время вы не обязаны компилировать ядро в монолитном режиме и можете использовать *модульное* ядро (более подробно об этом рассказывается в технической документации используемого вами комплекта Linux). В предыдущих версиях ядра (2.0*jc* и ниже) для брандмауэра рекомендовалось использовать *монолитное* ядро. Однако с появлением kmod — загрузчика модулей ядра — модульные ядра стали значительно более безопасными, чем это было раньше. Это связано с тем, что kmod является подпрограммой уровня ядра, а не пользовательским процессом (как, например, kerneld). Некоторые параметры, которые вам необходимы, могут использоваться только в случае, если включена поддержка модулей, поэтому механизм поддержки модулей следует включить. Кроме того, следует добавить в ядро загрузчик модулей ядра (Kernel module loader), благодаря чему модули будут устанавливаться по мере необходимости.

```
CONFIG_MODULES=y    рекомендуется
CONFIG_KMOD=y       рекомендуется
```

В разделе General присутствует один жизненно важный параметр, о котором легко позабыть (к счастью, по умолчанию его значение установлено равным yes): поддержка Sysctl. Этот параметр создает дерево /proc/sys, как об этом рассказывается в главе 6, он обязателен при построении брандмауэра.

```
CONFIG_SYSCTL=y     необходимо
```

Вы также должны внимательно изучить параметры, установленные в остальных разделах. Вам потребуется обеспечить поддержку вашего оборудования (жестких дисков), файловой системы, сетевых карт Ethernet и других коммуникационных устройств (модемов, устройств ISDN, и таких протоколов, как PPP), а также форматов ELF. Однако звук и другие ненужные параметры необходимо отключить.

ПРИМЕЧАНИЕ

Если вы имеете дело с домашней сетью или ваша сеть является сетью небольшого предприятия с относительно небольшой пропускной способностью каналов, возможно, для вас будет излишним использование всех тех параметров, речь о которых пойдет далее. Взвешивайте риск и действуйте в соответствии с конкретной ситуацией.

Выполнив настройку других разделов конфигурации ядра, обратите внимание на раздел Networking options (сетевые параметры). Об этих параметрах рассказывается в следующем разделе главы. Параметры этого раздела являются наиболее важными для темы, обсуждаемой в данной главе.

Сетевые параметры

По сравнению с ядрами версий 2.0.x в ядрах версий 2.2.x появилось множество новых сетевых параметров. Новые параметры могут показаться малопонятными, к тому же электронная подсказка не всегда может оказаться достаточно информативной, поэтому в данном разделе я подробно описываю параметры ядра 2.2.13. В ядрах более поздних версий количество параметров может быть больше. Некоторые из параметров необходимы, другие рекомендуются, а некоторые могут использоваться по вашему желанию.

Параметры, которые помечены как не рекомендованные, могут ослабить защиту вашего брандмауэра. Если вы точно знаете, что они вам не нужны, отключите их.

ПРИМЕЧАНИЕ

Параметры, помеченные далее как по (n), будут показаны в вашем файле .config следующим образом #<имя_параметра> is not set

Вот перечень параметров ядра.

- Первый параметр Packet Socket (пакетный сокет) необходим для функционирования некоторых программ, включая tcpdump. Программа tcpdump, равно как и любая программа, обращающаяся к tcpdump, переводит карту Ethernet в режим прослушивания сети (promiscuous mode), поэтому использовать подобные программы не рекомендуется. Если вам кажется, что этот параметр вам необходим, вы можете скомпилировать соответствующий код в виде отдельного модуля. Этот модуль не должен загружаться в память автоматически, это надлежит делать по запросу. Если какая-либо программа выдает ошибку настройки сокета (setsocket error), загрузите данный модуль и посмотрите, поможет ли это решить проблему.

CONFIG_PACKET=m не рекомендуется, если только не откомпилировать код в виде отдельного модуля

- Сокет связи между уровнем ядра и пользовательским уровнем (Kernel/User netlink socket) для связи требует наличия устройств со старшим номером, (major number) равным 36. Если вы устанавливаете этот параметр, вы также должны установить параметры Routing messages и IP: firewall packet netlink device. Без этих двух параметров данный параметр неуместен.

CONFIG_NETLINK=y рекомендуется

- Параметр **Routing messages** (маршрутизация сообщений) требует создания устройства /dev/route со старшим номером 36, младшим номером 0, благодаря чему вы получаете возможность читать информацию о маршрутизации. Запись в это устройство бессмысленна, однако программы, работающие на пользовательском уровне, в случае необходимости могут читать из этого устройства информацию о маршрутизации.

CONFIG_RTNETLINK=y не обязателен, но рекомендуется

- Эмуляция устройства netlink используется только для обеспечения обратной совместимости, в будущем этот механизм будет удален.

CONFIG_NETLINK_DEV=y необходим

- Если вы намерены использовать брандмауэр фильтрации пакетов, маскировку адресов IP (IP masquerading) или перенаправление портов (port forwarding), вы должны присвоить этому параметру значение yes (да). Если же вы не намерены использовать ipchains или ipmasqadm, а вместо этого используете только лишь брандмауэр proxy (или компьютер является обычным сетевым узлом), присвойте этому параметру значение по.

CONFIG_FIREWALL=y/n необходим в случае, если вы используете ipchains, в противном случае не рекомендуется

- Параметр **Socket Filtering** (фильтрация сокетов) используется программами пользовательского режима для того, чтобы назначать фильтры в соответствие сокетам. Для ipchains этот параметр не требуется.

CONFIG_FILTER=n не рекомендуется

- Параметр **UNIX domain sockets** (сокеты домена Unix) добавляет в ядро код, разрешающий использование сокетов в Linux. Без этого кода механизм syslog не будет работать. Установка этого кода необходима, однако вы можете установить данный код в виде модуля. Для встраиваемых систем (embedded systems) этот параметр должен быть отключен.

CONFIG_UNIX-y необходим

- Параметр **TCP/IP networking** (поддержка сетевого обмена TCP/IP) является еще одним необходимым параметром. Если вы используете TCP/IP для обмена данными с другим узлом, включая localhost, X или какую-либо другую программу, этот параметр необходимо установить.

CONFIG_INET=y необходим

- Многоадресная передача данных (IP multicasting) — это технология, позволяющая одному узлу обмениваться данными одновременно с несколькими сетевыми узлами. Эта технология используется некоторыми программами, обрабатывающими аудио- и видеоинформацию. Этот параметр, как правило, используется в отношении сетевых узлов, а не для брандмауэров.

CONFIG_IP_MULTICAST=y не обязателен

- Параметр **IP Kernel Level Autoconfiguration** (автоконфигурация на уровне ядра) предназначен для бездисковых сетевых узлов, которые загружают свою корневую файловую систему с другого сетевого узла. Брандмауэр должен обладать способностью работать в автономном режиме, поэтому он должен содержать в себе свою полную конфигурацию, таким образом, загружать конфигурацию брандмауэра с другой системы крайне не рекомендуется. Следующие два параметра зависят от данного.

CONFIG_IP_PNP=n не рекомендуется
CONFIG_IP_PNP_BOOTP=n не рекомендуется
CONFIG_IP_PNP_RARP=n не рекомендуется

- Параметр **IP Firewalling** (брандмауэр IP) необходим в случае, если вы планируете использовать ipchains для создания пакетного фильтра или брандмауэра с маскировкой IP-адресов. Если вы используете squid или брандмауэр proxy, данный параметр не является обязательным.

CONFIG_IP_FIREWALL=y необходим

- Параметр **IP Firewall Packet Netlink Device** позволяет создать устройство, куда ipchains будет записывать пакеты. Чтение этого устройства может осуществляться программой пользовательского режима, которая может функционировать в соответствии с содержимым пакетов. Этот параметр необходим в случае, если вы намерены использовать (или написать) подобную программу пользовательского режима.

CONFIG_IP_FIREWALL_NETLINK=y не обязателен

- Параметр **IP Use FWMARK As Routing Key** (использовать значение FWMARK в качестве ключа маршрутизации) позволяет использовать пометку, размещаемую программой ipchains, для маршрутизации.

CONFIG_IP_ROUTE_FWMARK=y не обязателен

- Параметр **IP Transparent Proxy Support** (поддержка прозрачных proxy) необходим для ipchains REDIRECT и для прозрачных proxy. Если вы не используете ни того, ни другого, этот параметр можно выключить. Если вы сомневаетесь, включите его.

CONFIG_IP_TRANSPARENT_PROXY=y не обязателен/необходим

- Параметр **IP Masquerading** (маскировка IP) необходим в случае, если вы используете брандмауэр маскировки IP.

CONFIG_IP_MASQUERADE=y не обязателен/необходим

- Параметр **IP ICMP Masquerading** (маскировка ICMP) необходим только в случае, если вы выбрали параметр CONFIG_IP_MASQUERADE и желаете маскировать исходящие сообщения ICMP. Без этого параметра утилита ping не будет рабо-тать. Кроме того, не будет работать также утилита Microsoft tracert (так как она основана на ICMP вместо UDP).

CONFIG_IP_MASQUERADE_ICMP=y не обязателен/рекомендуется

- Параметр **IP Masquerading Special Modules Support** (поддержка специальных модулей маскировки IP) сам по себе ничего не означает, если вы его выбираете, автоматически устанавливаются следующие три параметра, использование которых требует применения утилиты ipmasqadm.

CONFIG_IP_MASQUERADE_MOD=y не обязателен/необходим

- Параметр **IP autofw masq support** обеспечивает поддержку протоколов, при передаче которых через брандмауэр могут возникнуть проблемы.

CONFIG_IP_MASQUERADE_IPAUTOFW=y не обязателен

- Параметр **IP ipportfw masq support** разрешает перенаправление портов для соединений, поступающих на выбранный порт, при этом перенаправление осуществляется для любого узла и порта. Для

этого необходима утилита ipmasqadm.

CONFIG_IP_MASQUERADE_IPPORTFW=y не обязателен

- Параметр **IP ip fwmard masq-forwarding support** позволяет воспользоваться для перенаправления пакетов возможностью маркировки пакетов, поддерживаемой ipchains.

CONFIG_IP_MASQUERADE_MFW=y не обязателен

- Параметр **IP Optimize As Router Not Host** (оптимизировать как маршрутизатор, а не как обычный сетевой узел) отменяет выполнение операций, направленных на проверку целостности пакета (копирование и сравнение контрольной суммы). Для обычных сетевых узлов использовать данный параметр не рекомендуется, однако для маршрутизаторов данный параметр позволяет получить прирост производительности.

CONFIG_IP_ROUTER=n не обязателен

- Параметр **IP Tunneling** (туннельная передача IP) позволяет выполнить инкапсуляцию IP внутри IP. Этот параметр полезен для портативных компьютеров, которые часто перемещаются из одной подсети в другую и для которых требуется обеспечить использование статического IP-адреса.

CONFIG_NETJPIP=n не обязателен

- Параметр **IP GRE Tunnels Over IP** (туннельная передача GRE через IP) позволяет шлюзу обмениваться сведениями о маршрутизации с маршрутизатором CISCO. Этот параметр можно использовать в случае, если вы подключены к маршрутизатору CISCO.

CONFIG_NET_IPGRE=n не обязателен

- Параметр **IP Broadcast GRE Over IP** (широковещательная передача GRE через IP) разрешает широковещательную передачу сведений о маршрутизации маршрутизаторам CISCO, благодаря чему вы можете сформировать сеть WAN через Интернет. Для этого необходимо включить параметр CONFIG_NET_IPGRE (см. ранее).

CONFIG_NET_IPGRE_BROADCAST=n не обязателен

- Параметр **IP Multicast Routing** (маршрутизация многоадресной передачи) требуется включить только в случае, если ваша система будет осуществлять маршрутизацию многоадресных пакетов. Для этого необходимо включить параметр CONFIG_IP_MULTICAST (см. ранее).

CONFIG_IP_MROUTE=y не обязателен

- Маршрутизация многоадресной передачи данных может осуществляться в одном из трех режимов: плотный режим (dense mode), который используется по умолчанию, разреженный режим 1 (sparse mode version 1) и разреженный режим 2 (sparse mode version 2). Разреженный режим первой версии используется чаще, чем разряженный режим версии 2, поэтому версия 2 потребуется вам только в случае, если нужно обеспечить взаимодействие с системой, работающей в этом режиме. В общем случае поддержка разреженного режима версии 2 вам не потребуется. Таким образом, поддержка IP PIM-SM vl обеспечивает работу в наиболее распространенном режиме, а режим IP PIM-SM v2 используется значительно реже.

CONFIG_IP_PIMSM_Vl=y не обязателен
CONFIG_IP_PIMSM_V2=n не обязателен

- Параметр **IP Aliasing** (псевдонимы IP) необходим только в случае, если вы хотите использовать механизм псевдонимов IP для одной или нескольких установленных в вашем компьютере сетевых карт Ethernet. Благодаря этому вы получаете систему с несколькими IP-адресами (multi-homed system), даже если эта система оснащена только одной сетевой картой. Эта возможность больше не доступна в виде отдельного модуля. Если используется механизм псевдонимов, некоторые сетевые карты переходят в режим прослушивания сети (promiscuous mode). Для брандмауэра настоятельно рекомендуется использовать только один IP-адрес для каждой установленной в нем сетевой карты.

CONFIG_IP_ALIAS=n не рекомендуется

- Если ваша система напрямую подключена более чем к 256 сетевым узлам и оснащена небольшим объемом оперативной памяти (менее 16 Мбайт), вы можете повысить ее эффективность за счет использования демона arpd. Для этого необходимо включить параметр **IP ARP Daemon** (демон ARP). В противном случае данный параметр следует отключить.

CONFIG_ARPD=n не обязателен

- Если ваша система подвержена атакам SYN Denial of Service, вы можете включить параметр **IP TCP SYN Cookie.** Однако включения данного параметра вовсе недостаточно, чтобы этот механизм начал работу, для инициации этого механизма следует отдать специальную команду: echo 1 > /proc/sys/net/ipv4/ tcp_syncookies. Если вы включили механизм TCP SYN Cookies, производительность вашей системы несколько понизится.

CONFIG_SYN_COOKIES=y рекомендуется

- Параметр **IP Reverse ARP** предназначен для совместного использования с демоном rarpd и позволяет системе отвечать на конфигурационные запросы по протоколу ARP. Для брандмауэра это не требуется.

CONFIG_INET_RARP=n не рекомендуется

- Параметр **IP Allow Large Windows** (поддержка больших окон) никоим образом не связан с графическим интерфейсом пользователя. В данном случае термином «окно» обозначается буфер статического размера. Выбрав этот параметр, вы увеличиваете размер этого буфера. Если вы имеете дело с длинными низкоскоростными линиями (например, через спутник) и если у вас в распоряжении более 16 Мбайт памяти, вы можете включить этот параметр.

CONFIG_SKB_LARGE=y не обязателен

- Этот параметр потребуется вам только в случае, если вы намерены поэкспериментировать с протоколом IPv6. Протокол IPv6 несовместим с поддержкой протокола IPv4, встроенной в ipchains, поэтому сообщения, переданные по протоколу IPv6, смогут проходить сквозь ваш брандмауэр вне зависимости от определенного вами набора правил.

CONFIG_IPV6=n не обязателен

- Следующие три параметра — **IPv6 Enable EUI-64 Token Format** (включить формат токена EUI-64), **IPv6 Disable Provider Based Addresses** (отключить адреса, основанные на провайдере) и **IPv6 Routing Messages Via Old Netlink**

(маршрутизация сообщений через старую сетевую связь) — оказывают влияние на вашу систему, работающую в составе экспериментальной сети IPv6. Для их использования также требуется включить параметр CONFIG_IPV6 (см. ранее).

CONFIG_IPV6_EUI64=n не обязателен
CONFIG_IPV6_NO_PB=n не обязателен
CONFIG_IPV6_NETLINK=n не обязателен

- Следующие несколько параметров — **The IPX Protocol** (протокол IPX), **IPX Full Internal IPX Network** (полная внутренняя сеть IPX), **IPX SPX Networking** (сетевая поддержка SPX) и **Appletalk DDP** — позволяют вашей системе работать с протоколами, не являющимися IP (к ним относятся IPX, SPX и AppleTalk). Если вы настраиваете брандмауэр, устанавливать все эти параметры не рекомендуется.

CONFIG_IPX=n не рекомендуется
CONFIG_IPX_INTERN=n не рекомендуется
CONFIG_SPX =n не рекомендуется
CONFIG_ATALK=n не рекомендуется

ПРИМЕЧАНИЕ

Программное средство ipchains работает только с IP, и если вы добавите в систему поддержку каких-либо других протоколов помимо IP, вы раскроете перед злоумышленниками дополнительные возможности обхода вашего брандмауэра.

Очень немногие сетевые узлы нуждаются в поддержке следующих трех параметров, связанных с протоколами X25 и LLC. Если вы желаете обеспечить поддержку этих двух протоколов, внимательно проанализируйте, каким образом они взаимодействуют с используемым вами программным обеспечением брандмауэра.

- Параметры **CCITT X.25 Packet Layer** (пакетный уровень CCITT X.25) и **LAPB Data Link Driver** (драйвер линии связи LAPB) — это первые два параметра, связанные с поддержкой X25 в среде Linux, эти параметры не следует включать для брандмауэра.

CONFI6_X25=n не рекомендуется
CONFIG_LAPB=n не рекомендуется

- Параметр **Bridging** (мостовое соединение) используется для объединения нескольких сегментов Ethernet в единую сеть.

CONFIG_BRIDGE=n не обязателен

- Параметр 802.2 LLC — это последний из параметров поддержки протокола X.25.
CONFIG_LLC=n не рекомендуется

- Если вы обязаны обеспечить поддержку технологии Acorn Computers Econet, вы должны выбрать параметр AUN **over UDP** (AUN через UDP). Параметры **Acorn** Econet/AUN **Protocols** (протоколы Acorn Econet/AUN), AUN **over UDP** (AUN через UDP) и Native **Econet** (естественный Econet) служат для конфигурирования сетевого обмена по технологии Econet. Первый параметр необходим для обеспечения

поддержки AUN, второй — это единственный возможный вариант для использования на брандмауэре, а третий параметр для брандмауэра использовать не следует.

CONFIG_ECONET=n не обязателен
CONFIG_ECONET_AUNUDP=n не обязателен
CONFIG_ECONET_NATIVE=n не рекомендуется

- Параметр WAN **Router** (маршрутизатор WAN) может использоваться в случае необходимости для сетей IP. Использования других протоколов, не являющихся IP, следует избегать, так как благодаря им злоумышленники смогут обойти ваш брандмауэр.

CONFIG_WAN_ROUTER=n не обязателен

- Параметр **Fast Switching** (быстрое переключение) позволяет использовать программное обеспечение для организации замыкания между аппаратными уровнями сетевых карт Ethernet и, таким образом, игнорировать правила ipchains. Этот параметр не следует использовать на брандмауэре.

CONFIG_NET_FASTROUTE=n НЕ ИСПОЛЬЗОВАТЬ

- Некоторые карты Ethernet обладают возможностью замедления работы таким образом, чтобы ваша система получила возможность обработать трафик. Если ваша карта поддерживает такой механизм, вы можете использовать параметр **Forwarding between high-speed interfaces** (передача между высокоскоростными интерфейсами) для того, чтобы замедлить трафик.

CONFIG_NET_HW_FLOWCONTROL=n не обязателен

- Если сетевые карты, установленные в системе, не поддерживают снижение скорости, однако система работает недостаточно быстро для того, чтобы успевать обрабатывать весь трафик, вы можете воспользоваться параметром **CPU is too slow to handle full bandwidth** (процессор слишком медленный для обработки полной пропускной способности). Этот параметр является альтернативой предыдущему параметру CONFIG_NET_HW_FLOWCONTROL

CONFIG_CPU_IS_SLOW=n не обязателен

- Ядро поддерживает около 16 параметров, связанных с обработкой очереди пакетов на основе QoS и/или Fair Queueing. Я не описываю здесь каждый из этих параметров для краткости. Если ваш брандмауэр нуждается в использовании этих параметров, вы можете настраивать их в соответствии со своими предпочтениями.

CONFIG_NET_SCHED=n не обязателен

- Если в вашей системе вы используете программное обеспечение FreeS/WAN (рекомендуется), в данном разделе для вас доступны еще несколько параметров. Первым из них является параметр **IP Security Protocol (FreeS/WAN IPSEC).**

CONFIG_IPSEC=m рекомендуется

- В разделе, связанном с IPSEC, существуют два параметра, использование которых не рекомендуется, а также два параметра, использование которых зависит от вашего желания, все остальные параметры должны обладать значением yes (да).

CONFIG_IPSEC_IPIP=y рекомендуется
CONFIG_IPSEC_PFKEYv2=y рекомендуется

- Параметр **IPSEC Enable Insecure Algorithms** (разрешить использование незащищенных алгоритмов) разрешает использование незашифрованного туннеля. Использовать данный параметр, равно как и параметр **NULL Pseudo-Encryption** (см. далее), не рекомендуется.

CONFIGJPSEC_INSECURE=n не рекомендуется

- Некоторые узлы за брандмауэром могут быть сбиты с толку сообщениями KM P Path MTU. Если такое происходит, отключите данный параметр. При этом скорость соединения снизится, однако повысится надежность работы подобных узлов. Если вы используете только узлы Linux, при этом у вас не должно возникнуть каких-либо проблем.

CONFIG_IPSEC_ICMP=y не обязателен
CONFIGJPSEC_AH=y рекомендуется
CONFIG_IPSEC_AUTH_HMAC_MD5=y рекомендуется
CONFIGJPSEC_AUTH_HMAC_SHAl=y рекомендуется
CONFIG_IPSEC_ESP=y рекомендуется
CONFIG_IPSEC_ENC_3DES=y рекомендуется

- Не следует использовать параметр **NULL Pseudo-Encription Algorithm** (алгоритм псевдошифровки NULL), в противном случае ваше соединение не будет защищенным.

CONFIG_IPSECJNC_NULL-y не рекомендуется

- Параметр **IPSEC Debugging** (отладочный режим IPSEC) может оказаться полезным для решения проблем IPSEC, так как он увеличивает объем полезной диагностической информации, выводимой этим механизмом. Этот параметр рекомендуется, однако использовать его не обязательно.

DEBUG_IPSEC=y не обязателен

Немного о программном обеспечении

После того как вы построили ядро, вы должны удалить из системы все программное обеспечение, которое не требуется для функционирования брандмауэра. Все посторонние программы лучше удалить. Использовать X Window не рекомендуется, так как эта система связывает порты от 6000 до 6010, а также UDP-порт 177 (в случае, если вы используете xdm). Это не означает, что вы должны полностью удалить X Window из системы, просто вы не должны запускать в системе сервер X. Если вам не обойтись без X-сервера, используйте ipchains для того, чтобы блокировать ввод со стороны сети, которой вы не доверяете. Посторонними для брандмауэра являются NFS и другие службы, которые не используются или не нужны. Для управления брандмауэром вам, скорее всего, потребуется такая программа, как ipmasqadm, которая позволяет управлять набором правил ipchains. Эта программа не включается в стандартный комплект OpenLinux, однако она стоит того, чтобы ее установить, особенно если вы используете большое количество правил или собираетесь организовать перенаправление портов. Также удобной может оказаться программа Secure Shell (ssh). Возможно, полезным будет запустить в отношении портов, которые не перенаправляются и не используются (не следует запускать обычные службы на брандмауэре), программу TCP Wrappers. Также вам может пригодиться программа tripwire для наблюдения за файлами. Еще одна программа, которую можно добавить в этот комплект, это программа Perl под названием Courtney. Эта программа (среди прочих, например, perro) следит за попытками сканирования портов. Единственной службой, которую следует запустить на брандмауэре, является служба sshd, которая используется для удаленного администрирования.

Другие соображения

Брандмауэр не следует рассматривать как рядовой сетевой узел вашей сети. Брандмауэр — это компьютерное устройство специального назначения, к которому необходимо относится по-особому. Обычные пользователи не должны обладать возможностью подключения к вашему брандмауэру, кроме того, брандмауэр не должен делать доступными для сетевых пользователей какие-либо свои диски и файлы. Несомненно, для всех учетных записей на брандмауэре следует подобрать надежные пароли, кроме того, на брандмауэре следует использовать механизм сетевых паролей. Хочу специально отметить, что брандмауэр должен использовать пароль, отличающийся от паролей всех остальных узлов сети. Если злоумышленнику удалось подобрать пароль брандмауэра, это не должно означать, что он автоматически получает доступ к другим узлам вашей внутренней сети. Не делайте доступ к внутренним ресурсам слишком простым.

Помимо этого брандмауэр должен быть физически отделен от всех остальных узлов сети. Он должен быть расположен в защищенном месте, в котором неав-торизированные пользователи не смогли бы иметь к нему физический доступ. Любая машина, к которой можно получить физический доступ, может быть взломана иногда всего лишь за пару минут. Корпус компьютера должен быть заперт, а доступ к параметрам BIOS должен быть защищен паролем. Система должна быть настроена на загрузку только с собственного жесткого диска.

Простой брандмауэр фильтрации пакетов

В следующих нескольких разделах я рассмотрю этапы построения очень простого брандмауэра на основе ipchains. Рассматриваемый брандмауэр не следует использовать в том виде, в котором он представлен в данной книге, — вы должны определить, что именно вы хотите получить от брандмауэра, и осуществить соответствующую настройку. Прочитав данный материал, вы поймете, как спланировать и реализовать свой собственный брандмауэр и как самостоятельно сформировать для него набор правил. В реальности все эти процедуры выполняются далеко не так просто, так как очень сложно рассказать обо

всем этом в одной главе, в то время как настройке брандмауэра можно посвятить целую книгу.

Для того чтобы начать, вы должны знать кое-что о сети, из которой исходят соединения, и о сети, в которую направлены эти соединения. В рамках данного раздела я использую следующие параметры:

Внутренняя сеть (сеть, которой вы доверяете): 20?.191.169.128/25 Внешняя сеть (сеть, которой вы не доверяете): 209.191.169.0/25 Узел-бастион: foo с интерфейсами fool/foo2, 209.191.169.1/209.191.169.129

Планирование

Для начала необходимо выбрать базовую политику функционирования вашего брандмауэра. Эта политика может быть одной из следующих: «разрешить все, что не запрещено» или «запретить все, что не разрешено». Первая из этих политик подразумевает более простое изначальное конфигурирование, однако вторая политика является более естественной для брандмауэра, не выполняющего маскировку IP. Если вы имеете дело с брандмауэром, маскирующим IP (см. главу 18), разница между политиками несущественна, так как вы не можете миновать брандмауэр, просто указав адрес во внутренней сети. В рассматриваемом примере мы имеем дело с немаскирующим брандмауэром и выбираем политику запрета.

Внутренняя сеть спроектирована так, что ей можно полностью доверять, никакие работающие в ней службы недоступны для внешних пользователей. Все службы, которые компания желает сделать доступными для пользователей Интернета (анонимный ftp, http и т. п.), расположены в сети, которая не является доверенной. Этот раздел корпоративной сети часто называют демилитаризованной зоной (Demilitarized Zone, DMZ). Зона DMZ является передним краем обороны. Если злоумышленник пожелает проникнуть в вашу сеть, он прежде всего проявит себя именно в зоне DMZ, Если вы держите свои серверы, предлагающие публичный доступ, в зоне DMZ, это означает, что в вашей сети существуют дополнительные системы, в большей степени подверженные внешним атакам, для наблюдения за которыми требуется дополнительное внимание с вашей стороны. Однако, с другой стороны, такой подход препятствует проникновению злоумышленников внутрь доверенной сети, чего нельзя избежать в случае, если вы организуете перенаправление портов через ваш брандмауэр.

Так как сеть рассматривается как сеть с низким уровнем риска, принимается решение запустить на брандмауэре сервер электронной почты. Сервер POP позволит пользователям принимать почту вне зависимости от того, находятся ли они в офисе или дома (от применения ШАР решено отказаться из соображений безопасности). В дальнейшем, когда будет организовано перенаправление портов через брандмауэр, службы SMTP и POP планируется перенести во внутреннюю сеть. После того как эти службы переместятся внутрь (см. главу 18), набор правил брандмауэра останется прежним, так как локальный доступ, равно как и доступ через перенаправление портов, выполняется одинаково. Система DNS будет работать внутри и будет обслуживать только внутреннюю сеть. Первичный и вторичный серверы DNS будут работать на стороне интернет-провайдера, то есть вне сети.

Внутренним клиентам будет разрешено использовать все стандартные службы Интернета, за исключением NNTP. Мы планируем реализовать следующую политику:

Резюме
Политика по умолчанию: запрет (prohibit)
Анонимный FTP-доступ: внешняя сеть (брандмауэр блокирует поступающие извне запросы)
http: внешняя сеть (брандмауэр блокирует поступающие извне запросы)
ssh: используется на брандмауэре -- отключить telnet
smtp: брандмауэр (в будущем: перенаправление портов на внутренний компьютер)
popd: брандмауэр (в будущем: перенаправление портов на внутренний компьютер)
DNS: внутренняя сеть (внешний доступ запрещен)
также следует блокировать входящие запросы ping

Общие сведения об ipchains

Чтобы освоить ipchains, необходимо понять, как работает эта программа. В следующих нескольких разделах я расскажу вам об основных принципах ее работы. Название ipchains происходит от английского слова *chain* — цепочка. В данном случае имеется в виду цепочка правил, в соответствии с которыми брандмауэр осуществляет передачу пакетов из сети в сеть. Подразумевается, что принимаемые брандмауэром пакеты проходят по цепочке правил, когда в отношении пакета выполняется некоторое из этих правил, брандмауэр выполняет в отношении этого пакета указанное в правиле действие. На самом деле ipchains видит пакет только в случае, если этот пакет является первым или единственным пакетом в последовательности. Все последующие фрагменты пакета не передаются по цепочке. Причина проста — сетевой узел не может собрать пакет из нескольких фрагментов, до тех пор пока он не примет самый первый фрагмент последовательности. Если самый первый пакет будет отброшен, все остальные пакеты также будут отброшены по истечении тайм-аута. По умолчанию, если вы выбираете параметр **IP**

Firewalling (брандмауэр), ядро Linux будет автоматически дефрагментировать поступающие пакеты. В результате все пакеты (после дефрагментации) будут передаваться по цепочке правил. Важное замечание относится к ядрам версий до 2.2.13, но после 2.2.10. В двух этих ядрах параметр IP_ALWAYS_DEFRAG сброшен, однако код дефрагментации добавлен некорректно.

ПРИМЕЧАНИЕ

Понятие «цепочка» означает список логически сгруппированных правил. Каждое правило цепочки — это тест, который применяется в отношении заголовка пакета IP.

Каждая цепочка содержит правила, пронумерованные начиная с единицы. Позже будет показано, что на некоторые правила можно ссылаться либо при помощи спецификации, либо при помощи номера.

Спецификация правила — это набор условий, которым должен соответствовать пакет, иными словами, спецификация правила — это тест, который служит для проверки пакетов на соответствие некоторым условиям. Одно правило может существовать одновременно в нескольких цепочках, поэтому при его идентификации следует указать имя цепочки.

Существует семь вариаций командной строки ipchains. Первые шесть из них используют в качестве первого аргумента некоторую команду. В качестве последнего аргумента все шесть этих вариаций принимают набор параметров.

Вот список команд, поддерживаемых ipchains (все команды должны начинаться с символа дефиса).
A (append — добавить) — команда использует два обязательных аргумента: имя цепочки и спецификация правила.
D (delete — удалить) — эта команда также принимает два обязательных аргумента: имя цепочки и спецификацию правила.
C (check — проверить) — (требуется указание -s, -d, -p и -i) — команда использует два обязательных аргумента: имя цепочки и спецификация правила.
I (insert — вставить) — это расширение команды Append (добавить), новое правило добавляется перед правилом, упомянутым в команде.
R (replace — заменить) — (комбинация Insert и Delete) — в качестве обязательных аргументов следует указать имя цепочки, номер правила и спецификацию правила.
D (delete — удалить) — в качестве обязательных аргументов принимает имя цепочки и номер правила; является вариацией ранее описанной команды Delete (удалить), только на этот раз вместо спецификации правила указывается его порядковый номер.
L (list — список) — отображает список правил.) F (flush — очистить) — удаляет все правила.) Z (zero — обнулить) — обнуляет счетчики.
N (new — создать) — создает определенную пользователем цепочку; требует указания имени цепочки, в противном случае срабатывает в отношении всех цепочек.
X (удалить определенную пользователем цепочку) — требуется указать имя цепочки, которая должна быть пустой, в противном случае срабатывает в отношении всех цепочек.
P (policy — задать политику) — в качестве обязательных аргументов принимает имя цепочки и целевое действие.
M (masquerade — маскировать) — требуются следующие обязательные аргументы:

- L (list — список);
- S (установить tcp tcpfin udp).

h (help — помощь) — электронная подсказка (в качестве аргумента можно указать icmp, и тогда на экран будет выведен список кодов и типов ICMP, которые можно использовать в качестве аргументов команд).

Команда masquerade (маскировать) в отличие от целевого действия MASQ требует указания либо L, либо S. Команда S требует указания трех аргументов: тайм-аут tcp (сеанс TCP), тайм-аут tcpfin (сеанс TCP после приема пакета FIN) и тайм-аут udp. Значения этих трех тайм-аутов указываются в секундах.

Седьмая вариация командной строки ipchains — это вызов электронной подсказки. При этом команда не указывается, вместо этого принимается только один параметр и, возможно, один аргумент.

Параметры ipchains

Программа ipchains использует набор параметров. Параметры используются для разных целей, например, для указания адреса или имени интерфейса, а также для поиска правил в обратном направлении и для инвертирования логического смысла некоторого параметра. Для указания маски можно использовать один из форматов: /N или N.N.N.N. Вместо адресов можно указывать имена сетевых узлов. Порты можно идентифицировать либо при помощи численных номеров, либо при помощи имен служб.

Параметр -b позволяет указать одно правило с адресом-источником и адресом-приемником, а

также предписывает ipchains создать правило, в котором эти адреса поменялись местами.

Символ ! (восклицательный знак) можно использовать совместно с некоторыми параметрами для того, чтобы инвертировать их смысл. Вот перечень параметров:

-p протокол — указание протокола. Здесь можно использовать символ ! Например, -p ! icmp — соответствует всем сообщениям, за исключением сообщений icmp. Чтобы идентифицировать все протоколы, можно использовать идентификатор alt;

- -s адрес — адрес-источник. При необходимости можно использовать !, сетевую маску или порт. Имейте в виду, что адрес О/О соответствует всем адресам и является значением по умолчанию в случае, если вы не указываете параметр -s. Для ICMP вместо порта (так как ICMP не использует порты) вы должны указать после -s либо имя ICMP (см. далее), либо номер типа. Перечень типов и кодов ICMP, поддерживаемых ipchains, выводится на экран по команде ipchains -h icmp. Если вы указываете имя, вы не можете одновременно с этим использовать -d код;
- -d адрес — адрес-приемник. К этому параметру относится все, что сказанов отношении параметра -s. Если вы используете -s и указываете номер типа ICMP, вы можете использовать -d и указать код;
- -i имя — имя интерфейса. Можно использовать !. Также к имени интерфейса можно добавить суффикс + для того, чтобы идентифицировать все интерфейсы данного типа. Например, ppp+ обозначает все интерфейсы типа ppp (pppO-pppM);
- -j целевое действие — целевое действие для правила (в качестве целевогодействия можно указать либо имя цепочки, определенной пользователем, либо одно из специальных значений), которое выполняется в случае, если пакет удовлетворяет этому правилу. Если указывается специальное значение REDIRECT, значит, можно указать номер порта;
- -m отметка — число, которым необходимо отметить подходящие пакеты;
- - n — цифровой вывод адресов и портов. По умолчанию ipchains пытаеnся определить соответствующие символьные имена;
- -l — документировать сведения о подходящих пакетах в журнале. Документирование осуществляется ядром;
- -o — вывод направляется в netdev — устройство пользовательского режима;
- -t и хог — маски для поля TOS. Используется для работы с полем TOS (см.далее);
- -v — режим вывода диагностической информации. Отображается адрес интерфейса, параметры правила (если такие есть), маски TOS, а также счетчики пакетов и байтов;
- -x — вывод чисел в расширенной форме. При отображении счетчиков пакетов и байтов сокращения К (кило), М (мега) и G (гига) не используютсяи вместо этого отображаются целиком все разряды числа;
- -f — второй и последующие фрагменты. Перед этим параметром можно поставить модификатор not (отрицание);
- -y — соответствует пакетам TCP, у которых бит SYN установлен. Перед этим параметром можно поставить знак !.

Далее перечисляются поддерживаемые ipchains типы и коды ICMP. - echo-reply (pong) — ответ на запрос ping. - destination-unreachable — место назначения недоступно:

network-unreachable — сеть недоступна;

host-unreachable — сетевой узел недоступен;

- protocol-unreachable — протокол недоступен;
- port-unreachable — порт недоступен;
- fragmentation-needed and DF set — требуется фрагментация, однако бит DF установлен;
- source-route-failed — сбой исходного маршрута;
- network-unknown — сеть неизвестна;
- host-unknown — сетевой узел неизвестен;
- network-prohibited — запрещенная сеть;
- host-prohibited — запрещенный сетевой узел;
- TOS-network-unreachable — TOS сеть недоступна;
- TOS-host-unreachable — TOS сетевой узел недоступен;
- communication-prohibited — обмен данными запрещен;

- host-precedence-violation — нарушение старшинства узлов;
- precedence-cutoff — обход старшинства.

source-quench — подавление источника.
redirect — перенаправление:

- network-redirect — перенаправление сети;
- host-redirect — перенаправление сетевого узла;
- TOS-network-redirect — TOS перенаправление сети;

TOS-host-redirect — TOS перенаправление сетевого узла.
- echo-request (ping) — запрос ping.

router advertisement — объявление о маршрутизаторе.
router-solicitation — принуждение к использованию маршрутизатора.
time-exceeded (ttl-exceeded) — время жизни истекло:

- ttl-zero-during-transit — истекло время жизни во время передачи;
- ttl-zero-during-reassembly — истекло время жизни во время сборки пакета.

parameter-problem — проблема параметра:

- ip-header-bad — плохой заголовок пакета;
- required-option-missing — требуемый параметр отсутствует.

timestamp-request — запрос отметки о времени.
timestamp-reply — ответ на запрос отметки о времени.
address-mask-request — запрос адресной маски.
address-mask-reply — ответ на запрос адресной маски.

В следующей таблице показаны значения, которые следует использовать в случае, если вы хотите реализовать маршрутизацию в соответствии с приоритетами, основанную на Type of Service (TOS).

Имя TOS	Значение	Пример использования
Minimum Delay (минимальная задержка)	0x010x10	ftp, telnet, ssh
Maximum Throughput (максимальный объем передаваемых данных)	0x010x08	ftp-data
Maximum Reliability (максимальная надежность)	0x010x04	snmp, DNS
Minimum Cost (минимальные затраты)	0x010x02	nntp, e-mail

Технология TOS доступна только в случае, если вы добавили соответствующий код в ядро (CONFIG_IP_ROUTE_TOS).

Для использования параметра -o также требуется поддержка на уровне ядра (CONFIG_IP_FIREWALL_NETLINK), а также устройство со старшим номером 36 и младшим номером 3.

Встроенные цепочки

Программа ipchains поддерживает три встроенных цепочки: input (ввод), forward (передача) и output (вывод). Программа ipchains позволяет создавать другие определяемые пользователем цепочки, а также уничтожать их, однако три встроенных цепочки не могут быть уничтожены и должны в любой момент времени содержать по крайней мере по одному правилу (это правило определяет политику по умолчанию). По умолчанию все эти правила определяют для всех пакетов целевое действие DENY (отклонить).

Когда ipchains принимает очередной пакет, этот пакет передается по цепочкам от правила к правилу. Сначала пакет сопоставляется с правилами цепочки input, затем он сопоставляется с правилами цепочки forward и, наконец, он проходит проверку с использованием правил цепочки output. Переход от правила к правилу цепочки осуществляется до тех пор, пока не встретится соответствие. Когда обнаруживается соответствие, обработка цепочки прерывается до тех пор, пока не будет выполнено целевое действие. Если целевого действия не указано, обработка цепочки продолжается со следующего правила.

Правило не обязано обладать целевым действием. Возможно, вы захотите узнать, какое количество пакетов удовлетворяет определенному правилу. Когда пакет удовлетворяет правилу, счетчик правила увеличивается на единицу. Если вы сравните это значение со счетчиком для цепочки, вы сможете определить, какая доля пакетов, передаваемых по некоторой цепочке, соответствует определенному правилу.

Если целевое действие определено, ipchains выполняет его. В качестве целевого действия может быть указано либо имя цепочки пользователя, либо специальное значение. Если в качестве целевого действия указано имя цепочки, определенной пользователем, ipchains приостанавливает обработку текущей цепочки и приступает к обработке цепочки, указанной в качестве целевого действия. Если в этой цепочке не обнаружено ни одного соответствия, ipchains возвращается к обработке изначальной цепочки и продолжает

ее обработку, начиная с правила, которое располагается следующим за тем правилом, в результате тестирования которого обработка изначальной цепочки была приостановлена.

Цепочки правил, определенные пользователем, — это аналог подпрограмм, вызов которых осуществляется при обнаружении соответствия с некоторым правилом.

Если в правиле указано целевое действие, которое не является именем цепочки определенным пользователем, значит, это должно быть одно из следующих специальных значений:

- ACCEPT — принять пакет и перейти к обработке следующей цепочки; - DENY — отбросить пакет и прекратить обработку всех цепочек;

- REJECT — отбросить пакет и прекратить обработку всех цепочек; то же, что и DENY, но при этом генерируется сообщение ICMP Destination Not Reachable (место назначения недоступно);

- MASQ — маскировать пакет — только цепочка forward и цепочки, определенные пользователем;

- REDIRECT — перенаправить пакет локально и перейти к обработке следующей цепочки — только цепочка input и цепочки, определенные пользователем;

- RETURN — немедленно перейти к концу цепочки и продолжить работу.

Цепочки, определяемые пользователем

Цепочки, определяемые пользователем, являются средством логической группировки правил. Такие цепочки вызываются из встроенных в ipchains цепочек в качестве целевых действий. Из любого места цепочки вы можете обратиться к цепочке, определенной пользователем. Если обработка цепочки, определенной пользователем, завершается и при этом в процессе ее выполнения не обнаруживается ни одного совпадения, происходит возврат к обработке вызывающей цепочки.

При создании цепочек, определенных пользователем, имейте в виду, что имя такой цепочки должно иметь не более восьми символов в длину. При именовании следует использовать только строчные (маленькие) буквы, так как заглавные буквы зарезервированы для будущего использования. Имя цепочки, определенной пользователем, не должно совпадать ни с одним из встроенных имен или специальных значений.

Как работает ipchains

Когда ipchains анализирует заголовок пакета IP, производятся следующие действия в указанном порядке.

1. Сравнение контрольной суммы: пакет либо принимается и передается даль ше, либо отклоняется и отбрасывается.

2. Проверка корректности: не является ли пакет искаженным? Если да, то такой пакет отбрасывается.

3. Обработка цепочки input: если не выполнены действия DENY или REJECT, про исходит переход к следующему шагу.

4. Демаскировка (demasquerade): в ответах, адресованных маскированным узлам, происходит перезапись адреса, в противном случае этот шаг пропускается.

5. Маршрутизация: передать пакет локальному процессу (local process) или в це почку forward.

6. Обработка локальным процессом: интерфейс изменяется на lo, и если пакет адресован локальному процессу, происходит обработка цепочек output и input, в противном случае производится обработка только цепочки output, а поддер живает ее локальный процесс.

7. Локальная маршрутизация: если пакет прошел через локальный процесс, но был отправлен не с локального сетевого узла — иными словами, пакет получен от удаленного узла, был обработан локально (см. ранее) для дальнейшего перенап равления (обработка proxy, перенаправление портов и т. п.), и предназначен для передачи на удаленный узел, — пакет передается в цепочку, forward, в противном случае (если пакет локальный) он передается в цепочку output, и если в ней не выполняются ни DENY, ни REJECT, пакет передается локальному узлу localhost).

8. Обработка цепочки forward: эта цепочка предназначена для обработки всех пакетов, которые используют данный узел как шлюз на пути к другому удаленному узлу.

9. Обработка цепочки output: для всех пакетов, покидающих данный узел.

СОВЕТ

Пункт «обработка локальным процессом» несколько сбивает с толку. Однако примите во внимание, что HOST и LOCALHOST — это два различных сетевых узла, и вам многое станет понятным. Чтобы перейти от HOST к LOCALHOST, пакет должен пройти сквозь все правила (за исключением forward), при этом он должен покинуть узел HOST и войти в узел LOCALHOST, затем пройти все процессы, предназначенные для удаленных узлов.

Простые политики брандмауэра

Теперь вы готовы к формированию набора правил брандмауэра. Прежде чем вы приступите к этому, я хочу заострить ваше внимание на нескольких важных моментах. Перед модификацией набора правил лучше изменить значение /proc/sys/net/ ipv4/ip_forward с 1 на 0 для того, чтобы отключить передачу пакетов через брандмауэр. Завершив формирование набора правил, передачу пакетов следует включить. Благодаря этому вы предотвратите непреднамеренное проникновение нежелательных пакетов во внутреннюю сеть, которое может произойти, пока вы редактируете набор правил. Значение этого параметра следует также проверить в случае, если пакеты не передаются через брандмауэр в то время, как вы этого хотите.

ВНИМАНИЕ

Если вы измените все политики во всех встроенных цепочках на DENY и REJECT, убедитесь в том, что вы не используете правил, которые требуют сопоставления имен. Вместо имен сетевых узлов используйте IP-адреса.

Также не стоит забывать о том, что проверка правил осуществляется в определенном порядке. Как только в процессе проверки некоторого правила, содержащего специальное значение, обнаруживается совпадение, обработка цепочки завершается (за исключением ранее описанных ситуаций). Таким образом, следует обращать внимание на то, какие правила располагаются в начале цепочки. Например, правило ipchains -I input 1 -j REJECT (самым первым правилом цепочки input становится правило REJECT) может сработать не так, как вы этого хотите. В правиле отсутствует параметр -s, поэтому оно будет действовать в отношении всех адресов. Более того, так как в нем нет параметра -i, оно будет применяться ко всем интерфейсам. Наконец, благодаря тому, что в нем отсутствует параметр -p, оно будет действовать в отношении всех протоколов. Фактически это правило отклоняет абсолютно все пакеты, даже сообщения локального узла. Иными словами, локальные процессы не смогут обмениваться сообщениями. Скорее всего, это не то, что вам нужно.

Правила необходимо делать простыми и понятными. Формируя набор правил, выполняйте тестирование на каждом этапе.

Что фильтровать и где

Иногда следует учитывать не только то, что вы фильтруете, но и где вы это делаете. Допустим, вы хотите предотвратить отправку ответов на внешние запросы ping для всех узлов внутренней сети. Вы можете использовать один из двух методов.

Во-первых, вы можете запретить или отклонять все эхо-запросы ping следующим правилом:

```
ipchains -A input -s echo-request -j DENY
```

Во-вторых, вы можете блокировать эхо-ответ ping, прежде чем он покинет сеть:

```
ipchains -A output -s echo-response -j DENY
```

Конечный результат обоих методов одинаков: эхо-ответы не будут покидать пределов внутренней сети, однако эффект обоих правил различается. Скорее всего, предпочтительным окажется первый метод. Первый метод блокирует проникновение в сеть любых пакетов ping извне, в то время как второй пропускает внутрь сети эхо-запросы, но блокирует исход из сети эхо-ответов. Если пакет ping, пытающийся проникнуть в сеть, окажется пакетом ping of death (большой ping) и будет адресован внутреннему узлу, подверженному такому типу атак, то первый метод предотвратит атаку, в то время как второй — нет. С другой стороны, первый метод будет блокировать помимо прочих пакеты ping, адресованные локальному узлу. Если в правиле вы укажете интерфейс и будете отбрасывать только пакеты ping, приходящие из Интернета, то вы сможете связываться с брандмауэром при помощи ping (в тестовых целях) как из внутренней сети, так и с узла localhost.

Что не следует отфильтровывать

Некоторые администраторы ошибочно считают, что пакеты ICMP можно не пропускать через брандмауэр, так как эти пакеты не так важны для нормального функционирования сети. Между пакетами ICMP любых типов и пакетами ping ставится знак равенства, что на самом деле неправильно. Помимо запросов ping протокол ICMP используется для передачи через сеть многих других чрезвычайно важных

сообщений. В частности, при помощи ICMP через сеть передаются сообщения destination-not-reachable (место назначения недоступно), поэтому если вы блокируете ICMP, внутренние сетевые узлы не смогут принимать подобные сообщения. Помимо этого система OpenLinux использует сообщения ICMP для настройки MTU (Maximum Transmission Unit — максимально допустимая единица передачи). Для сетевых карт Ethernet значение этого параметра в нормальных условиях равно 1500 для максимизации объема передаваемых данных. Фрагментация пакетов вызывает более длительные задержки, чем уменьшение MTU. По этой причине Linux устанавливает бит DF (Do not Fragment — не фрагментировать). Если сетевой узел или маршрутизатор требует фрагментации пакета, он не сможет это сделать, так как бит DF установлен, по этой причине пакет отбрасывается, а сетевому узлу-источнику посылается сообщение ICMP. Получив такое сообщение, Linux уменьшает размер MTU и пытается передать пакет снова. Так происходит до тех пор, пока пакет не получается передать к месту назначения. Но если вы полностью блокируете ICMP, операционная система не сможет подобрать наиболее эффективное значение MTU, поэтому обмен данными с некоторыми узлами может происходить чрезвычайно медленно.

Многие администраторы, зная о том, что DNS использует UDP через порт 53, пытаются блокировать TCP через этот порт. Однако следует учитывать, что когда DNS выполняет трансферт зоны или любую другую объемную передачу данных, эта система переключается на использование TCP.

Из всего этого следует сделать вывод, что если после введения в силу определенных правил вы начинаете сталкиваться с сетевыми проблемами, попытайтесь отменять правило за правилом до тех пор, пока проблемы не исчезнут. После этого заново вводите их в силу по одному, при этом включив документирование информации в журнале. Таким путем вы сможете локализовать проблему.

Внедрение политик

Теперь все, что вам осталось, это внедрить сформулированные ранее политики. Для начала определим несколько переменных, так как благодаря этому снизится вероятность ошибки. Выполним следующие действия.

1. Будем использовать следующие имена:

```
foolint=209.191.169.1/25          # это интерфейс eth0
foolint=209.191.169.129/25        # это интерфейс eth1
```

1. Блокируем входящие ping-запросы:

```
ipchains -A input -s echo-request -i ethl -j DENY
```

1. Теперь разрешаем пропуск трафика из внутренней сети наружу (за исключе нием NNTP)

```
ipchains -A input -s foolint !  119 -d 0/0 -i ethO -j ACCEPT
```

4. Блокируем службы, которые являются общей проблемой безопасности или не используются вами (такие как telnet, ftp, http и imap):

```
ipchains  -A input   -p  tcp -i    ethl    -s 0/0    -d  foolint   23 -j DENY
ipchains  -A input   -p  tcp -i    ethl    -s 0/0    -d  foolint   telnet -j DENY
ipchains  -A input   -p  tcp -i    ethl    -s 0/0    -d  foolint   80 -j DENY
ipchains  -A input   -p  tcp -i    ethl    -s 0/0    -d  foolint   imap -j DENY
```

На этом шаге (шаг 4) интерфейс eth1 можно не указывать, так как если соответствующие данные проникают в систему через интерфейс ethO, такие пакеты удовлетворяют правилу, определенному на шаге 3, и прохождение цепочки input для них на этом завершается.

5. Принимаем данные DNS (только через foolint), ssh и большинство портов с но мерами выше 1024 (для обеих сетей), а также SMTP и POP-3:

```
ipchains -A input   -p  all  -s O/O   -d  foolint domain -j ACCEPT
ipchains -A input   -p  tcp -s 0/0     -d  fooall 22 -j ACCEPT
ipchains -A input   -p  tcp -s 0/0     -d  fooall  1024:5999 -j ACCEPT
ipchains -A input   -p  tcp -s 0/0     -d  fooall 6010  -j ACCEPT
ipchains -A input   -p  tcp -s 0/0     -d  fooall 25 -j ACCEPT
ipchains -A input -p tcp -s 0/0 -d fooall pop-3 -j ACCEPT
```

1. Принимаем пакеты узла localhost:

```
ipchains  -A input  -i  lo -j ACCEPT
```

2. Блокируем все остальное:

```
ipchains -P input DENY
```

8. Обеспечиваем передачу пакетов в обоих направлениях — правила цепочки input осуществляют остальную фильтрацию:

```
ipchains  -P forward ACCEPT
```

9. Оптимизируем прохождение трафика:
   1) минимальная задержка для трафика Web, telnet и ssh:

   ipchains -A output -p tcp -d 0/0 80 -t 0x01 0x10 ipchains -A output -p tcp -d 0/0 telnet -t 0x01 0x10 ipchains -A output -p tcp -d 0/0 22 -t 0x01 0x10
   1. максимальный объем передаваемых данных для ftp-data:
      ipchains -A output -p tcp -d 0/0 ftp-data -t 0x01 0x08
   2. максимальная надежность для smtp:
      ipchains -A output -p tcp -d 0/0 smtp -t 0x01 0x04
   3. минимальные затраты pop-3:
   ipchains -A output -p tcp -d 0/0 pop-3 -t 0x01 0x10

В завершение следует добавить правило (если пакет прошел все тесты, его следует принять):

ipchains -P output ACCEPT

Следует, однако, позаботиться еще об одном. Необходимо предотвратить подделку IP-адресов (IP-spoofing), когда кто-то из внешних пользователей делает вид, что он находится во внутренней сети. Никто не должен подключаться к внешнему интерфейсу и при этом заявлять, что он находится внутри сети (или является узлом localhost). Следующие правила позволяют документировать сведения о подобных попытках в журнале и отклонять (REJECT) их:

```
ipchains -I input 1 -i eth1 -s foolint -l  -j REJECT
ipchains -I input 1 -i eth1 -s 127.0.0.1 -l   -j REJECT
```

ПРИМЕЧАНИЕ

*Чтобы предотвратить подделку IP-адресов, можно воспользоваться файлом /proc/sys/net/ipv4/conf/ */rp_filter. Достаточно добавить в этот файл единицу («1»). Однако при этом нарушится корректное функционирование модуля FreeS/WAN. Таким образом, для этой цели лучше использовать правила ipchains.*

Тестирование политик

Чтобы протестировать сформированный набор правил, проще всего придумать несколько тестовых случаев и использовать параметр -C программы ipchains. Среди тестовых данных нужно поместить как данные, которые не должны проникать во внутреннюю сеть, так и данные, которые должны туда попасть. Для получения дополнительной диагностической информации во время тестирования следует использовать ключ -v. Так вы сможете узнать, прошел ли пакет через брандмауэр или нет. Помните, что совместно с параметром -C помимо имени цепочки следует использовать ключ -s с указанием адреса и порта, ключ -d с указанием адреса и порта, ключ -p и ключ -i.

Иногда для некоторых правил полезно бывает включить документирование в журнале сведений о тестировании пакетов. Эту возможность следует использовать осторожно, в противном случае за короткое время будут созданы чрезвычайно длинные журналы. Как правило, удобно включать документирование только для одного правила за один раз.

Наблюдение

Не забывайте время от времени проглядывать содержимое журналов, в особенности если вы добавили в брандмауэр правила, специально предназначенные для обнаружения атак. На случай, если кому-то все-таки удалось несанкционированно преодолеть вашу защиту, вы должны использовать для документирования доверенный внутренний сетевой узел.

Применение таких механизмов защиты, как tcpd, tripwire и courtney, впрочем, как и многих других, не даст никаких результатов, если вы не будете присматривать за ними. В первую очередь атакующий будет пытаться нейтрализовать именно эти механизмы. В случае атаки брандмауэр позволяет вам выиграть дополнительное время. Однако время работает на атакующего, то есть против вас. Атаку лучше всего блокировать тогда, когда проникновение за барьер защиты еще не произошло, то есть тогда, когда атакующий еще только пытается обнаружить в вашей системе защиты слабые места. Не выполнив поиска слабых мест, атакующий не сможет проникнуть в систему, поэтому прежде чем произойдет взлом системы, вы получите предупреждение об этом. Заметите ли вы его? Это зависит от вашей бдительности.

Перенаправление портов

Перенаправление портов (port forwarding) — это перенаправление соединения с одного узла на другой узел. С задачей перенаправления портов хорошо справляются брандмауэры proxy. Например, если вы подсоединяетесь с узла foo к узлу bar через порт 80 и при помощи программного механизма этот порт перенаправляется в порт 80 на узле baz, узел foo будет думать, что он подключился к узлу bar через порт 80, в то время как на самом деле он взаимодействует с узлом baz через порт 80. В то же время узел baz будет полагать, что к нему напрямую подключается узел foo. Эта схема может работать в обоих направлениях, поддерживая исходящие соединения внутренних клиентов и входящие соединения внешних клиентов, однако при этом вы получаете возможность контролировать потоки данных.

Программа ipchains не позволяет выполнять перенаправление портов. Одним из целевых действий этой программы является REDIRECT (перенаправление), однако это целевое действие предназначено только для локальных перенаправлений и не позволяет перенаправлять данные на другой сетевой узел. Если вы используете ipchains и желаете организовать перенаправление портов, вы должны использовать модуль ipportfw и программу ipmasqadm. Программа ipmasqadm является полезной оболочкой программы ipportfw. Перенаправление внешних входящих соединений может потребоваться, в особенности если во внутренней (доверенной) сети используется технология маскировки IP (IP masquerading).

ССЫЛКА

Более подробная информация об этом содержится в главе 18 «Защита Samba — маскировка IP и перенаправление портов».

После того как вы настроили набор правил так, как вы этого желаете, и после того, как вы убедились в том, что брандмауэр работает так, как это планировалось изначально, вы можете воспользоваться одной из двух утилит: ipchains-save и ipchains-restore. Название каждой из этих утилит хорошо отражает их основное предназначение: первая сохраняет размещенный в оперативной памяти набор правил в файле на жестком диске, вторая — считывает набор данных из файла в оператив- ную память. Запустив ipchains-save, вы увидите на экране используемые в данный момент цепочки правил. Если содержимое и конфигурация этих цепочек вас устраивают, вы можете запустить утилиту вновь и перенаправить ее вывод в файл. Этот файл в дальнейшем можно использовать для восстановления правил в памяти. Процедуру сохранения правил в файле удобно использовать каждый раз перед тем, как вы внесете в набор правил какие-либо модификации. Благодаря этому в случае необходимости вы сможете восстановить набор правил в его прежнем виде. Кроме того, вы можете скопировать созданный таким образом файл для того, чтобы в дальнейшем добавить в него новые правила, изменить или удалить некоторые из существующих.

Ядра Linux семейства 2.4.x и netfilter

В данном разделе я кратко расскажу об изменениях, которые ожидаются в семействе 2.4.x ядер Linux. Эти сведения основаны на текущем (на момент написания данной книги) состоянии экспериментального ядра 2.3.x. Код netfilter может претерпеть существенные изменения, однако конфигурация ядра 2.3.x, скорее всего, значительно не изменится.

Конфигурационные изменения по сравнению с ядрами 2.2.x

Здесь приводится перечень новых или видоизмененных конфигурационных параметров ядра. Неизменившиеся конфигурационные параметры не обсуждаются и не упоминаются — их описание содержится ранее в данной главе.

- Первым новым параметром является параметр **Packet Socket** (пакетный сокет). Этот параметр имеет отношение к mmap I/O, и его использование не обязательно.

CONFIG_PACKET_MMAP=y не обязателен

- Параметр **Network Packet Filtering** (фильтрация сетевых пакетов) — это то же самое, что и параметр **Network Firewalls** в ядре серии 2.2.x. Если вы формируете ядро для брандмауэра, вы обязаны использовать этот параметр. Параметр **Network Packet Filtering Debugging** (отладка фильтрации сетевых пакетов) может оказаться полезным в случае, если при работе netfilter возникают проблемы. В будущем этот второй параметр может исчезнуть из ядра.

CONFIG_NETFILTER=y необходим
CONFIGJETFI LTER_DEBUG рекомендуется

- Параметр **Kernel Httpd Acceleration** (акселерация httpd на уровне ядра) обеспечивает работу процесса httpd в режиме ядра. Этот процесс является службой, и он не должен быть запущен на брандмауэре.

CONFIG_KHTTPD=n не рекомендуется

- Параметр **Asynchronous Transfer Mode** (ATM) обеспечивает поддержку ATM на уровне ядра и позволяет воспользоваться четырьмя дополнительными параметрами.

CONFIG_ATM=y не обязателен

- Параметр **Classical IP Over ATM** (классический IP через ATM) позволяет передавать данные IPv4 через линию ATM.

CONFIG_ATM_CLIP=y не обязателен

- Параметр **Do NOT send ICMP if no neighbor** (не передавать ICMP, если нет соседа) позволяет обойти проблему исчезающих соседей. Соответствующие сообщения ICMP просто отбрасываются.

CONFIG_ATM_CLIP_NO_ICMP=y не обязателен

- Параметр **LAN Emulation (LANE) support** (поддержка эмуляции LAN) обеспечивает передачу протоколов LAN через ATM. Этот механизм не относится к конкретному протоколу и поэтому может стать причиной того, что пакеты, не являющиеся пакетами IP, смогут проходить сквозь ваш брандмауэр.

CONFIG_ATM_LANE=n не рекомендуется

- Протокол **Multi-Protocol Over ATM (MPOA) support** (Поддержка MPOA) также не относится к конкретному протоколу и может стать причиной проникновения в защищаемую брандмауэром сеть пакетов, не являющихся пакетами IP.

CONFIG_ATM_MPOA=n не рекомендуется

- Параметр **DECnet Support** (поддержка DECnet) включает поддержку еще одного протокола, не входящего в семейство IP. Включать его не следует.

CONFIG_DECNET=n не рекомендуется

Новые модули netfilter

Программа netfilter — эта программа, которая частично работает в режиме ядра. По сравнению с более старым программным средством ipchains программа netfilter более плотно взаимодействует с ядром, поэтому код netfilter работает быстрее. Однако код netfilter еще не интегрирован в ядро и доступен только в виде модулей ядра. Логика работы программы также отличается от ipchains. В отличие от ipchains в netfilter пакеты не проходят через все три цепочки. Если пакет предназначен для локального процесса, он проходит через цепочку input Если пакет предназначен или исходит от сетевого узла, расположенного за брандмауэром, он проходит через цепочку forward. Наконец, если пакет исходит от локального процесса и направляется к удаленной системе, он проходит через цепочку output. Благодаря этим нововведениям скорость работы пакетного фильтра существенно увеличивается.

Относительно большое количество модулей обеспечивает высокую степень настраиваемости. По мере завершения работы над новыми модулями набор модулей будет расширяться.

На момент написания данной книги доступными для использования были следующие модули netfilter:

- forward-fragment.o: поддерживает работу с фрагментами (ключ -f);

- ip_conntrack.o: поддерживает отслеживание соединения TCP/IP для того, чтобы следовать вдоль соединения TCP;

- ip_conntrack_ftp.o: поддерживает отслеживание соединения FTP через IP и обеспечивает работу активных сеансов FTP;

- ip_defrag.o: форсирует дефрагментацию;

- ip_nat.o: включает маскировку IP; является основным модулем маскировки IP (эта технология называется IP masquerading или NAT, Network Address Translation);

- ip_nat_ftp.o: поддерживает активный сеанс FTP в рабочей среде с маскировкой IP;

- ip_nat_map_masquerade.o: отображает маскированный IP-адрес за брандмауэром на IP-адрес, доступный из Интернета;

- ip_nat_map_redirect.o: поддерживает перенаправление;

- ip_nat_map_static.o: осуществляет статическое отображение (статическую маршрутизацию);

- ipt_LOG.o: включает документирование сведений в журнале для netfilter;

- ipt_QUEUE.o: разрешает iptables передавать пакеты в пользовательское адресное пространство;

- ipt_REJECT.o: разрешает отклонение пакетов;

- iptjcmp.o: разрешает программе netfilter работать с ICMP;

- ipt_limit.o: разрешает указывать в составе правил пределы; используется в основном для документирования сведений в журнале;

- ipt_mac.o: поддерживает в составе правил соответствие MAC-адресов;

- ipt_multiport.o: обеспечивает корректное функционирование программ, которые используют множественные порты UDP (например RealPlayer).

- ipt_state.o: поддерживает обработку соответствия состоянию соединения (-m state);

- ipt_tcp.o: разрешает программе netfilter работать с TCP; - ipt_UDP.o: разрешает программе netfilter работать с UDP; - ipt_unclean.o: выполняет некоторые базовые проверки корректности пакетов (-m unclean);

- iptables.o: поддерживает функционирование iptables; является основным модулем iptables;

- netfilter_dev.o: разрешает использование устройства netlink (старший номер 36, младший номер 3).

Перечисляемые далее модули используются для обеспечения обратной совместимости. В некоторый момент времени допускается использование только одного из модулей: iptables, ipchains или ipfwadm. Если вы используете один из них, значит, вы не можете использовать остальные. Кроме того, если вы используете любой из модулей, упоминаемых далее, значит, вы не можете использовать модули, перечисленные ранее.

- ip_fw_compat.o: используется для обеспечения общей совместимости с ipchains/ ipfwadm;

- ip_fw_compat_masq.o: используется для совместимости с ipchains/ipfwadm в отношении IP-маскировки;

- ip_fw_compat_redir.o: используется для совместимости с ipchains/ipfwadm в отношении перенаправления;

- ipchains.o: требуется для того, чтобы разрешить использование ipchains (исключает использование iptables);

- ipchains_core.o: необходимый основной модуль, который используется совместно с ipchains.o;

- ipfwadm.o: требуется для того, чтобы разрешить использование ipfwadm (исключает использование iptables);

- ipfwadm_core.o: необходимый основной модуль, который используется совместно с ipfwadm.o;

Система спроектирована так, чтобы ее можно было легко расширить за счет добавления новых модулей. На самом деле один из упомянутых модулей, ip_nat_ map_null.o, еще не существует, однако его использование планируется в будущих версиях системы.

Некоторые базовые правила netfilter

В среде iphains политикой по умолчанию для всех трех встроенных цепочек была политика DENY (запретить). В netfilter ситуация другая. Целевыми действиями теперь являются ACCEPT (принять), REJECT (отклонить), QUEUE (поставить в очередь), RETURN (вернуться из цепочки), LOG (внести в журнал) и DROP (отбросить). Если вы создали цепочку, определенную пользователем, вы также можете использовать ее в качестве цели. Действия ACCEPT и DROP являются встроенными; REJECT, LOG и QUEUE загружаются в виде модулей перед началом функционирования. Целевое действие RETURN используется для того, чтобы завершить выполнение цепочки. Как уже упоминалось, обработка цепочек теперь выполняется также по-иному. Цепочки INPUT и OUTPUT предназначены для локальных процессов. Политикой по умолчанию для них является политика ACCEPT. Цепочка FORWARD предназначена для пакетов, которые просто передаются через брандмауэр из одной сети в другую. Для цепочки FORWARD политикой по умолчанию является политика DROP. Если вы хотите, чтобы политикой по умолчанию для цепочки FORWARD была политика ACCEPT, при запуске iptables запустите модуль iptables с параметром forward=l. Чтобы сделать политикой по умолчанию REJECT, вы должны уже после запуска iptables добавить специальное правило (вы также можете вручную загрузить модуль ipt_REJECT.o). Обработка цепочек теперь выполняется иначе, поэтому правила приема пакетов следует добавить как в цепочку INPUT, так и в цепочку FORWARD, так как теперь пакеты, передаваемые через брандмауэр, не передаются по цепочке INPUT. Благодаря этому вы можете гибко запретить (DROP) все пакеты, предназначенные или исходящие от локального узла localhost, и при этом никак не повлиять на передачу пакетов через брандмауэр.

Предположим, что у вас есть домашняя сеть, состоящая из двух сетевых узлов, один из которых связан с Интернетом с использованием телефонной линии связи. Представим также, что (для простоты) каждая система обладает собственным IP и ни одна из них не предлагает каких-либо служб внешним пользователям. Чтобы обеспечить работу такой конфигурации с минимальным количеством проблем, достаточно выполнить следующие несколько действий (предполагается, что команда depmod -a уже отдана).

1. Загружаем модули ipt_state и iptables.

```
modprobe  ipt_state
```

Загружаем модуль ip_conntrack и разрешаем использование активного режима FTP. modprobe ip_conntrack_ftp

Создаем политику по умолчанию DROP для всех входящих пакетов, iptables -P INPUT DROP

Явным образом вводим в действие политику по умолчанию для перенаправ ления пакетов на случай, если она будет изменена в будущем.

```
iptables  -P FORWARD DROP
```

5. Создаем определенную пользователем цепочку, для того чтобы не дублировать правила, общие для цепочек INPUT и FORWARD.

```
iptables -N pass
```

6. Вставляем в цепочку INPUT команду перехода к цепочке pass, которая является цепочкой, определенной пользователем.

```
iptables -A INPUT -j pass
```

7. Вставляем в цепочку FORWARD команду перехода к цепочке pass, которая явля ется цепочкой, определенной пользователем.

```
iptables  -A FORWARD -j pass
```

8. Принимаем все входящие пакеты, имеющие отношение к установленному со единению.

```
iptables -A pass -m state --state ESTABLISHED,RELATED -j ACCEPT
```

9. Принимаем все новые соединения, исходящие отовсюду, кроме любой теле фонной линии.

```
iptables -A pass -m state --state NEW -i ! ppp+ -j ACCEPT
```

Можно использовать и другое правило:

```
Iptables -A pass -p icmp --icmp-type echo-reply -j DROP
```

Обе версии последнего правила действуют только в отношении TCP, так как UDP не устанавливает соединения.

10. Отбрасываем входящие пакеты ping.

```
iptables -A pass -p icmp --icmp --icmp-type echo-reply -j DROP
```

Если ранее вы когда-либо уже работали с ipchains, формат перечисленных здесь команд будет хотя бы приблизительно вам понятен. Однако можно заметить, что в netfilter добавлены также некоторые новые возможности.

Далее приводится перечень некоторых отличий (в основном добавлений в netfilter) между ipchains и netfilter. Этот перечень составлен исходя из текущего состояния разработки netfilter. Некоторые возможности, присутствовавшие в ipchains, отсутствуют в netfilter (например, -C или check). Следует отметить, что автор программного средства netfilter работает над созданием утилит iptables-save и iptables-restore, которые служат для сохранения на диске и загрузки с диска набора правил netfilter (аналогично утилитам ipchains-save и ipchains-restore). Те отличия, о которых уже было упомянуто, в данный перечень не включаются.

- В netfilter аналогом флага -i программы ipchains являются два флага: -i и -o. Первый из них (-i) применяется в отношении цепочки INPUT или входной части цепочки FORWARD. Второй флаг (-o) применяется в отношении цепочки OUTPUT и выходной части цепочки FORWARD. Использование флага -i в отношении OUTPUT или флага -o в отношении INPUT запрещено.

- При указании портов необходимо использовать флаг протокола (-p) и использовать формат —sport для порта-источника и —dport для порта-приемника.

- Параметр —tcp-flag (перед которым должен стоять флаг протокола) позволяет тестировать состояние флагов TCP (SYN, ACK, RST, FIN, URG, PSH, ALL, NONE) и действовать в соответствии с этим состоянием. Некоторые правила допускается записывать в сокращенной форме. Например, часть вашего правила может быть записана следующим образом:

```
--tcp-flags SYN.RST.ACK SYN
```

Здесь анализируется состояние флагов SYN, RST и ACK. При этом программа пытается обнаружить пакеты с установленным битом SYN и сброшенными битами RST и ACK. Того же результата можно добиться следующим образом:

```
-m state --state NEW
```

- В netfilter добавлена новая возможность: проверять соответствие определенным условиям, например состояние (одно из значений NEW, ESTABLISHED, RELATED или INVALID), MAC-адрес (разделенные двоеточиями шесть шест-надцатеричных чисел), предел (в основном для документирования сведений в

журнале). При использовании каждой из этих возможностей требуется указать один или более аргументов.

- Из iptables удалены целевые действия MASQ и REDIRECT. Теперь эти действия добавлены в модуль ipnatctl (см. главу 18).

Заключение

В данной главе вы узнали о том, что такое брандмауэр, какие различные типы брандмауэров используются в операционной среде Linux и как они работают. Вы также узнали о том, как работает ipchains и как осуществляется настройка этого программного средства. Вы освоили некоторые приемы формирования правил ipchains. Я также рассказал вам о будущем технологии фильтрации пакетов в Linux. Вы узнали о механизмах, которые должны появиться в ядрах Linux семейства 2.4.x.

17 Применение брандмауэров proxy с использованием Squid

В данной главе рассматриваются следующие вопросы:

что такое серверы proxy;

- конфигурирование Squid;
- конфигурирование клиентов.

В данной главе речь пойдет о настройке службы proxy. Эта служба является базовым типом брандмауэров, и ее можно использовать как отдельно, так и совместно с фильтром пакетов IP или с маскировкой IP (также называемой Network Address Translation, NAT). Комбинирование proxy и пакетного фильтра не обязательно, за исключением некоторых специальных ситуаций, но используя такую комбинацию, вы сможете воспользоваться преимуществами обоих механизмов.

На текущий момент наиболее популярной системой proxy для Linux является программное средство Squid. Эта система является как достаточно мощной, так и в достаточной степени гибкой. Однако, как вы уже, должно быть, знаете, за мощь и гибкость, как правило, приходится платить более высоким риском. К счастью, если только squid не связывает порт с номером меньшим 1024, нет необходимости запускать эту программу от имени root. По умолчанию Squid связывает порт 3128 и работает от лица пользователя nobody. Таким образом, в подобной конфигурации вы можете не беспокоиться за безопасность данного демона.

Прежде чем приступать к установке и использованию squid, необходимо понять, что это за программа и как она работает. Программа squid может выполнять множество разных функций. Она может работать, как обычная система proxy. Это означает, что она принимает входящее соединение через непривилегированный порт и перенаправляет его в привилегированный порт. Именно это является основной функцией брандмауэров proxy. Брандмауэр защищает внутреннюю, доверенную, сеть от внешней сети, которой вы не доверяете. Для этого трафик между двумя сетями блокируется, но при этом сохраняется возможность обмена данными. В этом случае любой желающий передать данные из внутренней сети наружу или в обратном направлении должен подключиться к брандмауэру. Сделав брандмауэр единственной точкой доступа, вы можете чрезвычайно тщательно контролировать вход в сеть и выход из сети.

Вы уже знаете, что порты с номером, меньшим 1024, должны быть связаны от имени пользователя root и что уязвимые места программ, работающих от имени root, могут стать причиной несанкционированного доступа к системе на уровне привилегий root. Чтобы предотвратить это, большинство брандмауэров не связывают какие-либо порты ниже 1024, а делают это только в отношении портов с более высокими номерами. В результате программа может работать от лица непривилегированного пользователя, благодаря чему уровень риска снижается. Возникает проблема: каким образом можно передавать соединения через брандмауэр, не выделяя при этом каждому, кто нуждается в доступе к Web, отдельной учетной записи? Сама по себе идея создания на брандмауэре каких-либо дополнительных учетных записей помимо учетной записи администратора является плохой идеей. Вообще же необходимость подключения к брандмауэру каждый раз, когда у вас возникает желание воспользоваться браузером (и проч.), создает массу неудобств. Для решения проблемы используются proxy. Большинство систем proxy работают прозрачно. Это означает, что вы не обязаны сообщать им имя и пароль для того, чтобы воспользоваться их услугами. На самом деле вам вообще не надо ничего делать, за исключением того, что вы должны перенастроить свой браузер для взаимодействия с портом proxy вместо стандартного порта службы.

По умолчанию squid связывается с непривилегированным портом 3128. После этого любое соединение, принимаемое squid через порт 3128, перенаправляется им далее через порт 80. В результате вы можете блокировать любой трафик через любые порты с непривилегированными номерами и благодаря этому повысить степень защиты вашего брандмауэра.

Однако перенаправление портов — это лишь одна из задач, для решения которых можно использовать squid. На самом деле squid обладает столь большим набором функций и наделяет вас столь большим количеством возможностей, что сокращенный список связанных с ним часто задаваемых вопросов (FAQ) по объему превышает 100 печатных страниц.

Одним из преимуществ программного обеспечения proxy по сравнению с пакетными фильтрами является возможность более глубокого анализа передаваемых пакетов. Благодаря proxy вы можете анализировать не только заголовок IP, но и полезную нагрузку, содержащуюся внутри пакета. Иными словами, вы можете анализировать разделы TCP и UDP пакетов, передаваемых через брандмауэр. На основе информации, содержащейся в заголовке пакета (в разделе IP) и в его полезной нагрузке (в разделе TCP), система proxy может определить, куда направляется пакет и какой службе он предназначен. Исходя из этого брандмауэр proxy принимает решение о том, каким образом следует обработать этот пакет. В частности, благодаря такому подходу squid позволяет организовать кэширование web-страниц для того, чтобы ускорить к ним доступ для большого количества пользователей Web (благодаря этому ускоряется доступ к web-документам и экономится пропускная способность канала связи). Таким образом, squid является помимо прочего кэширующим proxy. Эта программа в течение некоторого времени хранит в своем кэше результаты запроса и использует их для обслуживания последующих аналогичных запросов.

Однако за использование кэширующего программного обеспечения приходится платить дополнительными объемами памяти. Если вы хотите использовать squid, вы должны установить на компьютере объем памяти, достаточный для того, чтобы хранить кэш этой программы. Желательно хранить в оперативной памяти как можно большую долю кэша (а лучше всего — весь кэш). Как только для работы squid становится недостаточно памяти, эта программа начинает использовать виртуальную память, то есть жесткий диск. Конечно же, чтение данных из виртуальной памяти происходит быстрее, чем прием данных через относительно медленную линию связи, однако при использовании виртуальной памяти вы ощутите существенное замедление функционирования, а также затяжные множественные обращения к жесткому диску. Старого компьютера с процессором 486 и памятью объемом 16 Мбайт (а возможно, и 32 Мбайт) будет недостаточно. Если в качестве брандмауэра вы можете использовать только подобную систему, значит, фильтрация пакетов (например, с использованием ipchains) — это лучший вариант, которым вы можете воспользоваться.

Еще одна область, в которой вы можете с успехом использовать squid, — это реализация концепции списков управления доступом (Access Control List, ACL) и списков прав на доступ (Access Rights List, ARL). Списки ACL и ARL снижают степень нецелевого использования соединений с Интернетом. Эти списки позволяют полностью запретить некоторые из соединений. Концепцию ACL и ARL можно применять в случае, если вы не хотите, чтобы ваша система использовалась для доступа к некоторым нежелательным web-узлам, например таким, которые содержат материалы, связанные с жестокостью или порнографией. Списки нежелательных web-узлов (в основном порнографических) можно загрузить опять же из Интернета. Один из них расположен, например, по адресу http:// www.hklc.com (Hong Kong Linux Center), другой можно получить по адресу http:// www.ineparnet.com/orso (Pedro Lineu Orso List). Вы можете использовать этот список прямо через Интернет, указав на него в конфигурационном файле Squid (инструкции содержатся на web-узле).

Механизм ACL и ARL является достаточно гибким, однако его не так просто настроить. Необходимо также понимать, каким образом осуществляется взаимодействие между ACL и ARL. Позже в данной главе я коротко расскажу вам о концепции ACL и ARL, а также поясню, каким образом осуществляется их совместное функционирование.

Наконец, squid можно запустить от имени пользователя root и связать его со стандартным портом httpd (80) для того, чтобы перенаправлять запросы и ускорить доступ к локальному web-узлу. В данном случае вы можете использовать для обслуживания входящих извне запросов HTTP специально выделенную для этой цели изолированную систему. Эта система будет принимать поступающие запросы HTTP и перенаправлять их таким образом, чтобы возвращать клиентам web-документы, размещенные на разных внутренних узлах. Благодаря этому, во-первых, ускоряется доступ к страницам (за счет механизма кэширования squid) а во-вторых, осуществляется распределение нагрузки между несколькими системами — ни одна из них не оказывается перегруженной благодаря обилию запросов.

Приведенные здесь примеры — это лишь некоторые из всевозможных схем использования squid. Далее я расскажу о некоторых достаточно простых приемах конфигурации squid и возникающих при этом проблемах.

Конфигурация по умолчанию

Здесь я не намерен рассказывать вам о том, как происходит компиляция и установка squid. Если вы хотите узнать об этом подробнее, ознакомьтесь с руководством пользователя, файлом часто задаваемых

вопросов (FAQ) и другими документами, предоставленными разработчиками squid. Все это можно получить по адресу http://www.squid-cache.org/DOC/.

В листинге 17.1 показана существенно сокращенная версия очень длинного (более 64 Кбайт) и очень хорошо аннотированного файла squid.conf (который размещается в каталоге/etc/squid/). В данном листинге все пояснения опущены (добавлены некоторые пометки на русском языке).

Листинг 17.1. Скращенный файл /etc/squid/squid.conf

```
#     NETWORK OPTIONS (Сетевые параметры)
#http_port 3128
#icp_port 3130
#htcp_port 4827
#mcast_groups 239.128.16.128
#tcp_incoming_address 0.0.0.0
#tcp_outgoing_address 0.0.0.0
#udp_incoming_address 0.0.0.0
#udp_outgoing_address 0.0.0.0
#     OPTIONS WHICH AFFECT THE NEIGHBOR SELECTION ALGORITHM (Алгоритм выбора соседа)
#cache_peer hostname type 3128 3130
#icp_query_timeout 0
#ncast_icp_query_timeout 2000
#dead_peer_timeout 10 seconds
#hierarchy_stop1ist cgi-bin ?
#acl QUERY urlpath_regex cgi-bin \?
#no_cache deny QUERY
#     OPTIONS WHICH AFFECT THE CACHE SIZE (Настройка размера кэша)
#cache_mem 8 MB
#cache_swap_low 90
#cache_swap_high 95
#maximum_object_size 4096 KB
#ipcache_size 1024
#ipcache_low 90
#ipcache_high 95
#fqdncache_size 1024
#     LOGFILE PATHNAMES AND CACHE DIRECTORIES (Имена файлов журналов и каталогов кэша)
#cache_dir /var/log/squid/cache 100 16 256
#cache_access_log /var/log/squid/logs/access. log
#cache_log /var/log/squid/logs/cache.log
#cache_store_log /var/log/squid/logs/store.log
#cache_swap_log
#emulate_httpd_log off
#mime_table   /etc/squid/mime.conf
#log_mime_hdrs off
#useragent_log none
#pid_filename /var/run/squid.pid
#debug_options ALL,1
#log_fqdn off
#client_netmask 255.255.255.255
#     OPTIONS FOR EXTERNAL SUPPORT PROGRAMS (Внешние вспомогательные программы)
#ftp_user Squid@
#ftp_list_width 32
#cache_dns_program /usr/bin/dnsserver
#dns_children 5
#dns_defnames off
#dns_nameservers none
#unlinkd_program /usr/bin/unlinkd
#pinger_program /usr/bin/pinger
#redirect_program none
#redirect_children 5
#redirect_rewrites_host_header on
#authenticate_program none
#authenticate_children 5
#authenticate_ttl 3600
#     OPTIONS FOR TUNING THE CACHE
#wais_relay_host local host
#wais_relay_port 8000
#request_size 100 KB
#refresh_pattern^ftp:       1440  20%   10080
#refresh_pattern^gopher:     1440  0%    1440
#refresh_pattern          0    20%   4320
#reference_age 1 month
#quick_abort_min 16 KB
#quick_abort_max 16 KB
#quick_abort_pct 95
#negative_ttl 5 minutes
```

#positive_dns_ttl 6 hours
#negative_dns_ttl 5 minutes
#range_offset_limit 0 KB
TIMEOUTS (Таймауты)
#connect_timeout 120 seconds
#siteselect_timeout 4 seconds
#read_timeout 15 minutes
#request_timeout 30 seconds
#client_lifetime 1 day
#half_closed_clients on
#pconn_timeout 120 seconds
#ident_timeout 10 seconds
#shutdown_lifetime 30 seconds
ACCESS CONTROLS (Управление доступом)
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl local host src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563
acl Safe_ports port 80 21 443 563 70 210 1025-65535
acl CONNECT method CONNECT
#Default configuration:
http_access allow manager local host
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access deny all
icp_access allow all
miss_access allow all
#proxy_auth_realm Squid proxy-caching web server
#ident_lookup_access deny all
ADMINISTRATIVE PARAMETERS (Административные параметры)
#cache_mgr webmaster
#cache_effective_user nobody
#cache_effective_group nogroup
#visible_hostname www-cache.foo.org
#unique_hostname www-cachel.foo.org
OPTIONS FOR THE CACHE REGISTRATION SERVICE (Служба регистрации кэша)
#announce_period 1 day
#announce_host tracker.ircache.net
#announce_port 3131
HTTPD-ACCELERATOR OPTIONS (Параметры акселерации HTTPD)
#httpd_accel_host hostname
#httpd_accel_port port
#httpd_accel_with_proxy off
#httpd_accel_uses_host_header off
MISCELLANEOUS (Разнообразные параметры)
#dns_testnames netscape.com internic.net nlanr.net microsoft.com
#logfile_rotate 10
#append_domain .yourdomain.com
#tcp_recv_bufsize 0 bytes
#err_html_text
#memory_pools on
#forwarded_for on
#log_icp_queries on
#icp_hit_stale off
#minimum_direct_hops 4
#cachemgr_passwd secret shutdown
#cachemgr_passwd lesssssssecret info stats/objects
#cachemgr_passwd disable all
#store_avg_object_size 13 KB
#store_objects_per_bucket 50
#client_db on
#netdb_low 900
#netdb_high 1000
#netdb_ping_period 5 minutes
#query_icmp off
#test_reachability off
#buffered_logs off
#reload_into_ims off
#anonymize_headers
#fake_user_agent none
#minimum_retry_timeout 5 seconds
#maximum_single_addr_tries 3
#snmp_port 3401

```
#forward_snmpd_port 0
#Example:
#snmp_access allow public local host
#snmp_access deny all
#snmp_incoming_address   0.0.0.0
#snmp_outgoing_address   0.0.0.0
# DELAY POOL PARAMETERS (all  require DELAY_POOLS compilation option)
#delay_pools 0
#delay_pools 2
#2 delay pools
#delay_class 1 2
# pool 1 is a class 2 pool
#delay_class 2 3
# pool 2 is a class 3 pool
#delay_access 1 allow some_big_clients
#delay_access 1 deny all
#delay_access 2 allow lotsa_little_clients
#delay_access 2 deny all
#delay_parameters pool   aggregate
#delay_parameters pool  aggregate individual
#delay_parameters pool   aggregate network individual
#delay_parameters 1  -1/-1 8000/8000
#delay_parameters 2 32000/32000 8000/8000 600/64000
#delay_initial_bucket_level   50
#incoming_icp_average  6
#incoming_http_average  4
#min_icp_poll_cnt  8
#min_http_poll_cnt  8
#uri_whitespace  deny
#acl  buggy_server url_regex ^http://....
#broken_posts allow buggy_server
#prefer_direct on
#strip_query_terms  on
```

Прежде чем приступать к редактированию конфигурационного файла, вы должны принять во внимание, что некоторые конфигурационные параметры требуют, чтобы компиляция squid была выполнена с учетом этих параметров. Другими словами, если параметр присутствует в конфигурационном файле, это не значит, что он действует. Если вы не уверены в том, был ли включен тот или иной параметр во время компиляции squid, включите его в конфигурационном файле, перезапустите squid и выполните тестирование.

NETWORK OPTIONS

Весь набор записей файла squid.conf разбит на несколько логических групп. Первый раздел является разделом сетевых параметров Network Options. В этом разделе содержатся все параметры, имеющие отношение к обмену данными через сеть, начиная с номеров портов, через которые squid будет ожидать поступления запросов. По умолчанию squid ожидает поступления запросов только через порт 3128, однако он может «слушать» одновременно несколько портов. Набор портов можно задать либо в конфигурационном файле через пробел, либо в командной строке. Если порты указываются в командной строке, перечень портов в конфигурационном файле игнорируется. Порт 80 используется только для акселерации обработки входящих запросов, обращенных к серверу httpd, но не в качестве прозрачного proxy для пользователей, обращающихся к ресурсам Интернета. Если вы будете использовать squid подобным образом, вы рискуете ввести эту программу в бесконечный цикл обращения к кэшу.

В документации squid часто используются термины ICP и HTCP, определение которых зачастую отсутствует. Протокол ICP — это протокол кэширования в Интернете (Internet Cache Protocol), специально предназначенный для организации работы кэширующих proxy. Протокол ICP позволяет кэширующим proxy обмениваться между собой информацией, связанной с кэшированием. Протокол HTCP — это протокол кэширования гипертекста (Hyper Text Cache Protocol). Этот протокол используется для обмена документами, поиск которых осуществляется с использованием ICP. Протокол ICP использует UDP, в то время как протокол HTCP использует TCP. Если у вас есть родительский кэш и/или кэши, осуществляющие параллельное кэширование, то получив запрос на передачу документа, программа squid пытается определить, кэширован ли этот документ локально или он содержится в кэшах, являющихся соседями по отношению к локальному кэшу. Для этой цели используется ICP. Если документ отсутствует в локальном кэше, но при этом положительный ответ ICP приходит быстрее, чем эхо-ответ ping, адресованный интересующему пользователя web-узлу, тогда вместо того, чтобы загружать запрашиваемый документ с указанного пользователем узла, программа squid считает его из соседнего кэша. Таким образом, измеряя расстояние до различных web-узлов и сопоставляя это расстояние со скоростью работы кэша, squid

определяет наиболее быстрый метод получения документа. Используя squid, вы можете стать участником подобной многоуровневой кэширующей системы. Конечно же, для этого вам потребуется получить необходимое разрешение. Дополнительная информация содержится в FAQ. Однако для большинства небольших предприятий, равно как и для индивидуальных пользователей, подобная функциональность не нужна.

ПРИМЕЧАНИЕ

Расстояние в Интернете измеряется в количестве участков ретрансляции. Один участок ретрансляции иногда обозначают термином «хоп» (от английского hop). Если между локальной системой и двумя разными узлами содержится одинаковое количество участков ретрансляции, то расстояние измеряют временем передачи сообщения туда и обратно (round-trip time). Время передачи сообщения туда и обратно — это промежуток времени, который проходит между отправлением запроса некоторому узлу и получением ответа на этот запрос. Программа squid всегда измеряет расстояние временем передачи сообщения туда и обратно, так как для пользователя Web наиболее важным фактором является время, которое необходимо для получения страницы, расположенной на некотором удаленном узле.

Также в разделе Network Options содержатся параметры, связанные с многоадресной передачей данных. Если вы не знакомы с технологией многоадресной передачи данных или ваше соединение с Интернетом не обладает достаточной пропускной способностью, вы не должны пользоваться этими параметрами. Также нет необходимости назначать фиксированные порты для использования совместно с HTCP или ICP, за исключением некоторых специальных ситуаций.

ПРИМЕЧАНИЕ

Многоадресная передача данных (multicast) — это нечто похожее на широковещательную передачу данных. Многоадресная передача позволяет передавать одно сообщение одновременно нескольким IP-адресам. Чтобы воспользоваться этой технологией, вы должны откомпилировать ваше ядро с поддержкой многоадресной передачи данных. Также вы должны обладать клиентом, который поддерживает данную технологию. Кроме того, все маршрутизаторы, соединяющие ваш клиент с Интернетом, также должны поддерживать эту технологию. Многоадресная передача позволяет одному клиенту (в нашем случае Squid) обмениваться информацией с несколькими серверами одновременно. Программа Squid пытается использовать многоадресную передачу данных для обмена данными с другими кэширующими системами squid.

Алгоритм выбора соседа

Параметры алгоритма выбора соседа (Neighbor Selection Algorithm) используются только в случае, если вы взаимодействуете с удаленными кэширующими системами. В конфигурации по умолчанию все эти параметры закомментированы.

Размер кэша

Сразу же после установки squid использует размер кэша по умолчанию. Наиболее важным замечанием для данного раздела является то, что если вы настроили размер кэша, это не означает, что вы ограничили его размер. В случае необходимости размер кэша может превышать значение, заданное вами в конфигурации (а объем оперативной памяти, используемый squid, будет в несколько раз больше, чем размер кэша), однако при этом squid будет стараться как можно быстрее уменьшить размер кэша.

По умолчанию размер кэша составляет 8 Мбайт. Это значит, что squid может использовать более чем 32 Мбайт. Вы должны расценивать значение переменной cache_mem как размер кэша, который будет использоваться squid для «горячих» (то есть очень популярных) сетевых узлов. Однако Squid будет использовать больше памяти, чем указывается в данной переменной: эта программа очень нуждается в оперативной памяти. Чем больше памяти разрешается использовать squid, тем лучше и быстрее будет работать эта программа. Однако объем используемой оперативной памяти будет зависеть также от интенсивности обмена данными с Интернетом через squid. Для домашних пользователей в большинстве случаев установленного значения будет вполне достаточно. Однако если вы имеете дело с корпоративной сетью, в которой в каждый момент времени доступ к Интернету через squid обеспечивается для нескольких сотен пользователей, размер кэша может стремительно вырасти. В зависимости от того, какие другие приложения работают в вашей системе, вы можете разрешить squid использовать в качестве кэша (аргумент cache-mem) вплоть до одной трети всего объема установленной в системе оперативной памяти. Однако если в системе работают какие-либо другие программы, отдавать треть всей физической памяти в распоряжение squid может оказаться слишком расточительным. Чтобы определить необходимый объем, можно воспользоваться следующим методом. Прежде чем запускать squid, запустите программу free и обратите внимание на вторую строку с пометкой -/+ buffers/cache:. В этой строке указываются два

значения: используемая и свободная память. Вы можете либо сохранить в конфигурации squid значение размера кэша по умолчанию (8 Мбайт), либо использовать треть от второго значения строки -/+ buffers/ cache: (с округлением до ближайшего целого количества Мбайт). Например, для системы, оснащенной ОЗУ объемом 64 Мбайт, при помощи данного метода вы получите рекомендуемый размер кэша 12 Мбайт.

В разделе Cache Size Options также содержатся параметры, имеющие отношение к использованию виртуальной памяти. Учитывая склонность squid к интенсивному расходу ОЗУ, вы можете быть уверены в том, что в процессе эксплуатации этой программы могут возникнуть моменты, когда физической памяти будет недостаточно и программа squid будет вынуждена обратиться к виртуальной памяти. В этой категории существует несколько параметров, однако для большинства ситуаций, предусматривающих небольшую и среднюю нагрузку, значений по умолчанию будет вполне достаточно. Если же система squid будет использоваться интенсивно, вы должны обратить внимание на верхний и нижний пределы допустимого расхода виртуальной памяти, так как разница даже в один процент может означать увеличение или уменьшение расхода виртуальной памяти на 25 Мбайт.

LOGFILE PATHNAMES AND CACHE DIRECTORIES

Некоторые наиболее важные элементы, на которые следует обратить внимание в данном разделе, не указываются в конфигурационном файле. Однако обратите внимание, что каталоги журналов и кэша по умолчанию в файле указаны. Вы должны убедиться в том, что пользователь, от лица которого запущен squid (это может быть пользователь squid или любой другой непривилегированный пользователь), обладает правом записи в эти каталоги. Об этом требовании легко позабыть, если squid работает на уровне привилегий более низком, чем root (вообще-то squid не должен запускаться от имени root за, исключением тех случаев, когда вы намерены связать его с портом ниже 1024).

Вы также должны убедиться в том, что на диске достаточно места для хранения журналов и кэшей, которые будет создавать squid. Кэш следует разместить на дисковом разделе, размер которого в два или в три раза превышает размер кэша, установленный для нормального функционирования. В начале эксплуатации squid вы должны обращать внимание на то, насколько интенсивно squid расходует дисковое пространство.

EXTERNAL SUPPORT PROGRAMS

В данном разделе можно указать пользователя ftp. Многие узлы Интернета, обеспечивающие анонимный доступ к ftp, не обращают внимания на то, как именно вы к ним подключаетесь, однако некоторые проверяют корректность адреса электронной почты. Правила хорошего тона и сетевой этикет предусматривают, что сообщаемый вами адрес электронной почты является корректным. Конечно же, все это является для вас актуальным только в случае, если вы используете squid для обеспечения доступа к серверу FTP для тех клиентов, которые используют proxy. Однако proxy используют далеко не все пользователи FTP.

В данном разделе устанавливается также количество запущенных процессов dnsserver, а также параметры, связанные с другими внешними программами. На небольшой системе вы можете уменьшить количество дочерних процессов dnsserver. По умолчанию это количество равно пяти, и этого более чем достаточно для домашних систем. Если вы полагаете, что это количество слишком велико, вы можете уменьшить его, однако для корпоративной сети, в которой количество пользователей, одновременно работающих с Интернетом, в каждый момент времени достаточно велико, установленное значение может оказаться слишком мало.

TUNING THE CACHE

Эти параметры имеют отношение к различным особенностям работы механизма кэширования, например, размер индивидуального кэша на диске, а также время жизни (Time To Live, TTL) для кэшированных DNS-запросов и web-страниц. Эти параметры являются параметрами тонкой настройки. Значения по умолчанию хорошо подходят для домашних пользователей и сетей небольших предприятий. В крупных организациях, скорее всего, потребуется перенастроить эти параметры после того, как вы более тщательно проанализируете нагрузку на squid и интенсивность обращения к кэшу.

TIMEOUTS

Параметры этого раздела служат для изменения значений, заданных в конфигурации TCP/IP по умолчанию. Для большинства состояний IP-соединений значения, по умолчанию используемые squid, соответствуют базовым значениям по умолчанию, заданным в конфигурации TCP/IP. Эти тайм-ауты также влияют на подключения через сокеты Unix к самой программе squid. Очень маловероятно, что у вас может возникнуть необходимость модифицировать значения этих параметров. Если вы плохо представляете себе,

как каждый из этих параметров влияет на сокетные соединения и соединения TCP между клиентами и серверами, вы не должны выполнять модификацию этих значений.

ACCESS CONTROLS

Управление доступом является наиболее важной функцией squid. Если вы не настроите должным образом механизм управления доступом, ваша система будет широко открыта для любого доступа извне (несмотря на то, что в этом случае squid будет использовать некоторую конфигурацию управления доступом по умолчанию). Даже если вы не намерены модифицировать какие-либо другие разделы конфигурационного файла squid, вы должны уделить внимание редактированию раздела параметров управления доступом.

Как уже говорилось ранее, управление доступом включает в себя две составляющие, которые комбинируются для того, чтобы разрешить или запретить доступ к squid или определенным ресурсам или URL. Первой частью системы управления доступом являются списки управления доступом ACL (Access Control List). Запись, определяющая список управления доступом, начинается с метки ad, за которой следует уникальное имя (два разных списка ad не могут обладать одним и тем же именем). После имени следует спецификация, за которой указывается один или несколько аргументов. Аргументы спецификации логически складываются, и если они являются IP-адресами, вместе с ними необходимо указывать сетевую маску. В качестве аргументов можно использовать имена сетевых узлов (при этом подразумевается сетевой узел, обладающий указанным и никаким другим именем), однако делать это следует с осторожностью, так как если вы желаете блокировать доступ к узлу с указанным именем, доступ к этому узлу можно будет получить, указав другое имя или IP-адрес.

После того как вы сформировали список ad, вы можете использовать имя этого списка в следующих далее списках прав доступа ARL (Access Rights List). Список прав доступа указывает, соответствуют ли ресурсам, указанным в acl, какие-либо права доступа. Список прав доступа — это группа прав на доступ по определенному протоколу. За идентификатором протокола следует одно из ключевых слов — allow (разрешить) или deny (запретить), а затем перечень списков ACL, которые логически перемножаются. Иными словами, для того, чтобы был использован некоторый список ARL, необходимо, чтобы были удовлетворены ВСЕ условия, определяемые перечисленными в ARL списками ACL.

Для каждого протокола squid ищет список прав доступа, удовлетворяющий условиям некоторого соединения, и как только такой список ARL обнаруживается, squid прекращает поиск и предпринимает действие, указанное в ARL (либо ALLOW — разрешить, либо DENY — запретить). Если для некоторого соединения не удалось найти ни одного подходящего списка ARL, программа squid смотрит на действие, указанное в последнем правиле (ALLOW или DENY), и выполняет прямо противоположное действие. Таким образом, на самой последней позиции следует расположить либо ARL, который будет соответствовать всем соединениям, для которых не удалось подобрать соответствия ранее, либо правило, которое является противоположным политике по умолчанию.

Начнем с наиболее простого набора правил, приведенного в листинге 17.2. Данный набор списков ACL и ARL является наиболее простым из всех возможных. Он разрешает абсолютно все.

Листинг 17.2. Разрешается полный доступ для всего окружающего мира
```
acl all src 0.0.0.0/0.0.0.0
http_access allow all icp_access allow all
```

В первой строке идентифицируются абсолютно все возможные IP-адреса. Во второй строке для всех этих адресов разрешается доступ через squid по протоколу HTTP. В третьей строке всем этим адресам разрешается также доступ к кэшу squid. Очевидно, что положение вещей, определяемое данным набором правил, не может быть для вас удовлетворительным — еще бы, ведь каждый желающий в любом конце земного шара сможет использовать кэш вашей системы squid. Существуют программы, которые специально предназначены для обнаружения серверов proxy, которые позволяют подключаться к ним всем желающим. Как только такой сервер обнаружен, злоумышленник может взломать его и использовать для каких-либо целей помимо услуг proxy. Взломанная таким образом система proxy может использоваться для сканирования других систем и нападения на другие системы. Чтобы избежать подобного печального исхода, следует ограничить доступ к вашей proxy-системе. В листинге 17.3 представлена несколько более реалистичная базовая конфигурация squid.

Листинг 17.3. Расширенная базовая конфигурация
```
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl allowed_hosts src 192.168.0.0/255.255.0.0 127.0.0.1/255.0.0.0
http_access allow allowed_hosts
http_access deny all
icp_access allow allowed_hosts
icp_access deny all
```

Первая строка листинга 17.3 такая же, как и в листинге 17.2. Вторая строка указывает на протокол

доступа к кэшу squid. В третьей строке идентифицируются узлы, для которых мы хотим разрешить доступ. В этой строке к узлам сети 192.168 добавляется локальный узел localhost (вместо сети 192.168 можно было бы указать сеть 192.168.0 или любую другую допустимую сеть). В большинстве примеров узел localhost идентифицируется IP-адресом 127.0.0.1, однако на самом деле для узла localhost можно использовать любой IP-адрес из диапазона адресов 127. Все адреса этого диапазона, за исключением самого первого, зарезервированы для специального использования, однако указывать их вполне допустимо, но не всегда благоразумно.

Первая связанная с HTTP строка разрешает группе узлов allowed_hosts (соответствующая строка ad добавляет в группу allowed_hosts все узлы сети 192.168, а также локальный узел localhosts) обмен данными по протоколу HTTP. Вторая строка, имеющая отношение к HTTP, запрещает доступ по протоколу HTTP для всех остальных узлов (таким образом, никто из внешнего мира не сможет использовать ваш сервер squid в качестве proxy-сервера). Две строки, имеющие отношение к icp_access, формируют аналогичный набор правил доступа к кэшу вашего сервера squid. Если вы намерены выполнять функции родительского кэша или выполнять кэширование параллельно с другими кэширующими системами, вы должны либо добавить еще один список ad и еще одну строку icp_access, либо, по крайней мере, расширить определение группы allowed_hosts.

Далее я рассмотрю, как выглядит конфигурация, которая входит в состав squid по умолчанию. На этом примере можно понять, насколько гибкими возможностями обладает squid. В листинге 17.4 содержится фрагмент конфигурационного файла squid, полный текст которого приведен в листинге 17.1.

Листинг 17.4. Конфигурация управления доступом, извлеченная из конфигурационного файла squid по умолчанию

```
acl  all   src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563
acl Safe_ports port 80 21 443 563 70 210 1025-65535
acl CONNECT method CONNECT
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access deny all
icp_access allow all
miss_access allow all
```

Первая строка листинга 17.4 совпадает с первой строкой листингов 17.3 и 17.2. В ней определяется группа all, в которую входят все допустимые IP-адреса. Вторая строка определяет протокол, который используется для доступа к кэшу squid. В третьей строке определяется локальный узел localhost. Его адрес 127.0.0.1.

Первая строка http_access разрешает обращение к кэшу squid только для локального узла. Другими словами, данный сервер не будет обеспечивать доступ к своему кэшу для соседних кэширующих систем (это хороший подход для домашних пользователей или небольших корпоративных сетей, в которых используется только один сервер squid). Вторая строка http_access запрещает доступ к кэшу для всех систем, для которых доступ еще не был разрешен (на текущий момент доступ к кэшу разрешен только для одной системы — localhost). Следующая строка http_access запрещает использование каких-либо портов, которые НЕ входят в группу Safe_ports (группа портов Safe_ports определена ранее при помощи соответствующей записи ad). Таким образом, squid не может подключиться к telnet (23), sendmail (25) и т. п. Данная конфигурация также запрещает новые соединения (входящие) к каким-либо портам, не являющимся портами SSL. Вы не сможете запустить какой-либо сервер, принимающий соединения из Интернета, если эти соединения не являются соединениями SSL. Последняя строка http_access запрещает все остальное. Строка icp_access разрешает всем желающим доступ к кэшу squid на данном сервере (возможно, это не самая лучшая идея), и наконец, финальная строка разрешает соседям (другим сетевым узлам squid) использовать ваш сервер squid в качестве родителя или для параллельного кэширования.

Родительским (parent) называется кэширующий узел, который, если не содержит запрашиваемого документа в собственном кэше, будет пытаться загрузить этот документ из сети. *Параллельным* (sibling) называется кэширующий узел, который при отсутствии документа в собственном кэше не будет загружать его из сети, а просто вернет вам сообщение о том, что документ в его кэше отсутствует. Эти правила следует учитывать при указании соседей вашей кэширующей системы. На возвращение клиенту кэшированного документа тратится не так много ресурсов, в то же время на загрузку документа из сети тратится достаточно большое количество ресурсов.

ADMINISTRATIVE PARAMETERS

В данном разделе содержатся некоторые базовые параметры, которые требуется настроить перед тем, как запускать squid. В частности, здесь вы указываете эффективного пользователя кэша (это особенно

важно, если вы запускаете squid от имени root), а также адрес электронной почты лица, ответственного за узел squid. Здесь также можно настроить некоторые другие важные параметры.

CACHE REGISTRATION SERVICE

В данном разделе содержатся параметры, облегчающие обслуживание и администрирование иерархий кэширующих систем.

HTTPD-ACCELERATOR OPTIONS

В этом разделе конфигурации содержатся параметры, которые влияют на работу squid в случае, если эта программа используется только в качестве акселератора HTTPD или кэширующей системы совместно с акселерацией HTTPD. Акселератор HTTPD — это кэш, который помогает вашему web-серверу. Программа Squid используется для кэширования документов из Интернета для того, чтобы обеспечить более быстрый доступ к ним при обслуживании последующих запросов. Акселератор HTTPD извлекает запрашиваемые клиентами документы с вашего web-сервера, поэтому клиенты из Интернета получают более быстрый доступ к вашему web-серверу, так как страницы кэшируются и по запросу клиента извлекаются из кэша.

MISCELLANEOUS

В этом разделе присутствует множество разнообразных параметров, связанных с протоколированием сведений в журналах, выделением памяти, протоколом SNMP и т. п. Для большинства обычных пользователей все эти параметры вряд ли могут представлять какой-либо интерес. Исключение составляют параметры, имеющие отношение к функционированию squid в режиме перенаправления портов (port forwarded), когда squid не должен передавать запросы внутренним системам. К этому режиму имеет отношение параметр never_direct. Этот параметр обрабатывается очень похоже на параметры группы Access Control Options.

DELAY POOL PARAMETERS

Последний раздел параметров имеет отношение к механизму пулов задержки (delay pools). Этот механизм доступен только в случае, если во время компиляции squid вы использовали соответствующий параметр компиляции. Механизм пулов задержки является достаточно сложным, поэтому я не буду подробно описывать его здесь и предоставляю читателям возможность самим определить, нуждаются ли они в использовании этого механизма.

Базовый конфигурационный файл

Теперь, когда мы познакомились с разнообразными конфигурационными параметрами squid, я предлагаю вам изучить базовый конфигурационный файл (squid.conf), который, на мой взгляд, является более удобной отправной точкой для формирования собственных конфигураций, чем конфигурационные файлы, которые предлагаются в комплекте поставки squid и в электронной документации. Предлагаемый мною файл представлен в листинге 17.5.

Листинг 17.5. Базовый конфигурационный файл squid.conf

```
acl  all  src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl allowed hosts src 192.168.0.0/255.255.0.0 127.0.0.1/255.0.0.0
acl SSL_ports port 443 563
acl Safe_ports port 80 21 443 563 70 210 1025-65535
acl CONNECT method CONNECT
http_access allow allowed_hosts
http_access deny manager
http_access allow manager localhost
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access deny all
icp_access allow allowed_hosts
icp_access deny all
miss_access deny all
cache_mgr david@localhost
cache_effective_user nobody
cache_effective_group nobody
```

Самый первый запуск squid

После того как вы должным образом сформировали конфигурационный файл squid.conf, вы можете запустить squid в самый первый раз, однако перед этим необходимо предоставить этой программе возможность самостоятельно выполнить кое-какие важные подготовительные процедуры. Для этой цели требуется выполнить команду squid -z. По этой команде squid подготавливает некоторые необходимые ему рабочие каталоги, включая каталоги, в которых будет размещаться кэш. После этого squid можно запускать в нормальном рабочем режиме либо из командной строки, либо из инициализационного сценария.

Параметры командной строки squid

Программа squid поддерживает набор параметров командной строки, которые могут оказаться полезными в процессе тестирования и отладки. Параметры, указанные в командной строке, обладают большим приоритетом, чем аналогичные параметры, заданные в конфигурационном файле squid.conf. В новых версиях программы squid набор параметров командной строки может меняться (в основном в него будут добавляться новые параметры).

-a порт — использовать указанный номер порта (по умолчанию 3128);) -d уровень — выводить в поток strerr отладочную информацию;

-f файл — вместо /etc/squid/squid.conf использовать конфигурационный файл с указанным именем;

-h — вывести электронную справку;

-k reconfigure|rotate|shutdown|interrupt|kill|debug|check|parse — выполнить обработку конфигурационного файла, затем передать сигнал функционирующей в системе копии (за исключением -k parse) и завершить работу;

-s — включить протоколирование syslog;

-и порт — использовать указанный номер порта ICP (по умолчанию 3130), значение 0 запрещает использование этого порта;
-v — отобразить версию;
-z — создать рабочие каталоги;
-C — не реагировать на фатальные сигналы;
-D — отключить инициализационное тестирование DNS;
-F — быстрая повторная сборка хранилища;
-N — отключение режима демона;
-R — не устанавливать REUSEADDR для порта;
) -V — акселератор httpd для виртуального сетевого узла;

-X — форсировать полную отладку;

-Y — в процессе быстрой перезагрузки возвращать только UDP_HIT или UDP_ MISS_NOFETCH.

При использовании ключа -d следует указать уровень отладки, который должен быть значением от 1 до 9. Значение 1 включает наименее информативный режим отладки, значение 9 соответствует наиболее информативному режиму отладки.

Отладка

Учитывая огромное количество конфигурационных параметров, поддерживаемых squid, можно предположить, что настройка этой программы может оказаться непростым делом. В некоторых случаях, модифицируя параметры, вы можете нарушить корректность работы squid. Даже если программа squid запускается, она может вести себя не так, как вы этого хотите. Чтобы должным образом настроить squid и обеспечить оптимальную производительность этого приложения, вы должны потратить время на изучение порядка функционирования этой программы. Для этой цели вы можете обратиться к разнообразной документации (например, FAQ и руководство пользователя), где вы сможете почерпнуть полезную информацию о том, как сформировать конфигурацию для вашего конкретного случая. Одна из наиболее неприятных проблем проявляет себя следующим образом: вы подставляете в squid конфигурацию, о которой заведомо известно, что она работает (например, содержимое листинга 17.5), однако имеющийся у вас исполняемый файл squid выдает на экран многочисленные сообщения об ошибках, которые говорят вам о том, что источником проблемы являются те или иные строки act (несмотря на то, что эти строки являются абсолютно корректными), после этого программа завершает работу со сбоем сегмента. Подчас программа squid говорит вам о том, что проблема кроется в строках, которые вообще отсутствуют в используемом вами конфигурационном файле squid.conf (например, программа жалуется на синтаксическую ошибку в строке ad с номером 58, в то время как в конфигурационном файле содержится всего лишь 40 строк). Если вы столкнулись с подобными сложностями, заново откомпилируйте squid, при этом используйте меньшее количество параметров компиляции. После этого установите squid заново — скорее всего, программа заработает как полагается.

На стороне клиента

После того как вы настроили и запустили squid, вы должны настроить ваших клиентов на использование этой программы. Если ваш брандмауэр выполняет перенаправление обычных соединений (иными словами, если вы можете соединиться с узлами Интернета, не перенастраивая ваш браузер на использование squid), скорее всего, будет лучше, если вы заблокируете этот маршрут таким образом, чтобы клиенты не смогли обойти squid для всех протоколов, обрабатываемых этой программой (http, ftp, gopher, wais). Это относится также и к SSL.

После этого перенастройте клиентское программное обеспечение таким образом, чтобы оно указывало на порт proxy. В Netscape Communicator для этой цели следует воспользоваться ниспадающим меню Edit (правка), где следует выбрать пункт Preferences (параметры). В разделе Advanced (дополнительно) содержится группа элементов Proxies. Выберите Manual proxy configuration (настройка proxy вручную), затем выберите View. После этого вы можете ввести в графы FTP proxy, Gopher proxy, HTTP proxy и WAIS proxy адрес сетевого узла и номер порта squid. Вы также можете указать, для каких узлов proxy использовать не следует.

Расширение squid

Программное средство squid пользуется большой популярностью, и к настоящему времени различными людьми разработано множество программ, которые облегчают работу с squid. Перечень подобных программ постоянно расширяется — в него добавляются новые программы, а также в нем появляются обновления и новые версии уже существующих программ. Среди подобных программ — средства, которые позволят вам выполнять множество полезных функций и операций, например, чтение файлов журналов, а также перенаправление портов и многое другое. Текущий перечень добавлений к squid содержится на web-узле squid по адресу http://www.squid-cache.org/.

Заключение

В данной главе я кратко познакомил вас с программой squid, которая является наиболее популярным кэширующим proxy-сервером для операционной системы Linux. Вы получили базовый набор знаний о функционировании squid и управлении этим программным средством. Вы получили представление о структуре конфигурационного файла, формате списков управления доступом ACL, а также других конфигурационных параметрах squid. Вы также узнали о том, как выполняется настройка программ на стороне клиента, а также о том, какими преимуществами наделяет вас Squid.

18 Маскировка IP и перенаправление портов

В данной главе рассматриваются следующие вопросы:
- базовые сведения о дизайне сети;
- маскировка IP: что это, зачем и почему;
- что такое перенаправление портов;
- что ожидается в ядрах Linux семейства 2.4.x.

В любой ситуации, в которой вам требуется подключить к Интернету более чем один компьютер (вне зависимости от того, одновременно или нет), вы обнаруживаете, что перед вами встает необходимость некоторым образом спроектировать вашу сеть. Многие из тех, кто не занимался этим ранее, не знают, с чего начать и какими возможностями они могут воспользоваться. Проектируя сеть, приходится иметь дело с множеством конфликтующих, подчас противоречащих друг другу обстоятельств, при этом вы должны найти компромиссное решение, которое было бы одновременно и гибким, и в достаточной степени безопасным.

Для домашнего пользователя, у которого не возникает необходимости поддерживать внутреннюю систему доменных имен и который не собирается предлагать свои службы другим пользователям Интернета, выбор весьма прост: использовать базовый сервер IP-маскировки, которого вполне достаточно для обеспечения обмена данными с Интернетом. В качестве сетевого узла, обеспечивающего маскировку, может использоваться любая устаревшая система. Если для подключения к Интернету вы используете телефонную линию, аналоговый модем или ISDN, для маскировки вполне сгодится компьютер 386-20 с памятью объемом 16 Мбайт, на котором можно установить урезанное ядро Linux (без X-сервера). Если у вас в распоряжении более быстрая линия, например 10 Мбит/с, функции маскировки сможет выполнять компьютер 486-33 с памятью объемом от 16 до 32 Мбайт.

Если у вас под рукой нет ни одного старого компьютера или вы желаете использовать для подключения к Интернету более быстрый компьютер, вы можете использовать для этой цели вашу рабочую станцию. Однако в этом случае необходимо принимать во внимание, что на рабочей станции будут функционировать дополнительные программы и службы, поэтому вы должны быть очень внимательным, выбирая, какие из них будут доступны для внешних пользователей. Эта система будет защищать все остальные ваши системы от злоумышленников, поэтому для обеспечения максимального уровня защиты вы должны запустить в этой системе как можно меньшее количество программ (желательно вообще не запускать на ней служб).

ССЫЛКА

Помимо информации, излагаемой в данной главе, в рассматриваемой ситуации полезным для вас окажется материал главы 16, посвященной использованию пакетных фильтров.

Если вы являетесь пользователем, подключающимся к Интернету через телефонную линию, скорее всего, на время подключения вам выделяется единственный IP-адрес. Вы можете обладать статическим IP-адресом, который не меняется от подключения к подключению, однако, скорее всего, при подключении вам будет выделяться динамический IP-адрес, случайно выбираемый из некоторого набора IP-адресов. Динамический IP-адрес будет меняться от подключения к подключению. На самом деле принципиальной разницы нет, и принципы, в соответствии с которыми вы должны проектировать вашу сеть, не зависят от того, используете ли вы статический или динамический IP-адрес. Более подробно о маскировке IP будет рассказано позже в данной главе. А сейчас давайте сосредоточим внимание на проектировании вашей сети.

Начнем с самого простого и постепенно будем усложнять задачу. Если вы имеете дело с домашней сетью или сетью небольшого предприятия, которая не обладает собственным доменным именем, вам не потребуется формировать какой-либо сложный сетевой дизайн. Простая сетевая схема, которую можно использовать в подобном случае, показана на рис. 18.1. Между внутренней сетью и Интернетом располагается простой бастионный узел, который обеспечивает также маскировку IP. Сетевая карта eth0 соединенного с Интернетом узла HostA подключается к концентратору, к которому подключены также все

остальные системы внутренней сети. Адрес интерфейса eth0 бастионного узла (192.168.0.1) для всех остальных систем является адресом шлюза по умолчанию. Если в Интернете узел HostA обладает статическим IP-адресом, он будет использовать в качестве шлюза шлюзовую систему интернет-провайдера. Если узлу HostA выделяется динамический IP-адрес, значит, в конфигурационных файлах этого узла будут содержаться параметры ipdefault и defaultroute, которые будут настроены таким образом, чтобы обеспечить динамическую маршрутизацию через интерфейс pppO в случае, если возникает соединение.

СОВЕТ

В Linux интерфейсы именуются в соответствии со следующим соглашением об именовании:

Ethernet: eth
Token Ring: tr
Point-to-Point (аналоговый модем): ppp
Point-to-Point (ISDN): ippp

Самый первый экземпляр интерфейса получает номер 0 (ноль), а все остальные интерфейсы нумеруются по порядку. Таким образом, если у в системе есть два интерфейса Ethernet, они будут обозначены eth0 и eth1. Если в системе есть один интерфейс Ethernet и один интерфейс ppp, они будут обозначены eth0 и ppp0. Это соглашение используется в данном тексте для обозначений. В качестве шлюза внутренние системы должны содержать в своей конфигурации только лишь адрес интерфейса etn0 бастионного узла.

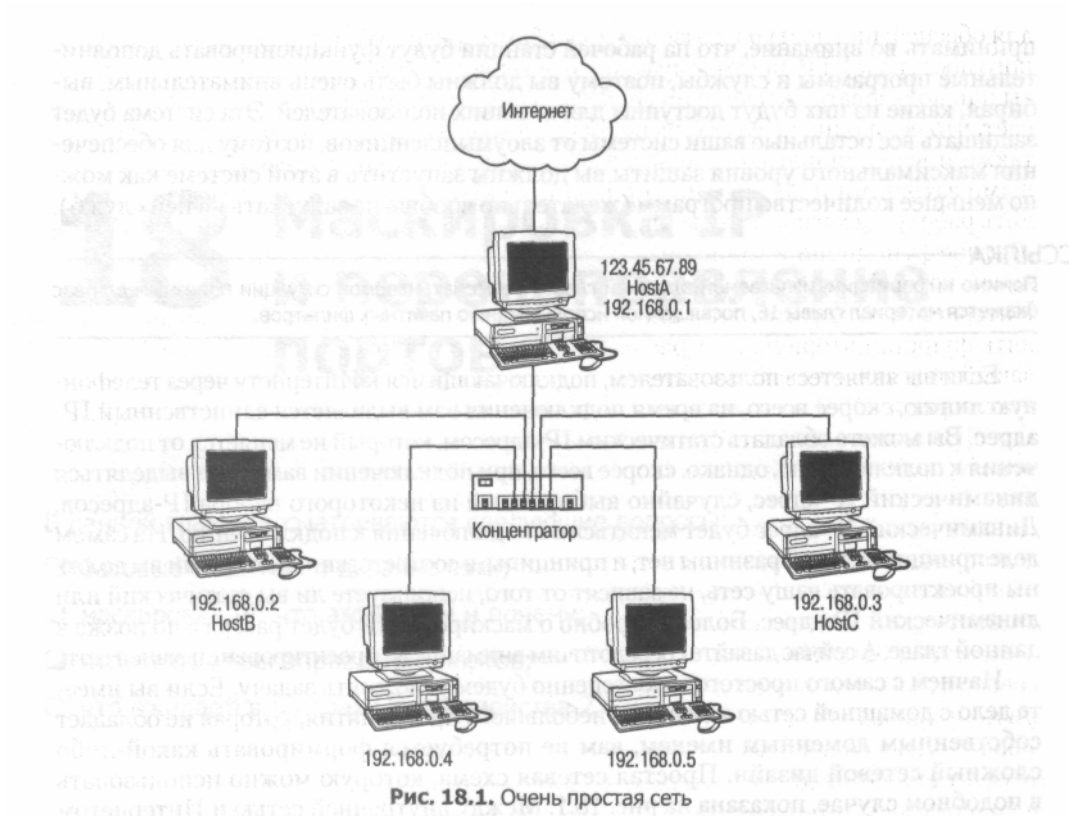

Рис. 18.1. Очень простая сеть

Если вы имеете дело с более крупной сетью, соединенной с Интернетом, при этом обладаете собственным доменным именем и предлагаете внешним пользователям свои собственные службы, вы должны изменить некоторые аспекты взаимодействия с Интернетом. Как правило, для крупных предприятий используется несколько IP-адресов. Многие интернет-провайдеры предоставляют своим клиентам подсети, включающие в себя восемь IP-адресов. При этом вы получаете в свое распоряжение пять эффективных IP-адресов, которые вы можете использовать для идентификации систем, обеспечивающих доступ к предлагаемым вами службам. Почему пять? Вспомним базовые сведения об адресации в сетях IP. Предположим, что вы получили в свое распоряжение подсеть 123.45.67.80/29 (сетевая маска 255.255.255.248). Другими словами, у вас есть IP-адреса начиная от 123.45.67.80 и заканчивая 123.45.67.87. Самый первый из них, который заканчивается на 80, является адресом сети, его нельзя использовать для идентификации сетевых узлов. Последний адрес диапазона, который заканчивается на 87, является широковещательным адресом. Этот адрес также нельзя использовать для идентификации сетевых узлов. Остается шесть IP-адресов. Однако один из этих шести адресов должен использоваться для идентификации интерфейса маршрутизатора, который связывает вашу сеть с Интернетом и принадлежит вашему провайдеру. Таким образом, у вас остается только пять свободных IP-адресов.

Один IP-адрес необходимо присвоить вашему брандмауэру или узлу, выполняющему маскировку IP. Остается достаточное количество адресов, необходимых для обеспечения доступа к базовым службам: серверу Web (http), серверу анонимного доступа к FTP, почтовому серверу и серверу DNS. Вы можете

снизить затраты и использовать только один сервер, сделав его многоцелевым, однако это означает, что если одна из служб окажется взломанной, под угрозой взлома окажутся также и все остальные службы. Вам приходится выбирать между дополнительными затратами и повышенным риском. Если вы можете позволить себе остановить работу всех ваших служб на несколько дней, пока вы не восстановите их корректное функционирование, значит, службы, которые вы предлагаете, не стоят тех денег, которые вы должны выложить за обеспечение непрерывного соединения с Интернетом. Обратите внимание на рис. 18.2. Сервер FTP (HostFTP) может выполнять функции вторичного сервера DNS. Однако принимая во внимание размеры вашей подсети, для вас, скорее всего, будет удобнее возложить функции вторичного сервера DNS на вашего интернет-провайдера. В этом случае даже если ваш маршрутизатор прекратит функционирование, весь остальной мир все равно сможет использовать принадлежащие вам доменные имена до тех пор, пока вы не восстановите функционирование (или не истечет время актуальности записей вторичного DNS). То же самое относится и к вашему узлу sendmail (HostMTA). Будет лучше, если функции резервной почтовой службы будет выполнять ваш провайдер, который будет принимать вашу почту до тех пор, пока ваша сеть или ваш сервер MTA (HostMTA) не восстановят свою работу.

Система HostWWW — это ваш web-сервер. Во многих случаях нагрузка на web-серверы является относительно небольшой, если только вы не используете каких-либо приложений баз данных и т. п. Однако, учитывая повышенную опасность таких технологий, как CGI, будет лучше, если вы будете эксплуатировать эту систему отдельно от других. Наконец, один из компьютеров выполняет функции первичного DNS-сервера (HostDNS). Приятной особенностью данной системы является то обстоятельство, что ее можно использовать для хранения копий конфигурационных файлов, файлов данных и бинарных файлов, благодаря чему в любой момент за короткое время вы сможете превратить этот компьютер в любую другую систему. Подобный резерв чрезвычайно полезен в случае, если одна из ваших систем выходит из строя. В случае выхода из строя узла HostMTA, если узел HostDNS указан в качестве резервного адреса MX, вам не потребуется делать вообще ничего, за исключением, конечно же, процедур, необходимых для восстановления функционирования вашего основного почтового сервера. Для всех остальных систем (за исключением брандмауэра) просто назначьте IP-псевдоним ethO как ethO:l *с* указанием адреса неисправного компьютера (HostFTP или HostWWW), запустите службу, и вы можете продолжать бизнес как обычно, до тех пор пока неисправная система снова вступит в строй.

Сети, о которых рассказывается в данной главе, можно разделить на три части. Первая часть — это собственно Интернет. Интернет начинается на дальнем от вас интерфейсе маршрутизатора вашего интернет-провайдера (на самом деле этот маршрутизатор может принадлежать и вам, однако в данной книге подразумевается, что он принадлежит вашему провайдеру). Вы не можете контролировать какие-либо аспекты Интернета. На ближней к вам стороне маршрутизатора начинается зона DMZ (Demilitarized Zone). Термин позаимствован из военного жаргона. Зона DMZ включает в себя подконтрольные вам ресурсы и является буфером, отделяющим вашу внутреннюю сеть от безжалостного Интернета. В зоне

DMZ вы обладаете лишь ограниченным контролем, однако, обнаружив злоумышленников, вы можете либо оборвать их соединения, либо отказать в доступе, исходя из их IP-адресов. Зона DMZ заканчивается у внешнего интерфейса вашего брандмауэра (по крайней мере вы на это надеетесь). За внутренним интерфейсом брандмауэра начинается внутренняя сеть, которую называют «доверенной» (trusted). Иногда термин «доверенная сеть» не является в полной мере точным, так как многие компании и организации обладают внутренними сетями, которым нельзя доверять. Это определяется не тем, насколько хорошо брандмауэр выполняет свои функции, а тем, насколько много пользователей работает во внутренней сети и каков характер взаимодействия этих пользователей с сетью. Вы полностью контролируете (по крайней мере теоретически, если, конечно, брандмауэр хорошо справляется со своими функциями), кто именно обладает доступом к этому разделу сети, куда разрешается обращаться тем или иным пользователям и что они могут делать.

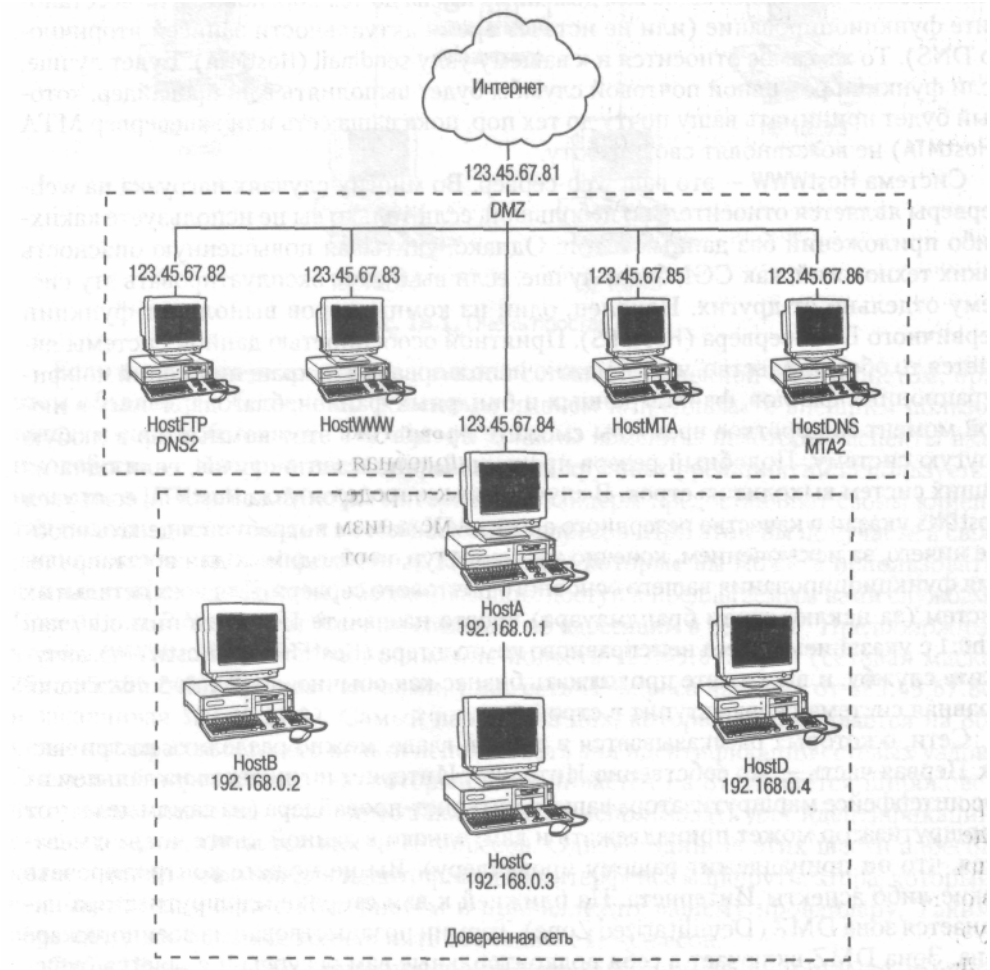

Рис. 18.2. Сеть небольшой компании с публично-доступными службами в зоне DMZ

Каждая подключенная к Интернету система, расположенная в DMZ, должна быть настолько самодостаточной, насколько это возможно. Ни одна из этих систем не должна использовать NFS. Подключение к этим системам обычных пользователей должно быть запрещено; благодаря этому вы получите стабильную контролируемую рабочую среду и в случае, если одна из систем окажется взломанной, вы можете не беспокоиться о важных пользовательских данных. Каждая система должна обладать своим собственным набором паролей. Благодаря этому *в* случае, если злоумышленник завладеет каким-либо из паролей, это не будет означать, что он автоматически получает доступ ко всем системам. По тем же самым причинам не следует использовать NFS или какие-либо другие ресурсы общего доступа. Наконец, сетевой узел, выполняющий функции брандмауэра, не должен использоваться для запуска каких-либо служб и не должен быть отмечен в базе данных DNS. Так этот узел сложнее будет использовать в качестве цели нападения. Если вы внесете сведения о брандмауэре в базу данных DNS, вы привлечете к нему лишнее внимание, взломщик сможет понять, что в сети есть такая система, он сможет попытаться обнаружить ее. В наши дни в Интернете, наверное, можно обнаружить пару-другую параноидальных FTP-серверов, которые отказываются обслуживать системы, в отношении которых невозможно выполнить обратное сопоставление доменного имени, однако сейчас это является скорее исключением, чем правилом.

Последняя сетевая конфигурация, которую мы с вами рассмотрим в данной главе, является несколько

более сложной. В рамках этой конфигурации зона DMZ частично перемещается за брандмауэр. Схема такой сети показана на рис. 18.3. Когда брандмауэр получает запрос через любой из портов 21 (FTP), 25 (MTA), 53 (DNS) или 80 (HTTP), он производит перезапись заголовков пакетов и перенаправляет пакеты соответствующему серверу. Опасность такой схемы состоит в том, что вы разрешаете злоумышленникам проникновение за брандмауэр, благодаря чему они могут попробовать «прощупать» службы вашей внутренней сети на предмет наличия уязвимых мест. Однако подобная схема обладает и преимуществом: она позволяет избежать сетевых атак определенных типов. Например, если злоумышленник пытается использовать механизм CGI для того, чтобы запустить на web-сервере процесс login через другой порт, этот порт будет недоступен для него, попытка взлома окажется безуспешной. Однако не стоит забывать, что атакующий, который пытается осуществить нападение через перенаправленный порт, обладает доступом к вашей внутренней сети. Исходя из этого рекомендуется разместить все ваши рабочие станции, а также файловые серверы и серверы печати в одной подсети, а серверы, обслуживающие клиентов из Интернета, — в другой подсети. Не следует слишком упрощать задачу злоумышленникам. В результате такого деления сети на две подсети вы сможете обеспечить раздельное функционирование сетевых служб. Более подробно о подобной сетевой конфигурации будет рассказано далее. В отношении интернет-серверов, расположенных под защитой брандмауэра, следует предпринять такие же меры безопасности, как и в отношении интернет-серверов, расположенных вне зоны, защищаемой брандмауэром: вы должны изолировать эти серверы друг от Друга и от других систем, вы должны снабдить их индивидуальными учетными записями и паролями, вы должны запретить подключение к ним обычных пользователей и отказаться от любого совместного использования ресурсов (несмотря на то, что последний пункт в данной конфигурации менее проблематичен).

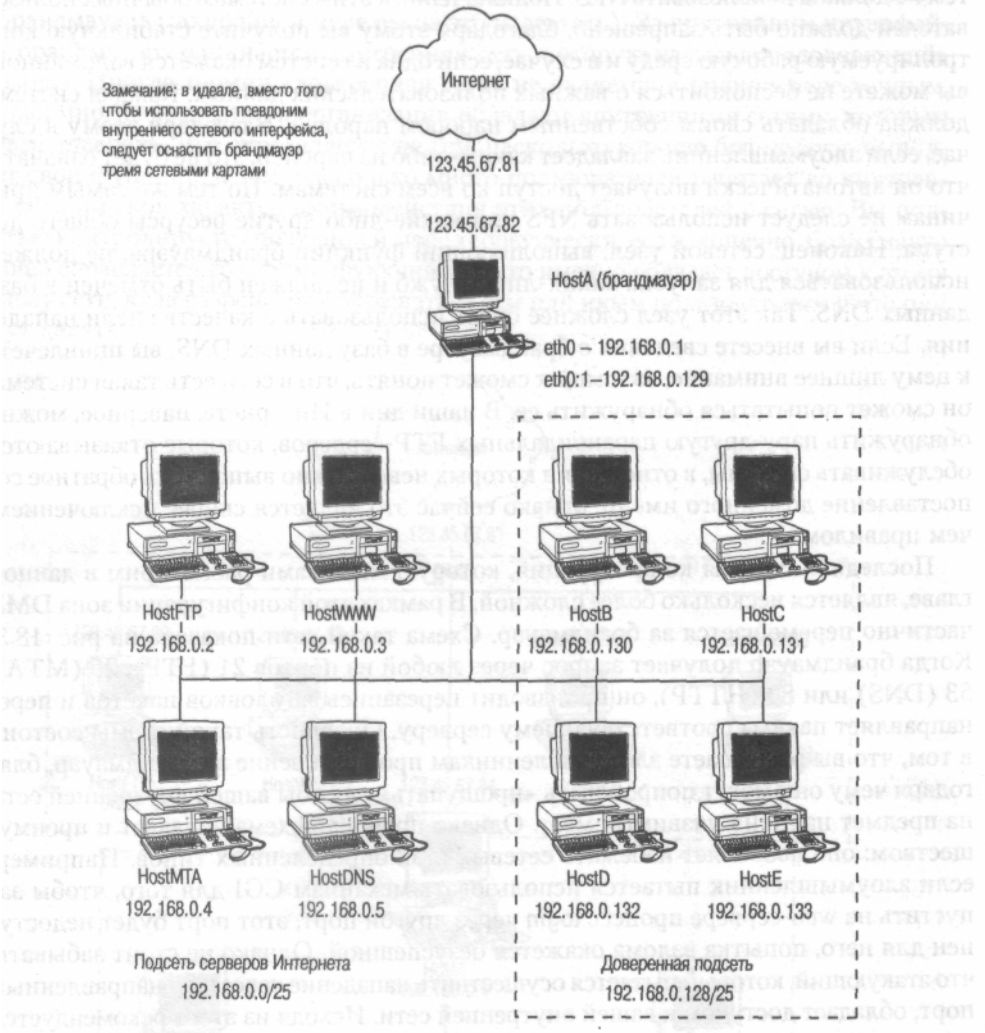

Рис. 18.3. Сеть небольшой компании, в которой службы Интернета работают внутри сети, защищенной брандмауэром

Другие соображения

Когда вы внедряете в своей сети механизм маскировки IP, некоторые процессы начинают выполняться иначе, а некоторые процессы вообще перестают выполняться. Например, находясь во

внутренней сети, вы по-прежнему можете просматривать Web и обращаться к системам, расположенным по ту сторону от брандмауэра, однако внешние пользователи Web не смогут увидеть ни одну из систем, расположенных в вашей внутренней сети.

ПРИМЕЧАНИЕ

Как правило, маскировка IP применяется для того, чтобы обеспечить использование в вашей внутренней сети одного из частных диапазонов IP-адресов. Частный диапазон IP-адресов — это специально зарезервированный диапазон, в который входят адреса, не предназначенные для маршрутизации через Интернет. Ни один из маршрутизаторов Интернета не будет пропускать пакеты, ассоциированные с IP-адресами из частного диапазона. Однако маскировку IP можно использовать не только совместно с частными диапазонами IP-адресов. Вы можете маскировать и маршрутизируемые IP-адреса. Все же концепция IP-маскировки в первую очередь разработана для того, чтобы расширить доступное адресное пространство IP.

Более того, до тех пор пока вы не настроите все должным образом, пользователи внутри сети, пытающиеся использовать стандартного клиента FTP (то есть не того, который встроен в Netscape), не смогут загружать файлы из Интернета. Это происходит потому, что стандартные клиенты FTP используют активный режим FTP. Для соединения с сервером FTP используется стандартный порт 21. Сервер отвечает через случайно выбранный на стадии установления соединения порт с номером выше 1024. Вы получаете возможность подключиться и даже перемещаться по каталогам. Однако как только вы попытаетесь получить содержание каталога или загрузить файл, имя и местоположение которого вам известно, вы получаете сообщение об ошибке. Причина в том, что по умолчанию сервер FTP открывает второй канал — канал данных — через порт 20 и пытается соединиться с клиентом для того, чтобы передать ему данные. (Кстати говоря, именно по этой причине серверу FTP можно приказать передать интересующий вас файл третьему лицу — изначальное соединение не используется, устанавливается второе соединение, которое может быть перенаправлено на любую другую систему, с которой может соединиться FTP-сервер, если только FTP-сервер, в частности такой, как в OpenLinux, не запрещает этого делать.)

ПРИМЕЧАНИЕ

Маскировка IP и proxy — это две разных концепции, которые можно использовать как по отдельности, так и совместно — в комбинации.

ССЫЛКА

Более подробно о proxy рассказывается в главе 17.

Проблема, которая возникает у FTP-сервера, пытающегося соединиться с обращающимся к нему клиентом, проиллюстрирована на рис. 18.4. Клиент соединяется с сервером через порт 21. Клиент (HostB с IP-адресом 192.168.0.2) соединяется с FTP-сервером в Интернете. Однако чтобы осуществить это, клиент HostB должен преодолеть шлюз HostA с IP-адресом 192.168.0.1. Когда узел HostA принимает пакеты, предназначенный для FTP-сервера, он маскирует эти пакеты таким образом, будто бы они исходят из интерфейса, связывающего узел HostA с Интернетом. Иными словами, в качестве адреса-источника пакетов подставляется IP-адрес 123.45.67.89 (адрес узла HostA в Интернете). Получив запрос на соединение, сервер FTP думает, что этот запрос исходит от узла с адресом 123.45.67.89 через порт 21. Таким образом, когда наступает время открыть второй канал, то есть канал данных, сервер пытается соединиться с клиентом по адресу 123.45.67.89 через порт 20. Однако узел HostA, которому принадлежит этот адрес, не ожидает поступления каких-либо соединений через порт 20, поэтому он отклоняет запрос на соединение. А в это время клиент, расположенный за брандмауэром, продолжает ожидание входящего соединения через порт 20, однако запрос на это соединение до него никогда не дойдет. Это потому, что узел HostA и не подозревает о том, что узел HostB ожидает входящего соединения.

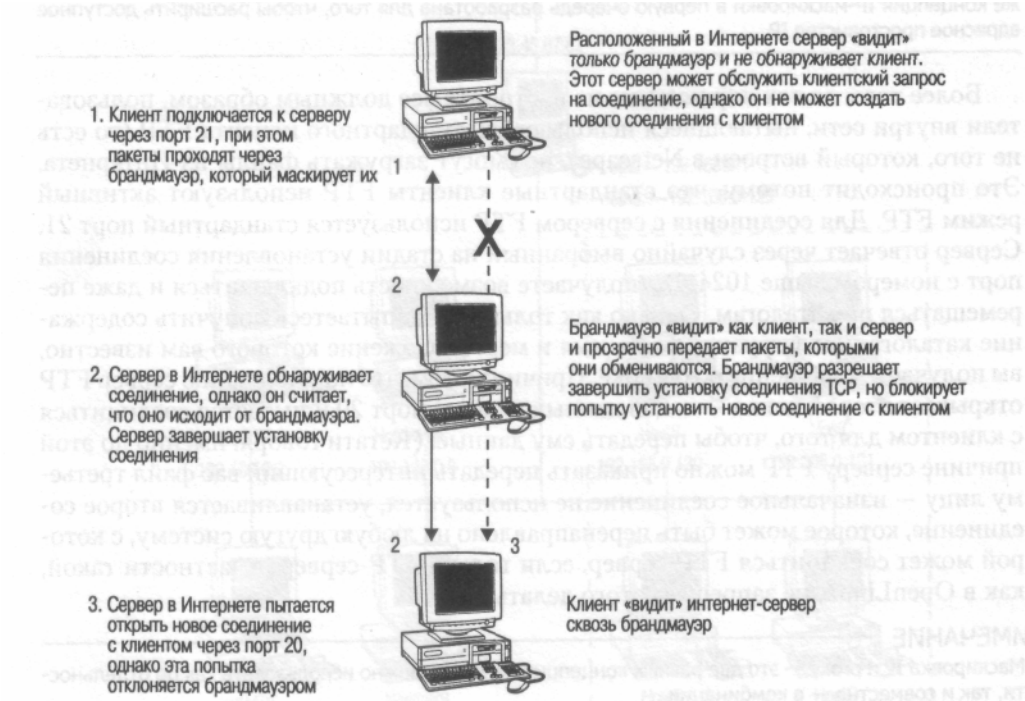

Рис. 18.4. Почему активный режим FTP не работает через брандмауэр

К счастью, разработчики Linux хорошо знают об этой проблеме. Вы можете воспользоваться одним из двух решений. Первое решение предусматривает использование пассивного режима работы FTP. Такой режим используется браузером Netscape Navigator, который позволяет загружать файлы с FTP-сервера, в то время как обычный FTP-клиент не может этого сделать. При использовании пассивного режима FTP сервер не пытается соединиться с клиентом через отдельный канал данных, вместо этого он терпеливо ждет, пока клиент откроет соединение через изначальный канал команд (порт 21). Второе решение является более элегантным и позволяет использовать FTP через брандмауэр в активном режиме. Для этого необходимо убедиться, что ваше ядро собрано с включением в него специальных модулей маскировки IP, содержащихся в разделе Networking Options (сетевые параметры). В листинге 18.1 показана конфигурация ядра, а в листинге 18.2 -перечень модулей, собранных в соответствии с данным списком.

ССЫЛКА

Более подробно о сборке ядра Linux с поддержкой ipchains рассказывается в *главе 16.*

Листинг 18.1. Конфигурация ядра для модулей маскировки IP

```
CONFIG_IP_MASQUERADE_ICMP=y
CONFIG_IP_MASQUERADE_MOD=y
CONFIG_IP_MASQUERADE_IPAUTOFW=m
CONFIG_IP_MASQUERADE_IPPORTFW=m
CONFIG_IP_MASQUERADE_MFW=m
```

В листинге 18.1 показано содержимое файла /usr/src/linux/.config после того, как этот файл настроен для маскировки IP. Первая строка добавляет в маскирующий раздел ядра код, который позволяет передачу пакетов ICMP. По умолчанию ядро передает только TCP и UDP. Протокол ICMP является самостоятельным протоколом, и его передача через сеть не зависит от IP, поэтому, включив в конфигурацию ядра данную строку, вы тем самым разрешаете распознавание заголовков ICMP и передачу соответствующих пакетов через сеть. Без этого пакеты ping, равно как и пакеты tracerout, отправляемые системами Microsoft, не будут передаваться через данный узел (протокол ICMP используется для tracerout только на системах Microsoft, все остальные системы используют для этой цели UDP, таким образом, если вы хотите обеспечить работу tracerout только для систем, отличных от Microsoft, вы можете не включать ICMP).

Вторая строка формирует набор модулей. Все эти модули, а также модули, соответствующие последним трем строкам листинга 18.1, перечислены в листинге 18.2. В результате добавления в конфигурацию этих трех строк формируются модули ip_masq_autofw.o (для протоколов, не отмеченных специально), ip_masq_ portfw.o (для обеспечения работы перенаправления портов) и ip_masq_mfw.o (для обеспечения работы перенаправления пометок ipchains). Последний модуль в файле электронной помощи ядра ошибочно называется именем ip_masq_markfw.

Эти модули не загружаются автоматически, как это происходит с другими модулями. Если вы

намерены воспользоваться предлагаемыми ими функциями, вы должны загрузить их либо во время начальной загрузки, либо вручную. В системе OpenLinux чтобы выполнить загрузку модулей в процессе начальной загрузки операционной системы, необходимо добавить их имена в файл /etc/modules/ default (при этом не следует добавлять к имени окончание .o). Вы также должны убедиться в том, что порт 20 не блокирован. Теперь, если обеспечить загрузку модуля ip_masq_ftp, ваш маскирующий брандмауэр сможет обнаружить, что FTP работает в активном режиме, в результате обратное соединение сервера с клиентом будет пропущено через брандмауэр во внутреннюю сеть. Как можно заметить, существуют аналогичные модули, предназначенные для пропуска во внутреннюю сеть соединений CUSeeMe, IRC, QUAKE, RealAUDIO, ViDeO LIVE и других специальных пользовательских протоколов. Ядро распознает соединения между сервером и клиентом на основе IP-адресов, портов и порядковых номеров пакетов. Модули не загружаются автоматически, однако их можно загрузить в процессе начальной загрузки ОС. В разных комплектах Linux для этой цели могут использоваться разные конфигурационные файлы. Например, в Debian для этой цели используется файл /etc/modules, в котором содержится список модулей, которые необходимо загрузить. В комплекте Caldera OpenLinux 2.3 имена модулей следует добавить в файл /etc/modules/default. В других комплектах имя и местоположение подобного файла может быть иным — вы должны свериться с электронной документацией вашего комплекта. Вместо того чтобы загружать модули в процессе начальной загрузки, вы можете выполнить загрузку модуля вручную из командной строки. Сделать это можно непосредственно перед тем, как возникнет необходимость в использовании некоторого модуля. Загрузить модуль можно при помощи следующей команды:

```
modprobe <имя_модуля>
```

Как при загрузке из командной строки, так и при добавлении в конфигурационный файл при указании имен модулей не следует добавлять к ним суффикс .o.

Листинг 18.2. Модули маскировки

```
ip_masq_autofw.o
i p_masq_cuseeme.o
ip_masq_ftp.o
ip_masq_irc.o
ip_masq_mfw.o
ip_masq_portfw.o
ip_masq_quake.o
ip_masq_raudio.o
ip_masq_user.o
ip_masq_vdolive.o
```

ПРИМЕЧАНИЕ

При использовании ядра новой серии Linux 2.4.x (которое еще не было выпущено на момент написания данной книги) все эти модули вам не потребуются. В отличие от работающей совместно с ядрами 2.2.x программы ipchains новое программное обеспечение netfilter, обеспечивающее работу брандмауэра, не является программой пользовательского режима. Благодаря этому данная программа работает быстрее.

Маскировка IP

Если с маскировкой IP (эту технологию так же часто называют Network Address Translation, NAT) связано такое большое количество проблем, зачем тогда использовать данный механизм? Этому существует несколько достаточно весомых причин, не самой последней из которых является более высокий уровень защиты.

В настоящее время Интернет целиком и полностью основан на стандарте IPv4, и если в ближайшее время ничего не изменится, то всего через несколько лет все IP-адреса, которые допускается использовать для идентификации сетевых узлов в Интернете, будут исчерпаны. Как уже отмечалось ранее, пространство IP-адресов включает в себя ограниченное количество этих адресов. Общее количество всех возможных IP-адресов в рамках схемы адресации IPv4 несложно подсчитать. В настоящее время IP-адрес состоит из 32 битов, а следовательно, всего существует чуть больше 4 миллиардов уникальных IP-адресов. На практике использовать можно меньше половины из них. Количество доступных для использования IP-адресов существенно выросло в 1993 году, после того как основанная на классах A, B и C схема адресации была объявлена устаревшей и ей на смену пришла технология бесклассовой адресации CIDR (Classless Inter-Domain Routing). Технология CIDR позволила сетевым администраторам более продуктивно использовать диапазоны IP-адресов, которыми они обладали. Однако этого все равно недостаточно. Именно поэтому интернет-провайдеры и другие организации, занимающиеся распределением IP-адресов, хотят получить обоснование (а зачастую и более высокую ежемесячную плату) от компаний и частных лиц, желающих получить в свое распоряжение дополнительные IP-адреса.

Применяя в своей внутренней сети IP-адреса, которые не предназначены для маршрутизации через

Интернет, вы можете получить столько IP-адресов, сколько вам нужно для обеспечения работы сети. Все, что вам нужно сделать, — это выбрать один из диапазонов частных IP-адресов, предназначенных именно для этой цели.

Такие диапазоны определяются документом RFC 1918, в котором три крупных блока IP-адресов специально выделяются для частного использования. Первым таким блоком является диапазон 10.0/8 (то есть от 10.0.0.0 до 10.255.255.255), который (если использовать устаревшую основанную на классах схему адресации) соответствует полной сети класса A. Второй блок соответствует шестнадцати диапазонам класса B от 172.16.0.0 до 172.31.255.255. Последний блок включает в себя 256 диапазонов класса C от 192.168.0.0 до 192.168.255.255. Все эти адреса не предназначены для маршрутизации через Интернет (любой маршрутизатор Интернета отбрасывает все ассоциированные с ними пакеты), поэтому вы можете использовать их в любой ситуации, не опасаясь, что может возникнуть конфликт с какой-либо другой сетью.

Маскировка IP наделяет вас существенным преимуществом: после того как вы настроите IP-адреса в вашей внутренней сети, вам больше не придется менять адресацию ваших компьютеров. Например, когда вы меняете провайдера, вы должны сменить адреса всех компьютеров, расположенных во внешней зоне DMZ, а также внести в базу данных DNS соответствующие изменения (имеется в виду как база данных DNS, поддерживаемая InterNIC, так и файлы данных ваших внутренних DNS-серверов). Однако ваша внутренняя сеть может оставаться в неизменном виде. Вторым преимуществом маскировки IP является то обстоятельство, что если вы не используете перенаправления портов, никто не сможет проникнуть в вашу внутреннюю сеть извне, не преодолев предварительно брандмауэр. Это означает, что все ваши усилия по обеспечению безопасности можно сосредоточить всего на одном сетевом узле. Обеспечив должную защиту брандмауэра, вы обеспечите безопасность систем вашей внутренней сети.

В ядрах серии Linux 2.2.x маскировка IP осуществляется очень просто. После того как вы выбрали подходящую подсеть, достаточно ввести всего три коротких строчки, и брандмауэр начнет выполнять маскировку. Предположим, вы выбрали подсеть 192.168.0.0/24, при этом ваш брандмауэр напрямую подключен к Интернету. Чтобы обеспечить маскировку, необходимо ввести команды, показанные в листинге 18.3. Перед этим, естественно, необходимо убедиться в том, что все необходимые для маскировки модули загружены в память.

Листинг 18.3. Правила ipchains для маскировки сети 192.168.0.0/24

```
ipchains -P forward -j DENY
Ipchains -A forward -b -s 192/168/0/0/24 -d 0/0  -j MASQ
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Если вы уже ознакомились с материалом предыдущих глав, данный листинг будет для вас вполне понятным. Для тех, кто еще не знаком с правилами ipchains, поясню, что первая строка определяет политику брандмауэра: DENY (запретить). Если вы хотите, можете заменить эту политику на ACCEPT (принять), так как большинство пакетов никогда не доходит до применения политики по умолчанию. Перед политикой по умолчанию в цепочке правил ipchains, как правило, располагаются другие правила. Как только обнаруживается, что пакет соответствует одному из них, обработка цепочки прекращается. В нашем случае все пакеты должны удовлетворять правилу, которое определяется во второй строке. Это правило выполняет основную работу по передаче пакетов из сети в сеть. Оно принимает все пакеты из подсети 192.168.0.0 и перенаправляет их в Интернет. Команда -A добавляет это правило к остальным правилам в цепочке forward (на текущий момент другие правила в цепочке отсутствуют, а политика по умолчанию всегда является последним правилом в цепочке). Ключ -s указывает на адрес-источник пакетов, а ключ -d указывает на адрес-приемник. Данное правило должно действовать в обоих направлениях, поэтому вы должны либо добавить еще одно правило, в котором аргументы -s и -d поменяны местами, либо использовать ключ -b, который указывает на то, что правило применяется при передаче пакетов в обоих направлениях. Любой пакет, удовлетворяющий этому правилу, будет маскирован и передан далее по маршруту. Данное правило указывает на то, что маскировка должна осуществляться в обоих направлениях. Последняя строка листинга 18.3 включает перенаправление IP пакетов. По умолчанию в ядре Linux этот механизм выключен. В «файле» /proc/sys/net/ipv4/ip_forward содержится ноль (0), что означает отсутствие перенаправления, однако, заменив это значение на 1, вы включаете перенаправление IP. Чтобы включить перенаправление IP, необходимо использовать следующую команду:

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

Перенаправление портов

Вместо того чтобы располагать интернет-серверы в зоне DMZ, делая их доступными для разного рода атак, вы можете переместить их внутрь вашей доверенной сети. Механизм перенаправления портов

(port forwarding) позволяет прозрачно перенаправлять соединение, поступающее через порт вашего брандмауэра, в некоторый порт некоторой системы, расположенной во внутренней сети. Порт брандмауэра, через который поступило соединение, может отличаться по номеру от порта внутренней системы, в который это соединение перенаправляется. Мало того, разным портам брандмауэра можно поставить в соответствие разные сетевые узлы вашей внутренней сети.

Подход, предусматривающий перемещение служб внутрь вашей доверенной сети, обладает как преимуществами, так и недостатками. Недостаток заключается в том, что если злоумышленник найдет способ взломать внутреннюю систему, в отношении которой осуществляется перенаправление портов, значит, он сможет получить доступ к вашей внутренней сети. Однако осуществить атаку через брандмауэр несколько сложнее, чем взломать узел, расположенный в зоне DMZ. Преимуществ у этого подхода много. Во-первых, вы обеспечиваете более высокий уровень защиты, так как взломщик вынужден работать через перенаправленные порты. Если разные порты брандмауэра перенаправляются на разные внутренние сетевые узлы, взломщик не будет об этом знать и не сможет использовать один порт для получения доступа через другой порт (если, конечно, эти два порта перенаправляются на разные компьютеры). Если ваш интернет-провайдер обеспечивает вас только одним постоянным IP-адресом вместо того, чтобы выделить вам блок адресов, перенаправление портов позволяет вам использовать для обслуживания внешних пользователей Интернета несколько внутренних систем вместо того, чтобы запускать интернет-службы прямо на брандмауэре (что является очень плохой идеей).

В комплекте OpenLinux перенаправление портов также выполняется очень просто, однако, к сожалению, пакет Caldera, прилагаемый к данной книге, не включает в себя средств, необходимых для того, чтобы организовать с его помощью перенаправление портов. Для этой цели вам потребуется найти и установить пакет ipmasqadm. Как только вы установите этот пакет, вы убедитесь, что перенаправление портов организуется не сложнее, чем маскировка IP.

ПРИМЕЧАНИЕ

На компакт-диске, прилагаемом к данной книге, содержится каталог col/security/, в котором содержится RPM-пакет ipmasqadm.

Чтобы использовать перенаправление портов, вы не обязаны использовать маскировку — однако в этом случае адрес, на который вы перенаправляете, должен быть корректным IP-адресом Интернета, а ваш шлюз должен работать как маршрутизатор, то есть он не должен выполнять маскировку IP.

Вы должны убедиться в том, что модуль ip_masq_portfw загружен либо в процессе начальной загрузки системы (имя модуля указано в файле /etc/modules/ default) или вручную, при помощи modprobe. Имейте в виду, что этот модуль не загружается автоматически только потому, что вы обращаетесь к ipmasqadm.

Убедившись, что модуль загружен, вы можете приступать к перенаправлению портов, введите необходимые правила ipchains, затем укажите правила маскировки IP. Наконец, после этого примените правила перенаправления портов. Необходимые команды показаны в листинге 18.4. После этого убедитесь, что вы включили перенаправление IP в ядре.

СОВЕТ

Ели вы загрузили все необходимые модули, однако ваш брандмауэр все равно не перенаправляет пакеты, чаще всего это происходит потому, что файл /proc/sys/net/ipv4/ip_forward по-прежнему содержит 0, запрещая тем самым перенаправление. Содержимое этого файла всегда следует проверять в первую очередь.

Листинг 18.4. Примеры правил перенаправления портов

```
ipchains -P forward -j ACCEPT
ipchains -A forward -b -s 0/0 -d 192.168.0.0/24 -j MASQ
ipmasqadm portfw -a  -P tcp -L 123.45.67.89 80 -R 192.168.0.2 80
ipmasqadm portfw -a  -P udp -L 123.45.67.89 53 -R 192.168.0.3 53
ipmasqadm portfw -a -P tcp -L 123.45.67.89 53 -F 192.168.0.3 53
 echo 1 > /proc/sys/net/ipv4/ip_forward
```

Первая строка листинга 18.4 устанавливает правило по умолчанию для цепочки forward равным ACCEPT (принять). Вторая строка обеспечивает маскировку IP (маскировка необходима, так как мы используем частные IP-адреса, в противном случае использовать это правило не нужно). Если вам не нужно второе правило, так как во внутренней сети вы используете «живые» маршрутизируемые IP-адреса Интернета, значит, вам необходимо первое правило (forward ACCEPT). При желании вы можете использовать политику DENY (запретить), однако в этом случае вам потребуются правила, в соответствии с которыми будут приниматься (ACCEPT) все пакеты через порты, которые вы перенаправляете.

В следующих трех строках определяются правила ipmasqadm. Обратите внимание на две команды, которые перенаправляют пакеты для службы DNS. Можно видеть, что в листинге присутствуют две

отдельные строки: для TCP и для UDP. Протокол DNS использует как UDP, так и TCP, поэтому, составляя правила ipmasqadm, необходимо упомянуть все используемые протоколы. Если вы не обладаете возможностью обмена данными с Интернетом (и поэтому не можете обеспечить разрешение доменных имен в процессе начальной загрузки), то в качестве последнего аргумента вы можете добавить ключ -n, который указывает на то, что следует использовать IP-адреса в численном виде. Наконец, следует включить перенаправление IP в ядре.

Команда ipmasqadm обладает пятью параметрами и шестью возможными аргументами. К параметрам этой команды относятся: a *(add* — добавить), d *(delete* -удалить), f *(flush* — очистить), l *(list* — вывести список правил), n *(numeric* — ввод в виде чисел). Параметр f не требует аргументов. Параметр l может принимать в качестве аргумента только -n. Ключ -n может использоваться с любыми дурги-ми параметрами, за исключением -f.

Указав параметры -a или -Ь следует указать за ними протокол в форме -P <про-токол>, затем -L и локальный IP-адрес и порт, куда будет приниматься трафик, а затем -R и удаленный IP-адрес и порт, куда этот трафик будет перенаправлен. Если вы хотите добавить несколько разных серверов (например, для того чтобы обеспечить распределение нагрузки), вы можете указать несколько правил, например для того, чтобы принимать трафик через порт 80 и перенаправлять его нескольким разным серверам. Следует использовать формат -p # (то есть строчная буква p, за которой следует число). Таким образом обеспечивается примитивная форма управления нагрузкой: # — это количество соединений, после достижения которого будет использовано следующее правило перенаправления для указанного локального IP-адреса и порта.

Ядро Linux 2.4.x

В ядрах нового семейства Linux 2.4.x механизмы маскировки IP и перенаправления портов являются частью программного средства netfilter. Новый программный пакет netfilter является модульным, в нем код брандмауэра отделен от кода IP-маскировки и кода перенаправления портов. К счастью, полный набор всех правил, в соответствии с которыми функционирует netfilter, можно узнать при помощи команды iptables -L. По этой команде вы увидите на экране правила перенаправления портов, правила маскировки, а также правила брандмауэра. Программный инструмент, осуществляющий перенаправление портов, носит название ipnatctl. При помощи этого средства реализуются все формы NAT, при этом вы должны загрузить специальные отдельные программные модули NAT.

Все эти изменения стали результатом путаницы, связанной с ipchains, а также невозможностью перенаправления портов на удаленные системы (в составе ipchains присутствует лишь механизм перенаправления портов внутри локальной системы). Все эти проблемы теперь решаются с использованием ipnatctl.

Прежде всего вы должны убедиться в том, что все необходимые вам модули загружены в память. Это будет выполнено автоматически в результате выполнения команды depmod -a после того, как модули будут скомпонованы и установлены. На момент написания данной книги механизм автоматической загрузки модулей работает не полностью, однако в момент выхода в свет ядра 2.4.x, скорее всего, все будет работать как надо. Модули netfilter еще не интегрированы в ядро, поэтому вы по-прежнему должны выполнить два различных действия: компоновка и установка netfilter, а затем запуск depmod. В будущем это, скорее всего, будет выполняться удобнее.

Если вы хотите загрузить модули вручную, запустите depmod -a а затем используйте modprobe для того, чтобы выполнить необходимые операции загрузки. Вам потребуется загрузить следующие модули: ip_nat_map_masquerade, ip_nat_ftp (для поддержки соединений FTP), ip_state, ip_defrag и ip_conntrack_ftp. Помимо этих будут загружены также и другие модули.

Формат использования ipnatctl не сильно отличается от формата ipchains. Запустив ipnatctl без аргументов, вы получите краткую справку об использовании этой программы. То же самое произойдет в случае, если вы укажете неправильный аргумент. Используя ipnatctl, вы можете вставлять (-I) или удалять (-D) любые интересующие вас правила. Вы также можете очистить (-F) список правил или отобразить его на экране (-L). При вводе любых параметров допускается использовать только численные аргументы (-n), при этом ipnatctl не будет осуществлять разрешение имен сетевых узлов или служб.

Сообщив ipnatctl, что, собственно, вы собираетесь делать (вставлять или удалять), вы должны указать один из нескольких параметров. Для маскировки используется следующее правило:

```
ipnatctl -I -o eth0 -b source -m masquerade
```

Ключ -o указывает на интерфейс вывода, в отношении которого будет применяться данное правило. Этот ключ связан со следующим параметром: -b source. Ключ -b означает *binding* — связывание. Здесь необходимо указать один из двух аргументов: source (источник) или destination (приемник). При помощи

ключа -b вы указываете, хотите ли вы осуществлять перезапись адреса в момент, когда пакет прибывает или когда он покидает ваш маскирующий узел. Аргумент source указывает на то, что модуль ip_nat должен перезаписывать адрес в момент, когда пакет покидает узел, а аргумент destination предписывает перезаписать адрес в момент, когда пакет прибывает на узел. Приведенное ранее правило осуществляет маскировку исходящих пакетов, поэтому перезапись адреса следует выполнять в момент, когда пакет покидает узел. Ключ -o <устройство> следует использовать только совместно с ключом -b source, точно так же как ключ -i <устройство> (идентифицирующий входной интерфейс) следует использовать только совместно с ключом -b destination. Последним аргументом является -m masquerade, который указывает на специальное отображение адресов — маскировку IP. Вместе с ключом -m можно использовать один из четырех параметров: msaquerade, redirect, null и static. На данный момент доступны только три из них: msaquerade, redirect и static. К моменту выхода в свет новой версии ядра станет доступным и последний: null. Как можно заметить, данный формат существенно проще, чем формат правил маскировки, используемый в ipchains.

Для перенаправления портов используется несколько другой формат. При реализации перенаправления портов с использованием ipnatctl аргументы больше напоминают формат ipchains, чем рассмотренный ранее синтаксис NAT. Например, если вы хотите перенаправить все соединения, входящие через порт 80, в порт 8080 другой системы, необходимо использовать следующее правило:

```
ipnatctl -I -p tcp -s 0/0 --sport 80 -d 192.168.0.5 --dport 8080 -b dest
```

В данном примере вы ожидаете поступления любых соединений из Интернета (0/0) через входящий порт 80. Пакеты будут перенаправлены по адресу 192.168.0.5 в порт 8080. В данном случае вместо входного и выходного интерфейса указываются адрес-источник и адрес-приемник. Так как вы используете расширенные спецификации -sport и --dport, вы должны указать параметр protocol (-p). Таким образом, если вы осуществляете перенаправление портов для DNS, вам потребуется определить два правила: одно для TCP, а другое — для UDP. Опять же необходимо использовать ключ -b, и так как производится перезапись прибывающих пакетов, необходимо указать -Ь destination.

Приведенное ранее правило перенаправления портов можно переписать следующим образом:

```
ipnatctl -I -p tcp -i eth0 --sport 80 -d 192.168.0.5 --dport 8080 -b dest -m redirect
```

Ключ -i eth0 и ключ -s 0/0 абсолютно идентичны. Кроме того, в данном правиле используется отображение redirect. Благодаря этому используется специальный модуль ip_nat_map_redirect.

Правило маскировки, приведенное ранее, не требует реверсивного правила (как это было необходимо при использовании ipchains) для того, чтобы обеспечить возврат пакетов. Эту проблему решает модуль ip_conntrack *(IP connection tracking* — слежение за соединениями IP). Применение правил netfilter для маскировки IP (NAT) существенно упростит администрирование.

После того как вы введете рассмотренные ранее правила в систему, по команде ipnatctl на экран будет выведен список, показанный в листинге 18.5.

Листинг 18.5. Пример вывода команды ipnatctl -L

```
masquerade  [SRC]  0.0.0.0/0->0.0.0.0/0 tap0T0:
generic [DST]  0.0.0.0/32->192.168.0.5/32 proto=6 srcpt=80 dstpt=8080 TO:
redirect  [DST]  0.0.0.0/0->192.168.0.5/32 eth0proto=6 srcpt=80 dstpt=8080 TO:
```

Обратите внимание на различие между вторым и третьим правилом. Во втором правиле осуществляется *перенаправление портов* (port forwarding) на адрес 192.168.0.5 в порт 8080 всех пакетов, поступающих с адреса 0/0 через порт 80. В третьем правиле осуществляется *перенаправление пакетов* (redirecting), поступающих через интерфейс eth0 через порт 80 на адрес 192.168.0.5 в порт 8080. Два эти правила обеспечивают один и тот же эффект, однако третье правило выглядит более понятным.

Заключение

В данной главе я рассказал о некоторых базовых приемах проектирования сетей. Первый из рассмотренных приемов хорошо подходит для организации работы простой домашней сети или сети небольшого предприятия. Второй подход является более приемлемым для разнопланового взаимодействия с Интернетом: он предусматривает размещение интернет-серверов в зоне DMZ, которая располагается вне границ внутренней сети, на внешней стороне относительно брандмауэра. Последний рассмотренный подход предусматривает размещение интернет-серверов во внутренней сети, на внутренней стороне относительно брандмауэра, в этом случае обмен данными с Интернетом осуществляется с использованием перенаправления портов.

После этого я рассказал вам о технологии маскировки IP. Я рассмотрел, зачем нужна эта технология и как ее использовать. После этого вы узнали, как при помощи механизма перенаправления портов можно

обеспечить внешним пользователям Web доступ к серверам, расположенным на внутренней стороне вашего брандмауэра. Я рассказал вам о преимуществах и недостатках данного подхода.

В конце главы я дал вам представление о том, какие изменения будут внесены в ядра Linux серии 2.4.x.

19 Безопасность Samba

В данной главе рассматриваются следующие вопросы:
- настройка swat;
- запуск swat с использованием inetd;
- запуск swat с использованием web-сервера;
- сетевая система Samba;
- что такое ресурсы общего доступа Samba;
- проблемы защиты Samba.

В наши дни большинство сетевых рабочих сред представляет собой смесь операционных систем. Как правило, в большинстве сетей небольших и средних компаний используют центральный сервер Microsoft Windows NT, рабочие станции Windows, а также, возможно, другие серверные или сетевые операционные системы, такие как Banyan VINES, Novell Netware или какую-либо разновидность Unix. Чтобы обеспечить взаимодействие с этими системами, программисты Linux были вынуждены преодолеть множество препятствий. Компания Microsoft избегает поддержки NFS (Network File System) — сетевой файловой системы, которая является общераспространенной в мире Unix (и которую поддерживает Linux). Вместо этого в мире Microsoft предпочтение отдается стандарту NetBEUI, который является расширением спецификации NetBIOS, основанной на протоколе LanManager компании IBM. Все же благодаря стремительному развитию Интернета компания Microsoft была вынуждена обеспечить поддержку передачи данных NetBIOS через каналы связи TCP/IP.

Samba

Разочарованные невозможностью обеспечить простой обмен данными через сеть с системами Microsoft, группа поклонников Linux решила разработать программное средство, которое позволило бы системе Linux обмениваться данными с системами Microsoft на языке Microsoft. Программисты Linux под предводительством Эндрю Тридгелла (Andrew Tridgell) из Австралии начали работу по расшифровке содержимого блоков SMB (Server Message Blocks) и воссозданию поддержки NetBEUI в среде Linux. Благодаря этому операционная система Linux получила возможность вести себя так, как ведет себя система Microsoft. В результате на свет появилось программное средство Samba.

Samba — это набор программ, которые позволяют операционным системам семейства Unix (включая Linux) обмениваться данными с узлами Microsoft. Компания Microsoft часто обозначает протокол NetBEUI сокращением CIFS, которое расшифровывается как Common Internet File System — общая файловая система Интернета. Этот термин не соответствует действительности, так как протокол NetBEUI широко поддерживается только системами Microsoft. NetBEUI является широковещательным протоколом. Он подразумевает, что каждая система, начинающая работу в сети, оповещает об этом все остальные системы при помощи широковещательного сообщения. Оповещение отправляется в сеть также в момент, когда система прекращает работу в сети. Также в процессе своего функционирования система, использующая NetBEUI, должна периодически напоминать остальным систем о своем присутствии.

ПРИМЕЧАНИЕ

В данном материале время от времени я буду использовать три термина: узел NT (это может быть любая рабочая станция NT, сервер-член домена NT, отдельный сервер NT, а также главный (PDC) или резервный (BDC) контроллер домена), на котором работает операционная система Windows NT, узел Win9x и узел Microsoft. Первые два термина (узел NT и узел Win9x) используются в ситуациях, когда важно подчеркнуть, какая именно операционная система Microsoft работает на том или ином сетевом узле. Третьим термином (узел Microsoft) я буду обозначать узлы, на которых работает любая операционная система Microsoft. Такое различие необходимо, так как узлы, оснащенные разными ОС, ведут себя по-разному. Более подробно об этом рассказывается в разделе «Сетевые рабочие среды Microsoft» чуть далее в данной главе.

Набор программ Samba включает в себя несколько различных программ. Две из них предназначены для того, чтобы имитировать работу узла Microsoft в сети. Одна предназначена для настройки Samba. Еще одна осуществляет тестирование конфигурации Samba. Также в пакет Samba входят несколько утилит. Набор программ Samba включает в себя:

- smbd — демон Samba, обеспечивающий обслуживание клиентов SMB;

- nmbd — демон сервера имен NetBIOS, который обеспечивает доступ к службам имен NetBIOS через IP, благодаря этому узел Unix появляется в разделе Network Neighborhood (Сетевое окружение) операционной системы Windows;

- smbclient — клиентская утилита, которая обеспечивает доступ к сетевым ресурсам SMB в стиле FTP;

- swat — средство администрирования Samba, основанное на Web, эта программа обеспечивает настройку файла smb.conf;

- testparm — средство проверки корректности конфигурационного файла smb.conf.

В данной главе я расскажу вам о некоторых из этих программ подробнее. Однако основное внимание будет уделено программе swat, так как должное конфигурирование swat не очевидно, а в результате неправильной настройки Samba может возникнуть масса проблем, начиная от неработоспособности сервера и заканчивая серьезным нарушением безопасности.

ВНИМАНИЕ

Вы должны обеспечить должную защиту файлу smb.conf (а также, очевидно, бинарному файлу swat). Файл smb.conf определяет, каким доступом обладают те, кто пользуется предлагаемыми вами службами Samba и обращается к вашему узлу OpenLinux. Демон smbd работает на уровне привилегий root, поэтому в результате неправильной настройки файла smb.conf подключающийся к Samba удаленный клиент может получить доступ к вашей системе от лица пользователя root. В этом случае программа smbd перекрывает любые параметры безопасности и разрешения, назначенные клиенту как стандартному пользователю Unix.

SWAT — средство администрирования Samba

В комплект поставки Samba 2.0 было включено новое средство администрирования Samba Web Administration Tool (SWAT). Средство администрирования SWAT было разработано в ответ на многочисленные запросы администраторов, нуждающихся в более удобном способе управления многочисленными параметрами конфигурационного файла smb.conf. Прилагаемая к Samba документация слишком объемна и слишком сложна, используя ее вряд ли можно выполнить быструю настройку Samba, особенно если вы новичок. Благодаря SWAT вы получаете простой в использовании инструмент конфигурирования, а также более удобную контекстно-зависимую электронную подсказку. Электронная документация SWAT все же может показаться несколько сложной для новичков, однако само по себе средство SWAT является чрезвычайно удобным инструментом управления конфигурационными переменными.

Подготовка к запуску SWAT

При установке с использованием механизма RPM, равно как и в процессе начальной установки комплекта Caldera OpenLinux, программное средство SWAT предлагается в виде отдельного пакета RPM. Обратите внимание, что в комплекте поставки Red Hat SWAT не является отдельным пакетом и устанавливается на компьютере в составе основного RPM-пакета Samba. В разных комплектах Linux установка SWAT выполняется по-разному, поэтому прежде чем устанавливать SWAT, вы должны изучить документацию, прилагаемую к вашему комплекту Linux.

При установке через RPM в комплекте Caldera (и возможно, во многих других комплектах) служба swat добавляется в файл /etc/inted.conf. При этом учитывается рекомендация разработчиков Samba, предписывающая использовать для swat порт 901. Таким образом, если порт 901 связан супердемоном inetd, значит, скорее всего, в вашей системе служба swat установлена именно таким образом.

Запуск SWAT с использованием inetd

Если Samba устанавливается в процессе начальной установки ОС, скорее всего, все необходимое конфигурирование будет выполнено автоматически, однако если вы загружаете из Интернета последнюю версию Samba, компилируете ее и устанавливаете в системе вручную и при этом намерены обеспечить запуск swat с использованием inetd, вы должны выполнить кое-какие настройки самостоятельно. Даже если программа установки выполнила все эти шаги за вас, я все равно рекомендую вам изучить описанную

далее процедуру, так как она может оказаться для вас полезной в процессе устранения неисправностей. Первый этап настройки связан с редактированием файлов /etc/services и /etc/inetd.conf. От имени пользователя root вы должны добавить в файл /etc/services следующую строку:

```
swat 901/tcp
```

Эта строка назначает порт 901 для TCP-подключения к программе swat. Далее в соответствии с документацией Samba необходимо добавить в файл inetd.conf следующую строку:

```
swat stream tcp nowait.400 root /путь/к/swat swat
```

Эта строка предписывает метадемону inetd запускать swat в поточном режиме tcp на уровне привилегий root. Документация Samba предлагает использовать параметр nowait.400. Это означает, что программу swat разрешается запускать до 400 раз в минуту. Если не использовать данный параметр, по умолчанию будет разрешен запуск swat не более 40 раз в минуту — скорее всего, этого будет вполне достаточно. Добавлять какое-либо число к параметру nowait вовсе не обязательно, однако в случае необходимости вы вполне можете сделать это в соответствии с вашими предпочтениями.

Если вы желаете, чтобы программа swat запускалась с использованием tcpd (механизм TCP Wrappers), как это обсуждалось в главе 15, вместо предыдущей записи вы можете добавить в inetd.conf следующую:

```
swat stream tcp nowait root /usr/sbin/tcpd /путь/к/swat
```

Вы можете указать любое удобное вам количество экземпляров swat, которое разрешается запустить в течение 60-секундного периода (например, 100).

ПРИМЕЧАНИЕ

Вместо строки /путь/к/swat необходимо указать корректное местоположение программы swat. При установке OpenLinux по умолчанию этот исполняемый файл расположен в каталоге /usr/sbin, однако он может быть расположен также в каталоге /usr/bin или где-либо в другом месте — все зависит от того, какой комплект Linux вы используете и каким способом вы устанавливаете swat. Для того чтобы найти swat, воспользуйтесь командой locate.

Утилиты Samba поддерживают работу с PAM. Иными словами, они обращаются к сведениям, содержащимся в модулях PAM (Password Authentication Module), расположенных в подкаталоге /etc/pam.d (см. главу 1). Как правило, при этом используется файл с именем samba. В данном файле содержится несколько строк, например:

```
auth        required        /lib/security/pam_pwdb.so shadow nullok
account     required        /lib/security/pam_pwdb.so
```

Если в любой из вышеуказанных строк содержится pam.deny.so, вы не сможете запустить swat. Возможны также другие комбинации, однако приведенные здесь строки являются строками, по умолчанию добавляемыми в комплекте Caldera. В других комплектах могут использоваться другие комбинации, кроме того, вы можете самостоятельно добавлять другие модули, как об этом говорилось в главе 1.

По умолчанию в комплекте Caldera в файл hosts.deny также добавляется строка, которая выглядит следующим образом:

```
swat:  ALL EXCEPT 127.0.0.1
```

Эта строка запрещает кому бы то ни было администрировать Samba удаленно. Если вы уверены в том, что ваша локальная сеть является хорошо защищенной и полностью безопасной, вы можете заменить адрес 127.0.0.1 на LOCAL. Учтите, что если вы выполняете установку по умолчанию, при подключении к swat имя пользователя и пароль будут передаваться через сеть в незашифрованном виде.

Теперь вы можете обратиться к swat. Для этого запустите ваш web-браузер, введите http://localhost:901/ в поле адреса и нажмите Enter. Вам будет предложено ввести имя пользователя и пароль. Если вы работаете от имени root, введите имя пользователя root и пароль пользователя root.

Вместо localhost вы можете использовать любое корректное ассоциированное с вашим узлом имя, включая IP-адрес 127.0.0.1. Если вы хотите обеспечить возможность администрировать Samba удаленно, очевидно, лучше использовать для этой цели защищенный web-сервер (с поддержкой SSL) и в ходе любых сеансов выполнять шифровку любых передаваемых через сеть данных.

ССЫЛКА

О том, как осуществляется конфигурирование защищенного web-сервера Apache, рассказывается в главе 20.

Если для подключения к swat вы будете использовать любую другую (отличную от root) корректную пару имя пользователя/пароль, вы сможете просматривать экраны swat, однако некоторые возможности будут для вас недоступны. Кнопка Commit Changes (сохранить изменения) не будет отображаться на экране, так как правом вносить изменения в файл smb.conf обладает только пользователь root (предполагается, что разрешения на доступ к этому файлу настроены должным образом). Также вы не будете обладать возможностью запускать и останавливать демоны smbd и nmbd.

Если в момент запуска swat файл smb.conf отсутствует, программа swat будет запущена с использованием конфигурационных значений по умолчанию.

ВНИМАНИЕ

Если у вас есть составленный вручную файл smb.conf, который вы хотели бы сохранить для последующего использования, сделайте его копию или не запускайте swat. Дело в том, что программа swat попытается извлечь из конфигурационного файла smb.conf любую полезную информацию, а затем перезапишет файл smb.conf в своем собственном формате. При этом будут утеряны комментарии, а также такие директивы, как include и copy.

Запуск SWAT с использованием Apache

Настройка swat для запуска с использованием web-сервера Apache выполняется несколько сложнее, и требует большего внимания к деталям, в противном случае вы рискуете нарушить безопасность вашей системы. Если на вашем компьютере уже установлен web-сервер Apache и вы обладаете опытом его настройки, описываемые далее действия помогут вам настроить ваш сервер так, чтобы он мог запускать программу swat.

Прежде всего в корневом каталоге документов вашего web-сервера (document root) следует создать подкаталог с именем swat. Если вы планируете, что администрирование системы должно осуществляться удаленно, будет лучше, если вы будете использовать защищенный корневой каталог документов. В каталог documentrooot/ swat следует скопировать содержимое каталога samba/swat Также в соответствующие подкаталоги следует скопировать содержимое подкаталогов help, images и include. Бинарный файл swat следует скопировать в подкаталог cgi-bin вашего web-сервера.

В подкаталоге swat корневого каталога документов вашего web-сервера создайте файл с именем .htaccess. В этот файл необходимо поместить следующий текст:

```
AuthName "swat restricted"
AuthType Basic
AuthUserFile  /etc/swat.users
require valid-user
```

Далее необходимо создать файл авторизированных пользователей. Это выполняется при помощи следующей команды:

```
htpasswd -c /etc/swat.users root
```

Вам будет предложено ввести пароль. После этого будет создан файл с пользователем root и хэшированным (то есть зашифрованным) паролем внутри.

ВНИМАНИЕ

Вы должны убедиться в том, что правом записи в данный файл обладает только пользователь root. Кроме того, хранящиеся в данном файле пароли должны быть хорошо подобраны, для того чтобы обеспечить должный уровень защиты.

Наконец, после этого в файле Apache access, conf необходимо найти строку, которая начинается с имени параметра AllowOverride. Необходимо убедиться, что данный параметр имеет значение AuthConfig. После этого следует отдать вашему серверу команду SIGHUP (для сервера Apache 3.1.x следует использовать команду apachectl с аргументом restart).

После этого вы сможете обратиться к swat при помощи web-браузера, указав в строке адреса http://localhost/swat/cgi-bin/swat.

ВНИМАНИЕ

Вне зависимости от того, какой из двух ранее описанных методов запуска SWAT вы предпочтете, вы не должны забывать, что пароли передаются через сеть в незашифрованном виде, если только вы не настроите ваш web-сервер Apache на использование SSL и не разместите swat в корневом каталоге документов SSL

Использование SWAT

Как только вы начнете использовать SWAT, вам будет уже сложно отказаться от столь удобного средства администрирования. Этот инструмент облегчит вам выполнение фактически любых операций конфигурирования smb.conf, которые можно выполнить с использованием командной строки, включая перезагрузку nmbd и smbd, а также получения сведений о состоянии Samba (рис. 19.1). Используя SWAT, вы также сможете просматривать относящиеся к Samba страницы электронной документации man pages, которые будут отображаться в отдельных окнах браузера. Когда вы впервые запускаете SWAT, система

предлагает вам ввести имя пользователя и пароль. Вы должны использовать учетную запись root, в противном случае вы сможете только просматривать конфигурацию файла smb.conf, не внося в нее каких-либо изменений. Правом модификации файла smb.conf обладает только пользователь root. При обращении SWAT открывает свою домашнюю страницу, внешний вид которой показан на рис. 19.1. Можно заметить, что в основном на этой странице располагаются ссылки на электронную документацию. Многочисленные ссылки указывают на разнообразную справочную информацию, однако при этом не запускается каких-либо программ. В верхней части страницы располагаются несколько значков, при помощи которых осуществляется доступ к другим страницам, через которые выполняется управление Samba: Home (страница, с которой вы начинаете), Globals (глобальная конфигурация), Shares (пользовательские ресурсы общего доступа), Printers (общие ресурсы печати), Status (состояние), View (конфигурация просмотра) и Password (пароль).

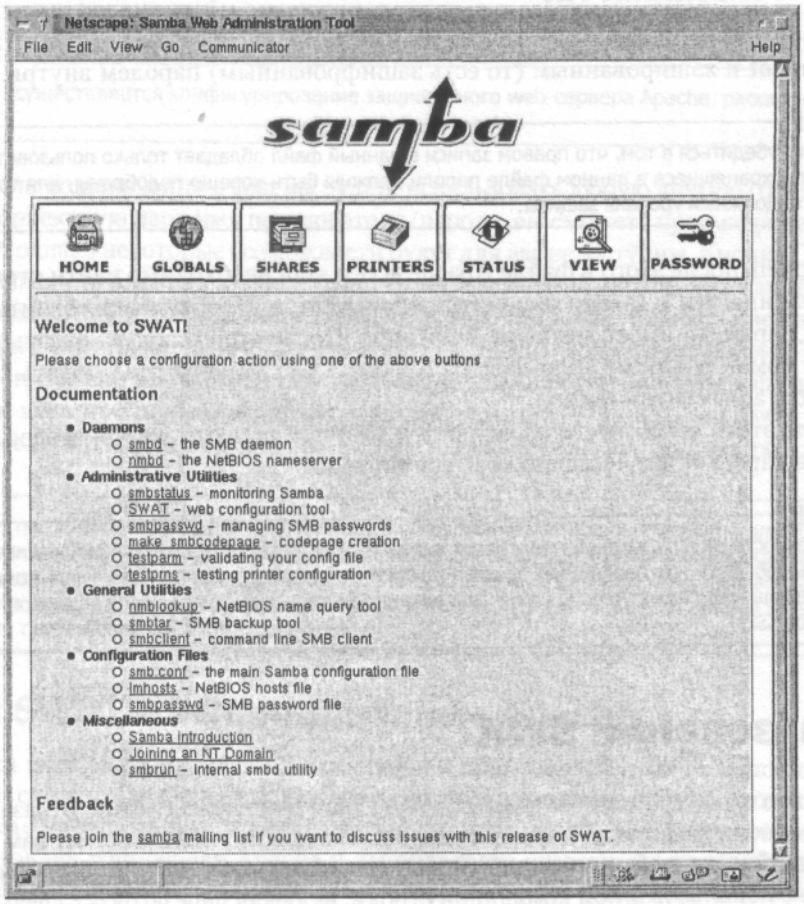

Рис. 19.1. Домашняя страница SWAT

ПРИМЕЧАНИЕ

Если для доступа к SWAT вы используете Netscape, система не будет предлагать вам вводить имя пользователя и пароль, если только вы не закроете браузер и не запустите его вновь. Дело в том, что браузер запоминает (кэширует) идентификационную информацию в течение всего сеанса работы. Такое поведение является нормальным. Если вы закроете браузер, а затем запустите его вновь и вновь обратитесь к SWAT, система вновь предложит вам ввести имя и пароль пользователя.

Если в момент запуска Samba файл smb.conf отсутствует, Samba будет использовать значения конфигурационных параметров по умолчанию. Взгляните на страницу View и вы увидите именно то, что будет записано в конфигурационный файл. Этот файл записывается на диск каждый раз, когда вы щелкаете на кнопке Commit Changes (сохранить изменения) или Create Share (создать общий ресурс) или Create Printer (создать принтер). Если вы покинете какую-либо страницу (Clobals, Shares или Printers), не щелкнув при этом на кнопке Commit Changes (внести изменения), все сделанные вами изменения не будут сохранены. По умолчанию файл smb.conf выглядит следующим образом:

```
Samba config file created using SWAT
from localhost (127.0.0.1)
Date: 1999/01/09 13:06:14
Global parameters
```

По мере того как вы будете вносить в конфигурацию по умолчанию изменения и сохранять эти изменения на диске, файл будет перезаписываться. Для того чтобы изменения вступили в силу, демонам smbd и nmbd необходимо послать сигнал SIGHUP.

ВНИМАНИЕ

В настоящее время в момент запуска SWAT не создает резервную копию файла smb.conf, поэтому прежде чем запускать SWAT, рекомендуется самостоятельно сделать резервную копию этого файла.

Сетевая рабочая среда Microsoft

Сеть, основанная на сетевых программных продуктах Microsoft, может принадлежать к одной из двух категорий. В одной сети могут использоваться одновременно оба эти подхода, однако такие ситуации встречаются редко. Среда первой категории — это домен NT. В составе домена NT присутствует один сервер NT, выполняющий функции первичного контроллера домена (PDC, Primary Domain Controller), а также зачастую в домене NT присутствует один или несколько резервных контроллеров домена (BDC, Backup Domain Controller). Также в домене могут функционировать другие серверы NT, которые являются рядовыми членами домена, и, конечно же, рабочие станции NT. Назначение и функционирование серверов PDC и BDC напоминает собой функционирование комбинации главного (master) и подчиненных (slave) серверов NTS (Network Information Server). PDC и BDC разрешают или запрещают пользователям сети доступ к ресурсам домена NT.

ПРИМЕЧАНИЕ

Термин «домен NT» не имеет никакого отношения к доменам IP-адресов, о которых мы говорили, обсуждая основы сетей TCP/IP. В данной книге термин «домен NT» используется в контексте сетей, основанных на операционной системе Microsoft Windows NT. Если я имею в виду домен IP-адресов, я говорю просто «домен».

Рабочая среда Microsoft второй категории — это одноранговая сеть. В сетях такого типа не существует сервера PDC. Все компьютеры обладают равными возможностями по защите своих данных. Компьютеры одноранговой сети могут обеспечивать совместный доступ к хранящейся на них информации, при этом каждый компьютер самостоятельно определяет, к каким файлам и каталогам будут иметь доступ внешние пользователи.

Сервер NT в качестве PDC в локальной сети

При запуске Samba в сети, в которой работает PDC, необходимо уделить внимание некоторым нюансам. Следует учитывать, что в домене NT каждому узлу соответствует некоторый параметр старшинства. Иными словами, некоторые члены домена NT являются более старшими по отношению к другим. Этот параметр старшинства определяется версией ОС и некоторыми другими характеристиками. Параметр старшинства узла влияет на то, каким образом будет вести себя узел, попадая в домен NT или покидая его.

Когда в сети появляется новый узел, использующий SMB, он анализирует текущий состав функционирующих в этой сети узлов. Если в сети не существует узла, обладающего более высоким, чем у него, показателем старшинства, данный узел начинает борьбу за звание главного обозревателя (browser war). Иными словами, осуществляется процедура выбора главного обозревателя сети. Главный обозреватель сети — это компьютер, который отвечает за просмотр доступных в сети ресурсов общего пользования. Процедура выбора главного обозревателя выполняется также в случае, если компьютер, выполнявший до этого функции главного обозревателя, покидает сеть. Для каждого узла Microsoft показатель его старшинства жестко задан, и его нельзя изменить, однако Samba не является продуктом Microsoft, поэтому вы можете менять это значение при помощи глобальной конфигурации (страница Globals). Иными словами, вы можете увеличивать или снижать старшинство узла Samba в зависимости от своего собственного желания. Вы также можете по своему желанию определить, сможет ли данный узел Samba инициировать борьбу за звание главного обозревателя.

ВНИМАНИЕ

Сервер Samba можно настроить таким образом, чтобы он был самым старшим сетевым узлом в сети Microsoft, при этом можно сделать так, чтобы он выигрывал абсолютно любую борьбу за звание главного обозревателя. Однако на самом деле это не лучшая идея, особенно в рамках домена NT. Несмотря на то, что сервер Samba в состоянии отобрать контроль над доменом у сервера PDC, при этом остальные узлы Microsoft могут потерять возможность корректно просматривать имеющиеся в сети сетевые ресурсы.

Вхождение в домен NT

В Samba версии *2*.0.x появилась возможность стать частью домена NT. Чтобы добиться этого, необходимо выполнить следующую процедуру.

Во-первых, при помощи утилиты Server Manager на главном контроллере PDC следует добавить запись с именем NetBIOS вашего сервера Samba. Это имя вовсе не обязательно должно совпадать с DNS-именем сервера Samba, однако в большинстве случаев для вас будет удобнее, если эти два имени будут одинаковыми. Сервер Samba должен быть указан как независимый сервер или как рабочая станция.

Остальные действия следует выполнить на сервере Samba. В данном тексте я предполагаю, что домен NT обладает именем NTDomain, сервер PDC обладает NetBIOS-именем PDC, а сервер BDC обладает NetBIOS-именем BDC.

Теперь следует остановить работу всех демонов Samba и добавить в файл smb.conf следующие записи (если вы используете SWAT, на странице Global вы должны выбрать Advanced View):

```
Security = Domain
Workgroup = NTDomain
Password Server = PDC BDC (и другие BDC)
encrypt password = yes
```

Возможно, вам потребуется также настроить сервер WINS. Для этого вам необходимо знать либо IP-адрес, либо имя DNS (но не NetBIOS) вашего сервера WINS. Выполните следующую команду:

```
smbpasswd -j NTDomain -r PDC
```

Если все сделано правильно, на экране появится сообщение Joined domain NTDomain (система вошла в состав домена NTDomain). При этом в частном каталоге samba будет создан файл с суффиксом .mac (что означает Machine).

Для настройки данного узла требуется выполнить еще кое-какие процедуры, однако данный узел появится в разделе Network Neighborhood (Сетевое окружение), а его ресурсы общего доступа станут доступными для других систем.

Пока вы выполняете все эти действия, вы должны быть уверенными в том, что в сети никто не использует программ прослушивания трафика (sniffer). Причина состоит в том, что в процессе начального обмена ключами NT передает через сеть в незашифрованном виде личный ключ для входа в домен. Любой, кто владеет этим ключом, сможет войти в домен под видом сервера, который вы настраиваете.

На завершающем этапе следует указать пароль для пользователей, которые будут подключаться к серверу Samba. Для этой цели следует использовать программу smbpasswd, которой следует передать аргумент -a (то есть *add* — добавить пользователя) и имя создаваемого пользователя. После этого система предложит вам ввести пароль. Операция добавления пользователя Samba закончится неудачей в случае, если добавляемый вами пользователь не является зарегистрированным пользователем системы Linux.

СОВЕТ

Чтобы сократить объем работ, связанных с администрированием, вы можете обеспечить динамическое создание пользователей по мере того, как они пытаются получить доступ к общим ресурсам сервера Samba. Для этого в глобальный параметр add user script следует добавить строку: /usr/sbin/useradd %u.

Перечисленные ранее процедуры подготовят сервер Samba для использования в рамках домена NT, однако после этого вы также должны создать на нем каталоги общего доступа (Shares).

СОВЕТ

Еще один способ экономии времени в процессе администрирования предусматривает использование параметра Unix password sync на странице Global раздел Advanced View. Если вы присвоите этому параметру значение true, пароль Unix будет обновляться каждый раз при изменении пароля NT.

Связь через сеть с PDC, расположенным в удаленной подсети

В некоторых случаях сервер Samba подключен к подсети, в которой отсутствует PDC или BDC. Иными словами, для того чтобы достичь PDC или BDC, трафик из локальной сети должен быть передан через шлюз. В данной ситуации говорят, что сервер Samba расположен в «широковещательно-изолированной» (broadcast isolated) сети. Протокол NetBIOS является широковещательным протоколом - главный обозреватель локальной сети периодически отправляет всем узлам локальной сети широковещательное сообщение, благодаря которому узлы локальной сети обновляют свои списки обзора

сетевых ресурсов. Когда в сети начинает работу еще один узел, он оповещает об этом все остальные узлы при помощи широковещательного сообщения. Однако широковещательный трафик не передается через шлюзы, поэтому информация NetBIOS доступна только для узлов локальной сети. Таким образом, узлы в широковещательно-изолированной сети будут обладать сведениями только об именах узлов локальной сети, однако имена компьютеров в других сетях будут для них неизвестны.

Чтобы решить проблему, используются локальные главные обозреватели (local browse master). Локальный главный обозреватель — это узел, выполняющий функции главного обозревателя для локальной подсети и знающий о том, где расположен главный обозреватель домена (как правило, главным обозревателем домена является сервер PDC). Сервер Samba может выполнять функции локального главного обозревателя. Для этого необходимо присвоить параметрам local master и preferred master значение true. Сервер Samba также должен обладать IP-адресом (или DNS-именем) сервера WINS. После перезапуска демона nmbd сервер Samba инициирует процедуру выбора обозревателя (борьбу за звание обозревателя). Чтобы сервер Samba всегда выбирался в качестве главного обозревателя, необходимо присвоить соответствующему параметру значение 65. По умолчанию уровень ОС (OS level) равен значению 0. Все обсуждаемые в данном абзаце параметры располагаются на странице Global в разделе Basic View. Если в локальной сети отсутствует сервер WINS, вы можете также включить поддержку WINS (wins support).

СОВЕТ

Если в подсети работает сервер WINS компании Microsoft, вы ни в коем случае не должны включать на сервере Samba поддержку WINS. Сервер Samba должен быть единственным сервером WINS в подсети, в противном случае обзор сетевых ресурсов будет выполняться некорректно.

Сервер Samba в одноранговых сетях Microsoft (NT и/или Windows 9x)

Этот режим работы является для сервера Samba режимом работы по умолчанию. Иными словами, сразу после установки сервер Samba готов к работе в составе одноранговой сети Microsoft. Если не вносить в конфигурацию никаких изменений, то для начала работы вы должны просто определить ресурсы общего доступа. Однако на практике даже для работы в одноранговой сети в конфигурацию Samba почти всегда требуется внести кое-какие изменения. Например, операционные системы Win NT 4 с установленными пакетами добавлений Service Pack 3 или 4, а также Win 95 с обновлением OSR2 или Win 98 требуют использования зашифрованных паролей. Некоторые представляющие интерес изменения конфигурации будут обсуждаться далее.

Все, что вам необходимо предпринять на данном этапе, — это создать общие папки и общие принтеры. Вы также можете приказать Samba выполнять функции сервера WINS для компьютеров Windows 9x и рабочих станций Windows NT, а также для независимых серверов.

Использование Linux в качестве замены PDC

Команда разработчиков Samba работает над механизмом эмуляции NT PDC. На момент написания данной книги соответствующий код по-прежнему является экспериментальным, однако он входит в состав комплекта поставки Samba.

ВНИМАНИЕ

На текущий момент данную конфигурацию не рекомендуется использовать в реальных производственных условиях, так как некоторые пользователи указывают на то, что код Samba может повредить базу данных SAM (база данных SAM используется операционной системой NT для хранения сведений об учетных записях и правах на доступ). Код эмуляции PDC должен рассматриваться как альфа-версия данного механизма.

Однако можно предположить, что в скором времени все ошибки будут устранены, и код эмуляции PDC можно будет безбоязненно использовать в условиях реального производства. Поэтому здесь я приведу краткий перечень действий, которые потребуется выполнить для того, чтобы создать Samba PDC и добавить в этот домен новые узлы. Вы должны иметь в виду, что в будущем данный набор действий может измениться. Для уточнения следует обратиться к документации, прилагаемой к комплекту Samba.

1. В раздел Global программы swat необходимо внести следующие изменения (некоторые из этих

параметров следует искать на странице Advanced View):

workgroup = SAMBA (здесь укажите имя вашего домена)
encrypted passwords = yes
domain logon = yes
domain master = yes
preferred master = yes
security = user (при использовании другого сервера smb здесь можно указать server, однако при этом вам потребуется также ввести значение "password server =", где следует указать IP-адрес или DNS-иня)

2. Также возможно потребуется добавить следующее:
wins support = yes
logon script = %U.bat

3. Чтобы использовать сценарии подключения, необходимо создать общую пап ку, где будут храниться сценарии подключения. Для этого следует воспользо ваться разделом Shares программы swat и добавить:
[netlogon]
path = /ioou/e/netlogon writeable = no guest = no

4. После этого на сервере Samba следует создать записи для всех компьютеров. Это то же самое, что создавать записи в Server Manager for Domains:
smbpasswd -m NetBIOSимя

Здесь следует указать имя (или имена) компьютера (или компьютеров), который(е) должен(ны) войти в состав домена SAMBA. Вам будет предложено ввести пароль. Используйте пароль machine.

Как уже отмечалось ранее, вы также должны будете создать некоторые учет ные записи с использованием smbpasswd (если вы этого еще не сделали).

Теперь перезапустите (SIGHUP) демон smbd.

На узле Microsoft выберите Start (Пуск) > Control Panel (Панель управления) >Networking (Сеть) и измените домен на SAMBA (подставьте имя, которое вы указали в строке workgroup =). Не следует выбирать Create an Account (создать учетную запись). Когда вы щелкнете на ОК, на экране должна появится запись Welcome to SAMBA Domain (добро пожаловать в домен SAMBA).

Перезагрузите узел Microsoft.

На компьютере Windows NT Workstations от момента времени, когда вы нажали на Ctrl+Alt+Del, до момента, когда на экране появляется приглашение на вход в систему, должно пройти не более 20 секунд, в противном случае что-то работает не правильно. Вы должны увидеть три графы, помеченные Name (имя), Password (пароль) и Domain (домен).

Общий доступ к каталогам

Организация общего доступа к каталогам с использованием Samba выполняется очень просто. Средство администрирования SWAT существенно упрощает процесс создания папки общего доступа. Для этого требуется всего лишь указать имя общей папки, а затем уделить внимание настройке некоторых параметров, например пути к соответствующему дисковому каталогу. Настройка большинства параметров не представляет проблем: вы просто отмечаете те параметры, которые вам нужны. Однако при использовании некоторых параметров могут возникнуть проблемы. О таких параметрах будет подробнее рассказано далее. Также следует учитывать, что имя общего ресурса Homes имеет специальное значение для Samba — об этом также будет сказано дополнительно. Раздел Share разбит на несколько функциональных областей, о которых рассказывается в последующих разделах.

Base options

Указав имя общего ресурса, просто щелкните Create, и SWAT занесет в конфигурационный файл smb.conf значения по умолчанию для данного общего ресурса. Любым общим папкам, за исключением общей папки Homes, по умолчанию ставится в соответствие каталог /tmp, а поле комментария остается пустым. При желании вы можете изменить это.

ВНИМАНИЕ
Общая папка определяет область файловой системы, в рамках которой может действовать удаленный пользователь. Пользователь не может покинуть эту область (указанный каталог и все его подкаталоги), если только параметр wide links не обладает значением yes. В рамках указанной области пользователь обладает привилегиями, определенными для него в рамках конфигурации Samba. Эти привилегии могут быть шире, чем те

привилегии, которыми пользователь наделяется в случае, если он подключается к системе как обычный пользователь Linux.

Security options

В данном разделе определяется набор разрешений, которым будет обладать пользователь (например, осуществлять запись, чтение и создание файлов). Здесь же вы можете идентифицировать пользователя guest и определить набор соответствующих разрешений. Если используется режим guest only, значит, при доступе к данной общей папке все пользователи обязаны использовать учетную запись guest

При помощи этого раздела вы также можете разрешать или запрещать доступ к обшей папке — механизм напоминает использование файлов hosts.allow и hosts.deny.

Параметр revalidate принуждает пользователей аутентифицировать себя каждый раз, когда они обращаются к общей папке.

Самыми непонятными параметрами данного раздела являются параметры группы create mask. В данном случае create mask является синонимом create mode. Здесь необходимо указать значение, которое обычно используется совместно с командой chmod.

Logging options

Здесь можно настроить механизм документирования доступа к общей папке, а также включить или отключить возможность просмотра с использованием smbstatus, обращается ли кто-либо к общей папке в данный момент. Значение этого параметра необходимо оставить равным yes.

Tuning options

По умолчанию максимальное допустимое количество подключений равно нулю, что означает, что пользователи могут подключаться к общей папке в неограниченном количестве. Ограничение количества подключений обеспечивается с использованием механизма блокирования (locks).

Два параметра, имеющих отношение к синхронизации, по умолчанию обладают значением по. Об этих параметрах следует рассказать чуть подробнее. Многие приложения Windows, включая Windows 98 Explorer (проводник), не делают разницы между *скидыванием буфера на диск* (flushing buffer) и *синхронизацией данных с диском* (sync to disk). В операционной среде Linux при выполнении синхронизации с диском (sync to disk) клиентский процесс, как правило, приостанавливает функционирование до тех пор, пока содержимое всех буферов не будет записано на жесткий диск. Если достаточно большое количество приложений Windows будет выполнять синхронизацию с диском, в то время когда на самом деле достаточно выполнить скидывание буферов на диск, производительность системы Linux может существенно снизиться. Именно по этому значение параметра strict sync по умолчанию равно по.

Параметр sync always определяет, должен ли системный вызов синхронизации с диском возвращать управление вызвавшему процессу прежде, чем будет завершена операция записи на диск. В этом случае сервер Samba будет осуществлять синхронизацию в соответствии с пожеланиями клиентов. Если значение параметра sync always равно yes, то значение параметра strict sync также должно быть равно yes, в противном случае параметр sync always будет игнорироваться.

Настраивайте эти параметры с осторожностью. Если оба параметра будут равны yes и при этом большое количество клиентов будет выполнять запись на диск, производительность системы может существенно понизиться.

Filename handling

Значения по умолчанию присвоенные параметрам этой категории вполне подходят для рабочих сред NT 4 и Windows 95 и более поздних версий Windows. Если в вашей рабочей среде до сих пор используются операционные системы DOS и Windows 3.x, вы должны уделить внимание должной настройке этих параметров, чтобы обеспечить корректный доступ к файлам для ваших клиентов.

В данном разделе также настраивается соответствие между некоторыми специальными категориями файлов Microsoft (например, скрытые или системные файлы) и файлами Unix (файлы с именами, начинающимися с символа точки). Взаимно-однозначного соответствия не существует, поэтому администратор должен специально определить порядок обработки разнообразных типов файлов.

Browse options

В этой категории есть всего один параметр: является ли общая папка просматриваемой или нет. В

Windows NT для скрытия общих папок используется стандартный метод. Если в конце имени общей папки стоит символ $, такая папка считается скрытой и не отображается в списке просмотра общих папок. В Samba вы можете скрыть папку с любым именем, для этого достаточно присвоить параметру browseable значение по. При этом возможность доступа к папке сохраняется, просто имя папки не отображается в окошке Network Neighborhood (Сетевое окружение). Если вы и без того знаете имя этой общей папки, вы по-прежнему сможете получить к ней доступ.

Locking options

Параметры данного раздела по умолчанию настроены так, чтобы защищать общие папки, в отношении которых разрешены операции чтения-записи. Также параметры настроены так, чтобы обеспечить быстрый доступ там, где это допустимо (не возникает конфликтов блокирования одновременного доступа к файлам). Однако для некоторых общих папок, из которых разрешается только читать, вы можете обеспечить значительное увеличение скорости доступа, присвоив параметру fake oplocks значение yes.

ВНИМАНИЕ

Перед тем как менять значения параметров блокирования доступа к общим папкам или файлам, убедитесь в том, что вы представляете себе возможные последствия подобных изменений. Например, если вы присвоите параметру fake oplocks значение yes в отношении общей папки, для которой разрешается не только чтение, но и запись, целостность содержимого папки может быть нарушена в случае, если несколько клиентов в одно и то же время попытаются выполнить запись в *один и тот же файл.*

Некоторые параметры имеет смысл перенастраивать в зависимости от того, с какой сетью вы имеете дело. Например, если в вашем распоряжении достаточно надежная сеть, установив параметр leaving oplocks (сохранять блокировку) равным значению yes, вы получите существенное увеличение скорости. Однако для ненадежной сети это может оказаться неэффективным, так как сохранение блокировки для отключающихся клиентов понизит скорость.

Miscellaneous options

В этом разделе содержатся разнообразные параметры, которые тематически нельзя включить ни в один из предыдущих разделов. При помощи параметров данного раздела контролируются различные аспекты функционирования Samba, например, доступность общей папки и порядок обработки символических ссылок.

Параметр wide links контролирует, может ли пользователь, получивший доступ к общей папке, перемещаться по ссылкам, указывающим куда-либо вне каталога, соответствующего этой общей папке. Если параметр wide links равен значению yes (это значение по умолчанию), это значит, что пользователь сможет воспользоваться символическими ссылками, указывающими на файловые ресурсы вне каталога, соответствующего общей папке. Параметр follow symlinks (значение которого по умолчанию равно yes) определяет, сможет ли пользователь вообще пользоваться какими-либо символическими ссылками. При использовании этих двух параметров в комбинации возможен один из трех вариантов настройки:

- если параметр follow symlinks равен значению по, символические ссылки не будут работать вне зависимости от значения параметра wide links;

- если параметр follow symlinks равен значению yes, а параметр wide links равен значению по, пользователи смогут использовать только те символические ссылки, которые указывают на ресурсы, расположенные в рамках общей папки; символические ссылки, указывающие на ресурсы, расположенные вне каталога общей папки, обрабатываться не будут;

- если параметр follow symlinks равен значению yes, а параметр wide links тоже равен значению yes, пользователи не будут ограничены только лишь содержимым общей папки, они смогут обращаться к ресурсам, на которые указывает символическая ссылка, даже если эти ресурсы расположены вне общей папки, то есть где угодно в системе; если при этом пользователь обладает возможностью создания или модификации символических ссылок, он сможет получить доступ фактически к любому файлу в системе.

ПРИМЕЧАНИЕ

Если параметр wide links равен значению по, просмотр содержимого каталогов существенно замедлится, так как Samba всегда будет проверять, где именно расположен ресурс, на который указывает ссылка: в общей папке или вне ее.

Определенный интерес представляют четыре параметра exec. Эти параметры позволяют выполнять

различные программы в момент, когда пользователь подключается к общей папке или отключается от общей папки, при этом также указывается, надо ли запустить программу от лица обычного пользователя или от лица пользователя root. Вы должны внимательно проанализировать уровень привилегий, на котором следует запускать программы. Например, команды mount и umount всегда следует выполнять от лица пользователя root, если только в файле / etc/fstab не присутствует запись, разрешающая выполнять монтирование и демонтирование от лица некоторого обычного пользователя.

Общая папка Homes

Общая папка Homes выполняет специальную функцию. Для каждого пользователя, обращающегося к этой папке, система Samba сравнивает имя пользователя с содержимым файла /etc/passwd и таким образом определяет домашний каталог этого пользователя. Содержимое этого домашнего каталога делается доступным для пользователя в виде содержимого общей папки, при этом пользователь не имеет никакого доступа к домашним каталогам остальных пользователей. Каждый пользователь сервера Samba получает в свое распоряжение индивидуальную общую папку, недоступную для остальных пользователей Samba. Логично предоставить пользователям доступ к подкаталогу домашнего каталога, в этом случае вы сможете избежать проблем, связанных с различными схемами именования файлов в средах Linux и Microsoft. Общая папка Homes является единственной общей папкой, которая по умолчанию не указывает на каталог /tmp, а вместо этого автоматически указывает на домашний каталог пользователя.

Ограничение доступа к службам

Система Samba обладает множеством возможностей, позволяющих гибко контролировать уровень привилегий, которым наделяется клиент Samba в отношении системы Linux. Samba позволяет вам воспользоваться преимуществами групп UNIX NIS (если в вашей сети используется система NIS). Помимо многих других способов контроля пользовательских привилегий система Samba позволяет вам использовать, например, следующие возможности:

- система обслуживает пользователя, подключившегося к общей папке, так, как будто он действует от лица некоторого предопределенного администратором пользователя системы Linux (например, guest, nobody, root, samba или любого другого пользователя Linux);

- система обслуживает пользователя, подключившегося к общей папке, так, как будто он является членом некоторой предопределенной администратором группы;

- Samba позволяет администратору определить имена пользователей, которым разрешается доступ к общей папке; остальным пользователям доступ будет запрещен;

- Samba позволяет администратору определить имена пользователей, которые будут обладать в отношении общей папки административными полномочиями;

- Samba позволяет администратору определить имена пользователей, которые будут обладать правом осуществлять, как чтение, так и запись файлов общей папки;

- Samba позволяет администратору определить имена пользователей, которые будут обладать только возможностью чтения файлов общей папки (запись файлов общей папки для них будет запрещена);

- Samba позволяет администратору определить имена пользователей, которые будут обладать правом печати.

Все перечисленные возможности настраиваются при помощи режима Advanced View конфигурационной страницы Share в разделе Security Options.

Используя все эти возможности, вы можете обеспечить очень гибкий контроль подключений пользователей к любой из общих папок сервера Samba. Также на странице Global вы можете настроить параметр root directory (корневой каталог), указав с его помощью любой удобный для вас каталог, отличающийся от корневого каталога по умолчанию (с именем «/»).

ПРИМЕЧАНИЕ

Если вы делаете корневым каталогом для Samba каталог, отличающийся от каталога с именем /, вы должны разместить в указанном вами каталоге все файлы, необходимые для запуска сервера Samba. В частности, в новый корневой каталог следует скопировать все бинарные файлы и все библиотеки, используемые пользователями Samba. Таким образом, вы создаете для Samba рабочую среду с измененным корнем, которую также называют тюрьмой с измененным корнем (change root jail). При создании такой тюрьмы следует принимать во внимание все соображения и все нюансы, о которых рассказывалось в главе 12, в которой была подробно рассмотрена процедура создания

тюрьмы с измененным корнем для службы DNS.

Samba позволяет вам создать рабочую среду с измененным корнем подобно тому, как это позволяет делать служба DNS (об этом рассказывалось ранее, в главе 12). При этом следует руководствоваться теми же самыми принципами, однако вам потребуется скопировать большее количество файлов. Если сравнивать с DNS, создание тюрьмы с измененным корнем для Samba осуществляется несколько сложнее, так как для этого вы должны будете скопировать в новый корневой каталог большее количество библиотек, устройств, конфигурационных и бинарных файлов.

Однако у этого подхода есть существенное преимущество: в новом корневом каталоге, который выполняет функции тюрьмы, содержатся копии абсолютно всех необходимых файлов, включая файл /etc/password, и любой из этих файлов вы можете модифицировать так, как посчитаете нужным. Это значит, что вы можете использовать совершенно другой набор паролей. Благодаря этому, даже если злоумышленник сможет взломать рабочую среду с измененным корнем, это не будет автоматически означать, что он взломал всю систему Linux.

Создание для Samba рабочей среды с измененным корнем может осуществляться по-разному — все зависит от конфигурации Samba и бинарных файлов, которые вы хотели бы сделать доступными для пользователей. Приведу общие рекомендации на эту тему.

Каталог /etc: здесь потребуется скопировать все имеющие отношение к Samba конфигурационные файлы, а также большую часть подкаталога pam.d/, а также те файлы в каталоге /etc, на которые ссылается файл pam.d. Также потребуется скопировать все файлы, имеющие отношение к аутентификации пользователей при подключении их к системе (passwd, group, shadow), а также все связанные с этим конфигурационные файлы, равно как и модифицированный файл syslog.conf с добавлением к syslog ключа -a, чтобы syslog мог читать этот конфигурационный файл.

Каталог /home: домашние каталоги пользователей, если они используются пользователями (если создана общая папка Homes).

Каталог /dev: любые устройства, которые могут потребоваться пользователям Samba, а также соответствующие им записи <корень Samba>/etc/fstab, чтобы предоставить пользователям возможность монтировать эти устройства.

Каталог /lib: библиотеки, необходимые для запуска бинарных файлов, а также подкаталог безопасности для PAM.

Каталоги /bin и /sbin: все относящиеся к Samba бинарные файлы, а также бинарные файлы для других необходимых программ, таких как mount (на случай, если пользователи должны обладать возможностью монтировать компакт-диски, гибкие диски или другие подобные устройства).

Каталог /var: подкаталог журнала и рабочий подкаталог (а также, возможно, другие подкаталоги).

Каталог /tmp: временный каталог с необходимыми привилегиями доступа (chmod 1777).

Хочу еще раз заострить ваше внимание на том, что в зависимости от конфигурации Samba для создания тюрьмы с измененным корнем может потребоваться копирование некоторых других файлов и каталогов.

Переменные, используемые в Samba

Работая с инструментом администрирования SWAT, вы можете обратить внимание на то, что при указании значений многих параметров используются многочисленные переменные (переменные используются на страницах Advanced View). Переменная обозначается сочетанием символов %X, где X — это либо заглавная, либо строчная латинская буква. Переменные используются для обеспечения большей гибкости Samba. Вот перечень этих переменных с пояснениями:

- %S — имя текущей службы (если такая есть);
- %P — корневой каталог текущей службы (если такая есть);
- %u — имя пользователя, от лица которого работает текущая служба (если такая есть);
- %g — имя первичной группы для пользователя %u;
- %U — имя пользователя для текущего рабочего сеанса (имя пользователя, от лица которого хотел бы работать клиент, это имя может отличаться от имени, которое будет назначено клиенту при подключении);
- %G — имя первичной группы для пользователя %U;
- %H — домашний каталог для пользователя %u;
- %v — версия Samba;
- %h — доменное имя (DNS-имя) узла, на котором работает Samba;

- %m — имя NetBIOS клиентского компьютера;

- %L — имя NetBIOS сервера; благодаря использованию этой переменной вы можете менять конфигурацию сервера Samba в зависимости от того, какое имя использует клиент для того, чтобы обратиться к этому серверу, иными словами, один сервер может быть «многоликим»;

- %M — доменное имя (DNS-имя) клиентского компьютера;

- %N — имя сервера домашнего каталога NIS (извлекается из записи auto.map системы NIS; без NIS значение этой переменной равно значению переменной %L);

- %p — путь к домашнему каталогу службы (извлекается из записи auto.map системы NIS; запись auto.map системы NIS имеет следующий вид: %M:%p);

- %R — выбранный уровень протокола после того, как стороны договорились о том, какой уровень будет использоваться; переменная может быть равна одному из значений: CORE, COREPLUS, LANMANl, LANMAN2 или NT1;

- %d — идентификатор процесса PID для текущего серверного процесса;

- %a — программная архитектура удаленного компьютера; определению подлежат лишь некоторые из архитектур, и то данный механизм нельзя считать в достаточной степени надежным; допустимые значения: Samba, WfWg, WinNT и Win95, остальные архитектуры идентифицируются значением UNKNOWN;

- %1 — IP-адрес клиентского компьютера;

- %T — текущая дата и время.

Конфигурационная страница Global

В листинге 19.1 показан перечень глобальных конфигурационных параметров и их значения по умолчанию. Наиболее часто модифицируемые значения располагаются на странице Basic View, однако многие другие часто используемые параметры скрыты. Полный перечень параметров впечатляет своими размерами: в раздел, связанный с принтерами, входит 25 значений, а раздел общих папок содержит в себе более чем в два раза больше. Всего существует более 190 настраиваемых параметров. Некоторые параметры являются взаимоисключающими. Некоторые параметры влияют на значения других параметров или зависят от их значений.

СОВЕТ

Глобальные параметры действуют в отношении всех общих папок, включая специальные общие папки. Значения глобальных параметров могут быть перекрыты значениями аналогичных параметров, относящихся к конкретной папке.

Листинг 19.1. Перечень глобальных параметров Samba и их значений в том виде, в котором он записывается на диск средством администрирования SWAT

```
workgroup = WORKGROUP
netbios name =
netbios aliases =
server string = Samba 2.0.0 beta5
interfaces =
bind interfaces only = No
security = USER
encrypt passwords = No
update encrypted = No
use rhosts = No
map to guest = Never
null passwords = No
password server =
smb passwd file = /usr/local/samba/private/smbpasswd
hosts equiv =
root directory = /
passwd program = /bin/passwd
passwd chat = *old*password* %o\n *new*password* %n\n *new*password* %n\n *changed*
passwd chat debug = No
username map =
password level = 0
username level = 0
unix password sync = No
```

log level = 1
syslog = 1
syslog only = No
log file =
max log size = 5000
timestamp logs = Yes
protocol = NT1
read bmpx = Yes
read raw = Yes
write raw = Yes
nt smb support = Yes
nt pipe support = Yes
announce version = 4.2
announce as = NT
max mux = 50
max xmit = 65535
name resolve order = lmhosts host wins beast
max packet = 65535
max ttl = 259200
max wins ttl = 518400
min wins ttl = 21600
time server = No
change notify timeout = 60
deadtime = 0
getwd cache = Yes
keepalive = 300
lpq cache time = 10
max disk size = 0
max open files = 10000
read prediction = No
read size = 16384
shared mem size = 1048576
socket options =
stat cache size = 50
load printers = Yes
printcap name = /etc/printcap
printer driver file = /usr/local/samba/lib/printers.def
strip dot = No
character set =
mangled stack = 50
coding system =
client code page = 850
stat cache = Yes
domain groups =
domain admin group =
domain guest group =
domain admin users =
domain guest users =
machine password timeout = 604800
add user script =
delete user script =
logon script =
logon path = \\%N\%U\profile
logon drive =
logon home = \\%N\%U
domain logons = No
os level = 0
lm announce = Auto
lm interval = 60
preferred master = No
local master = Yes
domain master = No
browse list = Yes
dns proxy = Yes
wins proxy = No
wins server =
wins support = No
kernel oplocks = Yes
ole locking compatibility = Yes
smbrun = /usr/local/samba/bin/smbrun
config file =
preload =
lock dir = /usr/local/samba/var/locks
default service =

message command =
dfree command =
valid chars =
remote announce =
remote browse sync =
socket address = 0.0.0.0
homedir map =
time offset = 0
unix real name = No
NIS homedir = No
panic action =
comment =
path =
alternate permissions = No
revalidate = No
username =
guest account = nobody
invalid users =
valid users =
admin users =
read list =
write list =
force user =
force group =
read only = Yes
create mask = 0744
force create mode = 00
directory mask = 0755
force directory mode = 00
guest only = No
guest ok = No
only user = No
hosts allow =
hosts deny =
status = Yes
max connections = 0
min print space = 0
strict sync = No
sync always = No
print ok = No
postscript = No
printing = bsd
print command = lpr -r -P%p %s
lpq command = lpq -P%p
lprm command = lprm -P%p %j
lppause command =
lpresume command =
queuepause command =
queueresume command =
printer name =
printer driver = NULL
printer driver location =
default case = lower
case sensitive = No
preserve case = Yes
short preserve case = Yes
mangle case = No
mangling char = ~
hide dot files = Yes
delete veto files = No
veto files =
hide files =
veto oplock files =
map system = No
map hidden = No
map archive = Yes
mangled names = Yes
mangled map =
browseable = Yes
blocking locks = Yes
fake oplocks - No
locking = Yes
oplocks = Yes
strict locking = No

share modes = Yes
copy =
include =
exec =
postexec =
root preexec =
root postexec =
available = Yes
volume =
fstype = NTFS
set directory = No
wide links = Yes
follow symlinks = Yes
dont descend =
magic script =
magic output =
delete readonly = No
dos filetimes = No
dos filetime resolution = No
fake directory create times = No

Безопасность при локальном использовании Samba

Изучив материал данной главы, вы, должно быть, получили представление о том, что сервер Samba может стать причиной проблем в локальной сети Microsoft, так как он может нарушить нормальное взаимодействие между узлами Microsoft. В большинстве случаев под угрозой оказывается механизм обозрения сетевых ресурсов Windows. Целые группы компьютеров могут неожиданно исчезнуть из списков доступных систем. Подобные неполадки могут возникнуть на всех системах Windows, входящих в состав сети. Основным подозреваемым в подобных ситуациях следует считать сервер Samba, который инициировал процедуру выбора главного обозревателя и выиграл борьбу за роль главного обозревателя сети, в то время как он не должен был этого делать. Проблемы могут возникнуть также в случае, если сервер Samba самопроизвольно приступил к выполнению функций сервера WINS, в то время как эти функции должен выполнять сервер NT.

Если проблемы обозрения ресурсов возникли только лишь на сервере Samba, причиной этого может быть неправильная настройка разрешений или попытка использования нешифрованных паролей или отсутствие в конфигурации IP-адреса сервера WINS. В мире Windows лишь очень небольшие подсети могут обойтись без использования сервера WINS. Однако если в сети нет такого сервера, в процессе обозрения сетевых ресурсов Microsoft у Samba могут возникнуть проблемы. В крупных сетях без серверов WINS не обойтись, особенно если сеть состоит из нескольких широковещательно-изолированных подсетей. Всегда, когда это возможно, в изолированных подсетях в качестве резервных серверов WINS следует использовать серверы NT, хотя на самом деле с этой задачей может справиться и сервер Samba.

Огромное количество зачастую малопонятных и плохо документированных параметров может стать причиной того, что система Samba настраивается неправильно. Благодаря большому количеству параметров сервер Samba представляет собой чрезвычайно гибкую систему, однако у малоопытных администраторов могут возникнуть проблемы. Значения, присваиваемые параметрам в рамках конфигурации по умолчанию, в большинстве случаев считаются «безопасными» — они не нарушают работу других систем Microsoft, и при этом позволяют сетевому узлу Samba войти в состав сети Microsoft, будь это одноранговая сеть или домен NT. Однако вхождение в состав домена NT может быть опасным. Став частью домена NT, сервер Samba может отобрать право контроля над сетью у сервера PDC, поэтому, добавляя сервер Samba в домен NT, следует быть особенно внимательным.

Благодаря механизмам защиты, действующим в любом из доменов NT, сервер Samba, включенный в состав такого домена, вряд ли может стать причиной нарушения безопасности домена, однако он вполне может нарушить работу механизма обзора сетевых ресурсов. Значительно большую угрозу с точки зрения безопасности сервер Samba представляет для одноранговых сетей Windows 9x. Дело в том, что в сетях, состоящих только из узлов Windows 9x, сервер Samba может самопроизвольно взять на себя функции сервера NT.

Еще одним вопросом, подлежащим тщательному рассмотрению в случае, если вы имеете дело с сетью Microsoft, является вопрос о том, будет ли протокол TCP/ IP единственным протоколом в сети или в сети будут использоваться как TCP/ IP, так и NetBEUI. В использовании обоих протоколов нет необходимости. Если вы используете TCP/IP, вы можете полностью отказаться от NetBEUI.

Использование в сети каких-либо других протоколов помимо IP увеличивает передаваемый через сеть трафик. NetBIOS, как стандарт, основанный на широковещательной передаче данных, будет использовать оба доступных протокола как для широковещательных сообщений, так и при поиске узла в

сети. Если вы сохраните в вашей сети только TCP/IP, вы добьетесь двух результатов: во-первых, вы ограничите широковещательный трафик рамками только одного протокола, во-вторых, вы сможете обеспечить передачу NetBEUI через IP за границами вашего брандмауэра.

Еще одной областью, требующей внимания, является уровень доступа, которым система Samba наделяет пользователей сервера Samba. Я не рекомендую наделять пользователей привилегиями администратора общей папки или привилегиями более широкими, чем те привилегии, которыми обладали бы эти пользователи, будь они обычными пользователями данной системы Linux. В крупных сетях с большим количеством департаментов и многочисленными пользователями рекомендуется использовать Samba в рабочей среде с измененным корнем, так как в этом случае вы сможете предоставить избранным пользователям более широкие полномочия, не опасаясь при этом нарушить безопасность всей системы Linux. Конечно, задачу запуска Samba в рабочей среде с измененным корнем нельзя назвать тривиальной, однако зачастую данный подход более чем оправдан с точки зрения безопасности.

Безопасность Samba при соединении подсетей через Интернет

Основная проблема подобных сетевых конфигураций состоит не в том, что Samba используется для обмена данными через публичные каналы связи Интернета, а в том, что механизмы VPN (Virtual Private Networking), предлагаемые компанией Microsoft, нельзя считать в полной мере безопасными. Последние исследования в этой области показывают, что, являясь *виртуальной* (virtual), сеть VPN, основанная на продуктах Microsoft, не является *частной* (private). Конечно, узлы Microsoft шифруют передаваемые через сеть пароли, однако базовая информация, передаваемая между подсетями через Интернет, не защищена достаточным образом.

Конечно, злоумышленники не смогут проникнуть в одну из ваших подсетей, выдавая себя за один из внутренних сетевых узлов, однако они обладают возможностью читать содержимое передаваемых между сетями пакетов. Таким образом, если вы передаете закрытую информацию, имейте в виду, что она может быть перехвачена и прочитана кем угодно. Чтобы создать действительно безопасную сеть VPN между узлами Microsoft, следует воспользоваться методом, описанным в главе 21, где рассматривается процедура создания шифрованного канала VPN между шлюзами Linux. Чтобы исключить просачивание каких-либо пакетов вне границ созданного вами тоннеля, в обеих сетях следует использовать только IP.

Формируя подсети, имейте в виду, что в каждой из них должен присутствовать локальный главный обозреватель, иначе системы не смогут получать информацию о сетевых ресурсах, расположенных в других подсетях. Если в вашем распоряжении три подсети с локальными обозревателями, осуществляющими обновление главного обозревателя, учтите, что между появлением в одной из удаленных подсетей нового сетевого узла и моментом появления этого узла в списке узлов, доступных для локальных систем, может пройти до 45 минут. По этой причине, прежде чем обвинять сервер Samba в том, что он отказывается показывать все узлы, присутствующие в домене NT, убедитесь в том, что все интересующие вас удаленные системы присутствуют в домене большую часть часа (более 45 минут).

Заключение

В данной главе я рассказал вам о сервере Samba и программе администрирования swat Вы узнали о проблемах безопасности, связанных с использованием swat, а также о том, как защитить бинарный файл swat, ограничив доступ к локальному узлу localhost (при помощи ipchains или tcpd), а также о том, как безопасно администрировать Samba через сеть с использованием сервера Apache и механизмов SSL. Вы также узнали о вопросах безопасности, связанных с использованием протокола NetBIOS, а также о том, какие проблемы могут возникнуть из-за того, что Samba работает от лица пользователя root. Я также рассказал вам о том, каким образом Samba может стать причиной беспорядка и хаоса в локальных сетях Microsoft. Вы также узнали об опасностях, которые могут подстерегать вас при использовании Samba в сетях, соединенных каналами Интернета.

20 Установка и запуск web-сервера Apache

В данной главе рассматриваются следующие вопросы:
- сборка Apache с добавлением SSL и PHP3;
- настройка Apache;
- использование файлов .htaccess совместно с AuthConfig;
- использование khttpd.

В наши дни web-сервер Apache является наиболее популярным web-сервером Интернета, и это не случайно, ведь Apache является наиболее мощным и наиболее гибко конфигурируемым web-сервером из всех существующих. Прочитав предыдущие главы данной книги, вы, должно быть, уже знаете, что за мощь, обилие возможностей и высокую гибкость приходится платить потенциальным снижением уровня защиты. Сервер Apache не является исключением. Однако, как будет показано в данной главе, на самом деле, выполнив должным образом конфигурирование, вы сможете обеспечить высокий уровень защиты вашего web-сервера, сохранив при этом все его преимущества.

Сборка Apache

Web-сервер Apache в том виде, в котором он устанавливается в вашей системе по умолчанию, уже является отличным инструментом. Однако в большинстве поставок Linux используется реализация Apache, в которой отсутствуют некоторые чрезвычайно важные механизмы (имейте в виду, что в будущем ситуация может измениться). Большинство комплектов Linux содержит в себе Apache в базовой конфигурации. Базовая конфигурация вполне подходит для обработки самых простых web-страниц и запуска некоторых CGI-приложений, однако если вы хотите обеспечить более высокий уровень защиты или намерены использовать Apache совместно с той или иной базой данных, вы очень быстро поймете, что без перекомпиляции Apache вам не обойтись.

Компиляция таких приложений, как Apache, — это несложный прямолинейный процесс, однако если вы намерены добавить в систему дополнительные возможности, например пакеты шифровки или другие пакеты, разработанные сторонними производителями, процедура компиляции несколько усложняется.

ПРИМЕЧАНИЕ

Прежде чем приступать к компиляции какой-либо программы, вы должны убедиться в том, что у вас есть все необходимые для этого библиотеки и программы. Вам потребуется несколько devel-пакетов RPM включая glibc-devel, libpam-devel и, возможно, devel-библиотеки, имеющие отношение к XFree. Вам также потребуется gcc, относящиеся к gcc библиотеки (libstdc++) и devel-пакеты, а также пакет as86. Возможно, вам потребуются также другие пакеты, о чем вы узнаете из сообщений об ошибках. Пакеты devel необходимы только для компиляции, позже вы можете удалить их из системы. Я также рекомендую добавить пакеты Perl.

В данной главе подразумевается, что вы намерены добавить в Apache поддержку SSL (Secure Sockets Layer) и PHP — это два чрезвычайно популярных пакета. Для обеспечения поддержки SSL мы будем использовать пакет mod_ssl (имейте в виду, что это не единственный пакет SSL, который можно использовать совместно с Apache). Пакет mod_ssl позволяет создавать защищенные шифрованные соединения между Apache и web-клиентом (глубина шифровки — 40 бит или 128 бит — зависит от клиента). Пакет PHP будет откомпилирован с поддержкой MySQL, поэтому вам потребуется установить пакеты MySQL и MySQL-devel. Если вас не интересует взаимодействие с базой данных, вы можете не обращать внимания на любые ссылки на PHP. Если вы намерены использовать PHP для взаимодействия с какой-либо другой базой данных (msql или Postgres), просто замените все ссылки на MySQL ссылками на интересующую вас БД. Конечно же, при этом вы должны установить соответствующие пакеты devel.

ВНИМАНИЕ

Библиотеки RSA содержат в себе некорректный код, из-за которого безопасность вашего сервера может быть нарушена. Подумайте о том, действительно ли вы должны использовать эту библиотеку, ведь в течение года патент будет обновлен. Если все-таки вам не обойтись без использования библиотек RSA, свяжитесь с RSA для того, чтобы получить исправленную версию библиотеки.

Получение необходимых пакетов

Самые свежие версии всех необходимых пакетов можно получить по адресам, перечисленным далее. Здесь я указываю номера версий, которые являются наиболее свежими на момент написания книги. При наличии таковых вы можете использовать более свежие версии пакетов.

Apache-1.3.9 (apache_1.3.9.tar.gz): ftp://ftp.apache.org/apache/dist/
PHP-3.0.13 (php-3.0.12.tar.gz): ftp://ftp.php.net/pub/distributions/php-3.0.12.tar.gz
mod_ssl-2.4.9 (mod_ssl-2.4.8-1.3.9.tar.gz): ftp://ftp.modssl.org/source/
OpenSSL-0.9.4 (openssl-0.9.4.tar.gz): ftp://ftp.openssl.org/source/
RSAref-2.0 (rsaref20.tar.Z): ftp://utopia.hacktic.nl/pub/replay/pub/crypto/LIBS/rsa/
Этот пакет больше нельзя получить непосредственно от RSA.
MM-1.0.12 (mm-1.0.12.tar.gz): http://www.engelschall.com/sw/mm/

ПРИМЕЧАНИЕ

Цифры 1.3.9 в имени пакета mod_ssl-2.4.8-1.3.9.tar.gz должны соответствовать номеру версии сервера Apache. Другими словами, если появляется версия Apache 1.3.10 и вы используете на своем компьютере именно эту версию, вы должны добавить в него пакет mod_ssl-x.x.x-1.3.10.tar.gz (используйте самую свежую версию этого пакета).

Несколько слов о шифровании и ограничениях экспорта, действующих в США

Информация, приведенная в данной врезке, не должна рассматриваться как достоверный юридический материал. Приведенные здесь сведения основаны на дискуссиях с аналитиками в области экспорта технологий кодирования, работающими в Департаменте Торговли Соединенных Штатов Америки. Если вы планируете предпринимать какую-либо серьезную деятельность в области технологий кодирования, вы должны проконсультироваться с профессиональным юристом.

Должно быть, вы обратили внимание на то, что ни один из упоминаемых в данной главе пакетов, связанных с шифрованием данных, не записан на прилагаемом к книге компакт-диске. Это сделано для того, чтобы не нарушить действующее в США законодательство об ограничениях экспорта кодирующих и шифрующих технологий. Технологии кодирования, которые в наше время важны как для домашнего, так и для делового использования, рассматриваются в рамках данного законодательства как военная амуниция, благодаря чему в их отношении действуют строгие правила ограничения экспорта. Любую технологию кодирования разрешается импортировать в США, однако ни одна технология кодирования не может экспортироваться или реэкспортироваться из США. Таким образом, вы можете загрузить любые связанные с кодированием файлы из любой сколь угодно далекой страны, например из Финляндии, однако после того как они попали на территорию Соединенных Штатов, вы не имеете права реэкспортировать их. Мало того, существующее законодательство запрещает гражданам США, путешествующим за границу, предоставлять технологии кодирования для граждан иностранных государств. Иными словами, если вы, будучи гражданином США, уезжаете за границу, там загружаете из Финляндии файлы, связанные с кодированием, а затем компилируете их для кого-либо, живущего в той стране, где вы находитесь, вы также нарушаете законодательство США. Библиотеки кодирования доступны в любом месте земного шара, где есть Интернет, однако если вы являетесь гражданином США и если вы компилируете шифрующую программу для японской компании в Японии (вместо Японии можно подставить в это предложение любую другую страну), вы можете попасть под суд. Мало того, под суд можно попасть даже в случае, если вы компилируете шифрующую программу для японской компании, расположенной на территории Соединенных Штатов. Теперь представьте, что вы при помощи telnet подключаетесь к серверу, расположенному в другой стране, и устанавливаете на этом сервере SSH или какую-либо другую программу, осуществляющую кодирование. Нарушение законодательства США? Именно! Столь жесткий ограничивающий закон превратил Америку из лидера в области криптографии в аутсайдера. Вряд ли когда-либо Соединенные Штаты смогут восстановить свои позиции в этой области.

Законодательство, связанное с криптографией, постоянно меняется, поэтому я рекомендую вам в случае возникновения вопросов связаться с представителями Департамента Торговли США.

Если вы являетесь гражданином США, вы можете добавить пакет RSA (если вы планируете коммерческое использование пакета SSL, вы обязаны использовать RSA). RSA обладает патентом на некоторые алгоритмы, используемые в большинстве криптографических пакетов. За пределами Соединенных Штатов все эти алгоритмы разработаны независимо и свободно доступны для всех желающих, однако на территории США, если вы не включаете в состав приложения пакет RSA, вы нарушаете американское патентное законодательство. Все остальные страны мира отказались от поддержки этого патента. Вы можете добавлять или не добавлять пакеты RSA на свой страх и риск. В книге я описываю эти пакеты для тех, кто вынужден использовать их.

Предварительная подготовка

Прежде чем приступить к сборке Apache, необходимо подготовить место, где вы сможете открыть и откомпилировать все необходимые пакеты. Некоторые из этих пакетов уже могут находиться в вашей системе, так как вы могли использовать их для компиляции других пакетов. Если вы уже откомпилировали и установили в системе некоторые из пакетов, вы можете настроить некоторые переменные среды и конфигурационные переменные таким образом, чтобы использовать имеющиеся в системе пакеты. Об этом будет рассказано далее в тексте. Для простоты изложения я предполагаю, что вы размещаете все необходимые пакеты в одном общем каталоге, откуда будет осуществляться их сборка и установка.

Подыщите раздел, на котором имеется достаточно свободного пространства (чуть более 150 Мбайт), создайте там каталог (в данном тексте предполагается, что каталог имеет имя $HOME/src/) и скопируйте в него все пакеты. После этого перейдите в этот каталог и откройте каждый из пакетов. Для этого в

отношении каждого пакета следует выполнить команду tar xzvf имя_файла. Исключение составляет пакет rsaref20.tar.Z. Для того чтобы открыть этот пакет, необходимо вначале создать каталог rsaref-2.0, затем перейти в этот каталог, а затем выполнить команду tar xzvf . ./rsaref20.tar.Z.

После того как все пакеты будут открыты, убедитесь в том, что на диске осталось достаточно свободного места для того, чтобы выполнить компоновку этих пакетов (должно оставаться более 100 Мбайт).

Компоновка пакетов

На первом этапе следует выполнить конфигурирование Apache. Этот этап необходим из-за того, что компоновка некоторых других пакетов (прежде всего PHP) выполняется в соответствии с имеющейся конфигурацией Apache. Начать лучше всего с использования тех же самых конфигурационных значений, которые используются в комплекте Caldera для входящего в его состав пакета Apache RPM. Таким образом, если вы установили Apache RPM, установка будет выполнена поверх существующих файлов. Содержимое листинга 20.1 извлечено из файла apache.spec.

Листинг 20.1. Конфигурация из файла apache.spec

```
./configure \
--prefix=/etc/httpd \
--disab1e-rule=WANTHSREGEX  \
--sysconfdir=/etc/httpd/conf \
--datadir=/home/httpd \
--bindir=/usr/bin \
--sbindir=/usr/sbin \
--libexecdir=/usr/libexec/apache \
--includedir=/usr/include/apache \
--logfiledir=/var/log/httpd \
--localstatedir=/var \
--runtimedir=/var/run \
--proxycachedi r=/var/cache/httpd \
--mandir=/usr/man \
--enable-module=most \
--enable-shared=max
```

Поместите содержимое листинга 20.1 в файле (apache.conf) в каталоге apache_1.3.9, сделайте файл исполняемым при помощи команды chmod 755 apache.conf, а затем выполните этот файл (./apache.conf). Вы сможете модифицировать конфигурацию позже. Большая часть параметров, приведенных в листинге 20.1, изменяет места размещения бинарных файлов по умолчанию. Три влияют на процесс сборки и один (--disable-rule=WANTHSREGEX) необходим, иначе сборка окончится неудачей. Позже вы еще вернетесь к этому файлу и добавите в него дополнительные записи.

Прежде чем продолжить, вы должны проверить, надо ли добавлять какие-либо другие модули (например, mod_dav — модуль ревизии авторства), и если да, то вы должны установить эти модули. Если для установки модуля требуется скомпоновать Apache, откажитесь от этой возможности.

Второй этап не обязателен, но рекомендуется в случае, если вы хотите добавить поддержку доступа к базе данных. Если вы нуждаетесь в доступе к базе данных, вначале следует установить пакеты RPM, относящиеся к MySQL. Если вы намерены использовать какую-либо другую базу данных, обратитесь к электронной справке ./configure -help и далее подставьте вместо mysql имя вашей базы данных. В листинге 20.2 показана приемлемая базовая конфигурация для PHP. Значения некоторых параметров можно изменить либо из командной строки, либо при помощи файла php.ini. Если вы не загрузили RPM-пакет imap-devel, вы можете удалить поддержку imap.

Листинг 20.2. Конфигурация PHP

```
./configure \
--with-apache=../apache_1.3.9 \
--with-mysql=/usr \
--with-imap \
--enable-sysvshm=yes \
--enable-sysvsem=yes \
--with-config-file-path=/etc \
--enable-debug=no \
--enable-track-vars=yes \
```

Вы можете скопировать содержимое листинга в файл, расположенный в каталоге php-3.0.12, сделать этот файл исполняемым (chmod 755), а затем запустить его. После этого просто выполните команду make && make install для того, чтобы скомпоновать и установить модуль php в иерархии исходных файлов apache. Настройку Apache для компоновки совместно с PHP следует выполнить позже.

СОВЕТ

Команда, при помощи которой рекомендуется выполнить сборку PHP (make && make install), на самом деле включает в себя две отдельные команды. Комбинация символов && указывает на то, что две команды должны быть выполнены одна за другой, однако вторая команда выполняется только в случае, если первая была выполнена успешно. Вы можете разделить две команды символом точки с запятой (;), однако при этом вторая команда будет выполнена после первой вне зависимости от того, насколько успешным было выполнение первой команды. Я рекомендую использовать символы &&, так как при этом выполнение будет прервано в случае, если команда make не сработала. При этом, исходя из отображенной на экране информации, вы сможете определить причину проблемы.

Следующий этап потребуется вам только в случае, если вы собираетесь использовать сервер совместно с SSL на территории США в коммерческих целях. В противном случае данный этап можно пропустить. Я предполагаю, что вы открыли пакет rsaref так, как я упоминал ранее, то есть вы разместили библиотеки RSA в каталоге rsaref-2.0. Перейдите в каталог rsaref-2.0, выполните команду cp -rp install/unix linux. По этой команде каталог unix будет скопирован в установочный каталог под именем linux. После этого перейдите в каталог linux и выполните команду make. После завершения ее работы вы получите файл с названием rsaref .a. Следует скопировать этот файл под другим именем при помощи команды cp rsaref.a librsaref.a, так как в процессе компоновки вместо файла rsaref.a система ищет файл librsaref.a.

Четвертый этап, в отличие от второго и третьего, является обязательным. На этом этапе выполняется компоновка библиотек openssl, которые требуются для работы mod_ssl. Переместитесь в каталог openssl. Здесь вы можете видеть два конфигурационных файла. Вы можете использовать любой из них (оба выполняют одни и те же действия), однако при использовании файла Configure требуется дополнительный аргумент (linux-elf). Все остальные аргументы одни и те же. При желании вы можете установить openssl у вас в системе. По умолчанию (если вы не меняли этого в процессе конфигурирования) openssl устанавливается в каталоге /usr/ local/ssl/. Именно в этом месте программы, нуждающиеся в этом пакете, осуществляют поиск файлов openssl по умолчанию. Таким образом, если вы измените такой порядок вещей, вы должны учитывать это, выполняя компиляцию с использованием openssl в будущем.

Как я уже говорил, в данном тексте описывается предлагаемая конфигурация сборки, однако при желании вы можете изменить ее. Если вы находитесь в Европе, возможно, вы захотите включить no-idea, для того чтобы исключить компоновку шифров idea. Также необязательным является параметр -fPIC, как и rsaref. На данном этапе предлагаемая строка конфигурации выглядит следующим образом:

```
./config -L$(pwd)/../rsaref-2.0/linux/ rsaref -fPIC
```

По этой команде система OpenLinux настраивается на компоновку с использованием RSAref.

ПРИМЕЧАНИЕ

Аргумент -fPIC необходим, если вы осуществляете компоновку с использованием общих библиотек. В отличие от статических библиотек (static libraries) общие библиотеки (shared libraries) являются более предпочтительным вариантом компоновки, за исключением некоторых специальных случаев (например, при работе во время начальной загрузки, когда общие библиотеки недоступны). Имена статических библиотек заканчиваются суффиксом ,a, а в конце имен динамических библиотек размещается суффикс .so.

Когда конфигурация завершена, введите make && make test. По этой команде комплект openssl будет скомпонован и протестирован. Если вы намерены установить этот комплект у себя в системе, перейдите на уровень root (su) и выполните команду make install. Если в вашей системе установлен пакет RSAref, вместо использования ключа -L экспортируйте переменную окружения RSAREF_BASE=/путь/к/rsaref.

Пятый этап, компоновка и установка библиотеки MM (Memory Module — модуль памяти), также является необязательным. Эта библиотека позволяет серверу Apache использовать доступную оперативную память вместо дискового кэша, благодаря чему увеличивается производительность. Этот модуль имеет смысл использовать в случае, если ваш узел обслуживает средний или большой по объему трафик. Чем больше трафик, тем больше вы выигрываете от использования данного модуля, однако даже узлы с небольшим по объему трафиком могут получить преимущества от использования данного модуля, при этом уровень безопасности фактически не снижается. Чтобы скомпоновать библиотеку MM, перейдите в каталог mm. Если вы хотите установить MM в вашей системе (а я рекомендую вам сделать это), просто выполните:

```
./configure && make
```

При этом будет выполнено автоматическое конфигурирование и сборка библиотек MM. После этого вы можете перейти на уровень привилегий root и выполнить команду make install. Эти действия приведут к тому, что библиотеки будут установлены в вашей системе так, как если бы вы передали configure параметр --prefix=/ usr/local. Иными словами, файлы будут установлены в подкаталогах /usr/local/lib, /usr/local/bin, /usr/local/indude и т. д. Если же вы устанавливаете файлы в каталогах /usr/lib, /usr/bin и т. д., вы должны

отдать ранее приведенную команду configure с параметром -prefix=/usr.

После того как вы установили библиотеку, убедитесь в том, что файл /etc/ ld.so. conf содержит строку, ссылающуюся на библиотеку (по умолчанию /usr/local/ lib), после чего на уровне привилегий root выполните команду Idconfig. По этой команде система распознает общие библиотеки MM. Если вы не хотите устанавливать в системе библиотеки MM, однако при этом желаете, чтобы сервер Apache смог воспользоваться этими библиотеками, настройте MM с использованием параметра --disable-shared, благодаря этому Apache не будет производить поиск общих библиотек.

Шестой (и предпоследний) этап является обязательным. На этом этапе вы осуществляете компоновку модуля SSL. Этот этап является наиболее простым, так как для его выполнения требуется выполнить всего одну команду: configure, которой необходимо передать один или три аргумента --with-apache=../apache_1.3.9, --with-crt=/путь/к/вашему/серверу.crt и --with-key=/путь/к/вашему/серверу.key. Первый аргумент является обязательным, а два других используются в случае, если вы обладаете серверным сертификатом (см. далее). В данном тексте подразумевается, что у вас нет серверного сертификата. В этом случае команда будет выглядеть следующим образом:

```
./configure  --with-apache=../apache_1.3.9
```

Когда эта команда будет выполнена, данный этап можно считать завершенным. Отображаемые на экране инструкции по поводу компоновки Apache можно игнорировать, так как в нашем случае компоновка подразумевает более обширный набор процедур.

Седьмой и последний этап — это финальная конфигурация Apache с последующей компоновкой и установкой его в системе. В следующем разделе речь пойдет о конфигурации httpd и запуске. Чтобы начать, переместитесь в каталог apache_1.3.9.

Помните ваш файл apache.conf? Теперь настало время добавить в него новые записи. Прежде чем вы это сделаете, возможно, будет лучше определить, какие модули в настоящий момент включены или отключены. Для того чтобы узнать это, запустите ./configure -help. Вам потребуется включить модуль SSL, однако обратите внимание также на другие модули, которые вы хотите включить или отключить. Вне зависимости от того, состояние каких модулей вы намерены изменить, в файл следует добавить строки из листинга 20.3. Прежде чем добавить в файл дополнительные параметры, не забудьте добавить в конец последней строки существующего файла (enable-shared=max) символ обратной косой (\).

Листинг 20.3. Минимальные добавления к содержимому листинга 20.1, apache.conf

```
--enable-module=so \
--enable-module=ssl  \
--acti vate-module=src/modules/php3/l ibphp3.a
```

К некоторым другим полезным модулям, которые можно включить, относятся модули so и info. Если вы планируете следить за обновлениями mod_ssl и при этом не хотите каждый раз заново выполнять компоновку всего сервера, подумайте о том, чтобы включить в конфигурационный файл строку --enable-shared=ssl. В этом случае модуль SSL будет скомпонован в виде общей библиотеки, которую можно будет обновлять отдельно от всего сервера. Вы также можете внести в конфигурацию какие-либо другие изменения, однако имейте в виду, что при добавлении suexec безопасность может оказаться нарушенной.

Прежде чем выполнить файл apache.conf, необходимо настроить некоторые переменные окружения. Набор этих переменных, равно как и их значения, зависят от того, какие действия вы выполняли ранее. Речь идет о трех следующих переменных: SSL_BASE, RSA_BASE и EAPI_MM. Переменная SSL_BASE является обязательной, ей необходимо присвоить местоположение установленного пакета ssl Если вы не устанавливали в своей системе пакет openssl, а только скомпоновали его для совместного использования с Apache, значит, вам надо выполнить примерно следующее присвоение: SSL_BASE=../openssl-0.9.4, в противном случае используйте SSL_BASE=SYSTEM или некоторый специальный путь к установленным пакетам (SSL_BASE=/usr/local/ ssl). Переменной RSA_BASE также можно присвоить значение SYSTEM, но, как правило, эту переменную определяют следующим образом: RSA_BASE=../rsaref-2.0/linux. Переменная EAPI_MM может быть равна либо SYSTEM, либо ../mm-1.0.12, либо ее можно не использовать, в зависимости от того, компонуете ли вы библиотеки MM. После объявления переменных не забудьте экспортировать любые объявленные переменные. Запустите сценарий configure и исправьте любые ошибки (это должны быть ошибки указания пути к RSAref, SSL или MM). Выполните команду make.

Компоновка должна пройти без каких-либо проблем. Однако учтите, что конфигурационный сценарий Apache содержит серьезную ошибку, связанную с egcs, используемым в Caldera. Если компоновка остановится и на экране появится сообщение об ошибке со ссылкой на -rpath, вы должны будете внести изменения в некоторые файлы Makefile в нескольких подкаталогах. Если компоновка завершилась успешно, пропустите следующий абзац.

Чтобы исправить ошибочные файлы Makefile, перейдите в подкаталог src/ modules, после чего перейдите в каждый из подкаталогов: standard, proxy, extra и ssl и в каждом из них измените следующие две строки в файле Makefile (вторая строка приведена частично):

```
LDFLAGS_SHLIB=  -rpath /usr/local/lib  -rpath /usr/lib/mysql   -shared
LIBS1=  -Wl,   -rpath./usr/local/lib  -Wl.   -rpath,/usr/lib/mysql
```

надо заменить на:

```
LDFLAGS SHLIB=  -L/usr/local/lib  -L/usr/local/lib/mysql  -shared
LIBS1= -W1,  -L/usr/local/lib -Wl,  -L/usr/lib/mysql
```

Ключ -rpath, за которым следует пробел или запятая, заменен ключом -L, за которым без пробела следует путь. В каждом из четырех файлов Makefile надо заменить четыре места, где встречается -rpath. Как только вы сделаете это, компоновка будет завершена без каких-либо проблем. Не вносите каких-либо изменений в сами пути! Конечно, используемые в вашей системе пути могут отличаться от тех, которые приведены в данном примере. Ваша задача — изменить только ключ -rpath.

Если у вас нет собственного сертификата, просто введите make certificate и следуйте приглашениям. В большинстве случаев вы можете использовать значения, которые предлагаются вам по умолчанию, однако сведения о сертификате, предлагаемые по умолчанию, следует заменить своими данными. Напоследок система спросит вас, хотите ли вы зашифровать серверный ключ при помощи ключевой фразы. Если вы зашифруете серверный ключ, при каждом последующем запуске сервера Apache в режиме SSL система будет просить у вас ввести ключевую фразу. Если вы хотите, чтобы сервер Apache смог перезапускаться без вашего участия, вам лучше не осуществлять шифровку серверного ключа. Однако, принимая решения, вы должны тщательно взвесить риск. Использование ключевой фразы PEM повысит уровень защиты, но при этом вы должны будете вводить эту фразу каждый раз при запуске сервера SSL.

ПРИМЕЧАНИЕ

Сертификаты безопасности сервера можно приобрести в таких организациях, как Verisign или Thawte Consulting. Если вы не намерены покупать сертификат, вы можете создать свой собственный. Смысл обладания коммерческим сертификатом состоит в том, что, выдавая вам такой сертификат, независимая третья сторона подтверждает, что вы — именно тот, за кого вы себя выдаете. Создавая свой сертификат, вы не тратите лишних денег и все равно получаете возможность шифровать передаваемые через сеть данные, однако при этом ваши клиенты не смогут быть уверенными в подлинности вашей личности. Шифровка данных не может осуществляться без сертификата. Если у вас нет сертификата, значит, вы не сможете шифровать данные. Однако сертификат можно либо купить, либо бесплатно создать собственный. Если вы не собираетесь предлагать пользователям Интернета какие-либо службы, которые должны быть хорошо защищены (например, службы финансовых операций через Интернет), значит, вы вполне можете обойтись сертификатом, который вы можете создать самостоятельно, руководствуясь инструкциями, приведенными в данном тексте.

Теперь перейдите на уровень привилегий root и используйте директиву make install для того, чтобы установить в вашей системе новый web-сервер Apache с поддержкой SSL. После завершения процесса компоновки утилита make сообщит вам, что вы можете запустить Apache с использованием команды apachectl с добавлением start для запуска обычного web-сервера или с добавлением startssl для запуска как нормального сервера, так и сервера SSL. Однако прежде чем запустить Apache, следует настроить кое-какие параметры.

Конфигурирование Apache

В свое время сервер Apache использовал четыре конфигурационных файла: httpd.conf, access.conf, srm.conf и mime.conf. В дальнейшем для удобства файлы access.conf и srm.conf были объединены и включены в состав файла httpd.conf, однако при желании их по-прежнему можно использовать по отдельности. Перейдите в каталог /etc/httpd/conf, где вы можете обнаружить эти три файла, а также подкаталог magic и несколько подкаталогов ssl.*. Для настройки рекомендуется использовать только один файл httpd.conf, и в данной книге мы будем осуществлять настройку именно таким образом.

Файл httpd.conf, входящий в комплект Apache по умолчанию, обладает значительным размером, но в него входит большое количество примеров и объяснений. Однако все эти объяснения недостаточно подробны. В данной книге я не собираюсь подробно объяснять различные записи этого файла. Те, кто заинтересован в дополнительных разъяснениях, могут обратиться к электронной документации Apache, которую можно найти по адресу http://www.apache.org/. Я сконцентрирую ваше внимание на некоторых параметрах, которые связаны с безопасностью вашего web-сервера.

Файл httpd.conf по умолчанию определяет весьма благоразумную с точки зрения безопасности конфигурацию web-сервера. Если вы не внесете в него каких-либо изменений, вы сможете запустить Apache только в нормальном режиме. Если вы запустили ваш web-сервер, не модифицировав предварительно файл httpd.conf (предполагается, что вы используете рассмотренный ранее файл httpd.conf, а не тот файл httpd.conf, который уже существует), при попытке обратиться к серверу с использованием web-браузера, нацеленного на URL-адрес http://localhost/, вы будете разочарованы. Дело в том, что порт, через который ваш web-браузер по умолчанию пытается установить связь, не совпадает с портом, который

по умолчанию связывается сервером Apache. Чтобы исправить ситуацию, вы должны модифицировать файл httpd.conf.

В начале файла httpd.conf большая часть значений по умолчанию вполне приемлема. Обратите внимание на параметры MaxClients и MaxRequestsPerChild. Первый из них ограничивает число клиентов, которым разрешается подключаться к серверу в любой момент времени. По умолчанию параметр имеет значение 150. однако вы можете увеличить или уменьшить это значение таким образом, чтобы оно соответствовало объему памяти, установленному на компьютере. Если вы не знаете точно, какое значение будет наиболее эффективным в вашей ситуации, на некоторое время оставьте параметр без изменений и проанализируйте, какая часть установленного в системе ОЗУ оказывается занятой в процессе подключения к серверу клиентов. Получив представление о нагрузке на сервер, модифицируйте данный параметр должным образом. Важным параметром также является MaxRequestPerChild. По умолчанию значение этого параметра равно нулю (0), что означает, что клиенту разрешается порождать неограниченное количество дочерних запросов. Такое положение дел нельзя считать вполне безопасным, так как система оказывается беззащитной перед атаками типа Denial of Service (DoS). Если вы присвоите этому параметру значение 25, этого, скорее всего, будет вполне достаточно.

Теперь перейдем далее по файлу. Пропустите список модулей и обратите внимание на параметр Port. По умолчанию данный параметр равен 8080, именно поэтому, запустив Apache и попытавшись подключиться к нему при помощи web-браузера, вы не сможете это сделать. Если вы намерены использовать стандартный порт для обеспечения доступа к web-службам, вы должны присвоить параметру Port значение 80. Помните, что при этом сервер Apache должен быть запущен на уровне привилегий root. Если вы хотите, чтобы сервер работал от лица пользователя, не являющегося пользователем root, вы должны связать web-сервер с портом, номер которого превышает 1024. Если вы запускаете Apache на этой стороне брандмауэра (то есть во внутренней сети), вы можете легко добиться этого -достаточно организовать перенаправление порта 80 в порт 8080, а порта 443 в порт 8443. Таким образом, ваш сервер Apache будет работать от лица непривилегированного пользователя, что существенно повысит уровень защищенности системы.

ПРИМЕЧАНИЕ

Ядро Linux 2.4.x обладает новыми механизмами обеспечения безопасности web-серверов, о которых будет рассказано далее, однако сейчас, чтобы соединиться с вашим web-сервером, вы должны либо связать его с портом 80, либо использовать URL http://localhost:8080/.

Несколько дальше в файле размещается раздел, который начинается с метки <IfDefine SSL> и заканчивается меткой </IfDefine>. В этом разделе содержатся два выражения Listen. Эти выражения определяют порты, через которые Apache будет ожидать поступления соединения, если сервер был запущен с аргументом startssl Эти параметры не зависят от параметра Port, который используется в случае, если Apache запущен с аргументом start. Портом незащищенной передачи данных по умолчанию является порт 8080 вместо стандартного 80, а портом защищенной передачи данных по умолчанию является порт 8443 вместо стандартного 443. Если вы намерены связать Apache со стандартными портами, вы должны изменить соответствующие параметры в файле httpd.conf. В противном случае для подключения к вашему серверу клиенты обязаны будут указывать номера нестандартных портов.

ПРИМЕЧАНИЕ

Чтобы подключиться к порту защищенной передачи данных, в графе URL-адреса вы должны указать: https://localhost:8443/. Обратите внимание, что при указании протокола на конце https стоит буква s, что означает secure — защищенный. Клиент (по крайней мере в случае использования Netscape) должен знать о том, что вы пытаетесь соединиться с сервером через защищенный порт, в противном случае соединение установить не удастся, так как через обычный порт данные HTTP передаются в виде обычного текста, а через защищенный порт данные HTTP передаются с использованием специального протокола, обеспечивающего защиту. Иными словами, в первом случае никакой шифровки не выполняется, а во втором случае передаваемые данные шифруются.

Следующими записями, которые представляют для нас интерес, являются записи, в которых определяются пользователь и группа (User and Group), от лица которых будут обслуживаться клиентские запросы. По умолчанию для этой цели используется пользователь и группа с именем nobody (в разных комплектах Linux имя группы может быть разным). В результате этого, если web-сервер запускается на уровне привилегий root (а следовательно, он связывается с портами 80 и 443), дочерние процессы, которые выполняют обслуживание поступающих от клиентов запросов, порождаются от имени пользователя nobody. Если вы взглянете на список процессов (ps aux) httpd, вы заметите, что один из процессов запущен на уровне привилегий root, а другие (по умолчанию пять процессов) работают на уровне привилегий nobody. Дочерние процессы — это те, которые выполняют обслуживание клиентов. Благодаря этому

ограничивается масштаб повреждений, которые эти процессы могут нанести системе. Таким образом, если клиент должен обладать возможностью записи в некоторый каталог, этот каталог должен быть доступен для записи для пользователя nobody. Разрешение на запись в этот каталог должно быть предоставлено для владельца или для группы, но не для всего мира. В отношении каталога, в который разрешено записывать пользователю nobody, должны действовать также и другие ограничения, о которых будет рассказано далее. При желании вы можете создать другого пользователя, от лица которого будут обслуживаться web-клиенты (например, для этой цели можно использовать учетную запись www). Однако для большинства приложений пользователя и группы nobody вполне достаточно.

ПРИМЕЧАНИЕ

Если Apache запускается от лица непривилегированного пользователя, параметры User and Group (пользователь и группа) игнорируются и все дочерние процессы запускаются от лица пользователя, от лица которого запущен Apache. В подобной ситуации Apache не сможет связать порты с номерами ниже 1024.

Следующим параметром, представляющим интерес с точки зрения безопасности, является параметр DocumentRoot (корневой каталог web-документов). Если вы компилировали Apache, используя рассмотренную ранее конфигурацию, по умолчанию параметр DocumentRoot обладает значением /home/httpd/htdocs. Этот параметр определяет тюрьму с измененным корнем, в границах которой разрешается действовать клиентам Apache (если, конечно, не определено никаких псевдонимов). Как правило, именно с этого каталога начинается взаимодействие клиентов с сервером сразу же после того, как они подключились. В данном каталоге должен располагаться файл index.html, который по умолчанию самым первым передается клиентам, подключающимся к серверу. В состав Apache входит такой файл, содержащий в себе текст по умолчанию.

После определения параметра DocumentRoot следуют один или несколько разделов, начинающихся с выражения <Directory> и заканчивающихся выражением </Directory>. Каждый из этих разделов содержит в себе набор параметров, специфичных для соответствующего каталога и каталогов ниже данного. Первый раздел Directory в качестве аргумента использует каталог с именем «/». То есть содержащиеся в нем параметры относятся к определенному ранее каталогу DocumentRoot. Последующие разделы <Directory> начинаются с данного корневого каталога web-документов. Внутри разделов Directory содержатся параметры, которые действуют в отношении некоторого каталога, а также всех его подкаталогов. Эти значения могут быть перекрыты значениями, определяемыми в других разделах Directory. Раздел Directory, соответствующий самому высокоуровневому каталогу документов web-сервера, по умолчанию содержит два объявления: Options None и AllowOverride None. Эти объявления в значительной степени ограничивают клиентов, работающих с вашим web-сервером. Из соображений безопасности эти записи менять не следует. В листинге 20.4 приведены выражения и объявления, которые можно использовать в разделах Directory, а также параметры, которые допускается при этом использовать. Любые аргументы являются списком разделенных пробелами значений (за исключением объявления Order).

Листинг 20.4. Выражения и объявления, которые допускается использовать в рамках разделов <Directory> файла httpd.conf

```
Options
None, Indexes, Includes, IncludesNoExec, FollowSymLinks, SymLinksIfOwnerMatch, ExecCGI, MultiViews, All.
AllowOverride
        None, Filelnfo, AuthConfig, Limit, All
Order
        либо allow,deny, либо deny,allow
Allow
        All или конкретные узлы, домены, сети, IP-адреса
Deny
        то же, что и для Allow
<Limit>
        раздел, который должен замыкаться меткой </Limit> и должен полностью входить в раздел <Directory>
        Аргументами <Limit> являются: GET, POST, OPTIONS, PROPFIND, PUT, DELETE, PATCH, PROPPATCH,
MKCOL, COPY, MOVE. LOCK, UNLOCK и объявления: Order, Allow, Deny, как показано ранее.
```

- Аргумент None для любого объявления запрещает любые изменения для данного объявления.

- Аргумент АИ соответствует полному списку всех аргументов, допустимых для данного объявления. Однако если вы хотите использовать аргумент MultiViews, вы должны явно добавить его к аргументу All, так как аргумент All для объявления Options не подразумевает использования MuitiViews.

- Аргумент Indexes разрешает клиенту просмотр содержимого каталога, если отсутствует документ Directory-Index (как определено в файле httpd.conf). Чтобы ограничить просмотр одного или нескольких файлов, см. описание выражения <Files> далее.

- Аргумент Includes разрешает включать в состав файла другие файлы, даже если эти файлы расположены вне каталога DocumentRoot или каталога, являющегося псевдонимом. Таким образом можно

обеспечить выполнение файла, включенного в include. Эта возможность чрезвычайно опасна и должна использоваться с особой осторожностью.

- Аргумент IncludeNoExec разрешает включение файлов в состав других файлов, однако запрещает исполнять эти файлы. Это более безопасная альтернатива аргумента Includes.

- Аргумент FollowSymLinks разрешает клиентам перемещаться по символическим ссылкам, указывающим на ресурсы вне текущего дерева каталогов. Этот аргумент является таким же опасным, как и аргумент Includes.

- Аргумент SymLinksIfOwnerMatch — обеспечивает значительно более безопасный способ обработки символических ссылок. Разрешает следовать по ссылке только в случае, если пользователь является владельцем ресурса. Непривилегированные пользователи редко когда владеют чем-либо, расположенным за рамками непривилегированных каталогов, поэтому аргумент SymLinksIfOwnerMatch обеспечивает гибкость (в особенности внутри пользовательских каталогов) и при этом не снижает безопасность так, как это делает аргумент FollowSymLinks.

- Аргумент ExecCGI — это еще один весьма опасный аргумент, который разрешает исполнение файлов. Если эти файлы являются файлами SUID, они запускаются от лица владельца файла. Используя этот аргумент, следует быть осторожным, размещая файлы в каталоге.

- Аргумент MultiViews разрешает серверу обсудить с клиентом возвращаемое в ответ на запрос содержимое в случае, если запрашиваемый клиентом файл не существует. Этот аргумент на самом деле создает в системе безопасности огромную дыру, так как сервер и клиент получают возможность договориться о том, какой файл будет передан в ответ на запрос. Вы не обладаете никаким контролем над данным процессом, единственное, что вы можете сделать, это наложить ограничения на то, какие файлы содержатся в данном каталоге.

ВНИМАНИЕ

Будьте очень осторожны в отношении аргументов Includes, FollowSymLinks, ExecCGI и MultiViews. Везде, где это возможно, используйте более безопасные аргументы IncludesNoExec и SymLinksIfOwnerMatch. Избегайте использования MultiViews.

- Объявление AllowOverride должно использоваться с осторожностью, так как в результате его использования могут возникнуть проблемы с безопасностью.

- Аргумент Options позволяет использовать файл .htaccess для изменения специальных параметров, определенных для данного каталога. Если вы не хотите использовать некоторые из параметров, вы не должны указывать AUowOverride Options или ALL

- Аргумент Filelnfo разрешает вносить изменения в сведения о файловом типе, как объявлено (или нет) в файле http.conf. Этот аргумент также позволяет изменять значки, которые используются для обозначения определенных файловых типов.

- Аргумент AuthConfig позволяет использовать в отношении подкаталога механизмы управления доступом, благодаря чему перед тем, как разрешить пользователю вход в каталог, система будет просить его аутентифицировать себя. Об этом будет подробнее рассказано далее.

- Аргумент Limit позволяет ограничивать изменения ограничений, объявляемых внутри файла .htaccess.

ВНИМАНИЕ

Объявление AUowOverride может как расширить, так и сузить возможности контроля текущего каталога. Некоторые специальные аргументы, такие как All, Options или Limit, следует использовать с осторожностью.

- Имя директивы Order (порядок применения) говорит само за себя — эта директива определяет порядок применения политик.

- Директивы Allow и Deny, доступ к которым осуществляется с использованием директивы Order, могут использоваться для того, чтобы наложить ограничения на местоположение клиентов, имеющих право обращения к системе. Это можно сделать множеством способов. Однако следует иметь в виду, что при перенаправлении клиентов с использованием механизма proxy сервер Apache может подумать, что все клиенты обращаются к серверу из одного места — proxy (такое поведение зависит от используемого proxy). При некоторых условиях директивы Allow и Deny могут обрабатываться не так, как вы этого ожидаете. Чтобы обеспечить их правильную обработку, убедитесь в том, что программное обеспечение proxy передает соединения клиентов абсолютно прозрачно.

- Выражение <Limit> позволяет гибко настраивать систему доступа к каталогу таким образом, что в

рамках этого каталога некоторые действия будут разрешены, в то время как некоторые другие действия будут запрещены. Внутри угловых скобок выражения <Limit> указывается разделенный пробелами список аргументов. Специальные объявления Order, Allow from и/или Deny from размещаются между метками <Limit> и </Limit>. Совместно с <Limit> допускается использовать следующие аргументы (методы):

GET — загрузить файл (через HTTP или FTP);
POST — разместить информацию в форме;
PUT — передать на сервер файл через FTP;
DELETE — удалить файл;
CONNECT — любое соединение;
OPTIONS — наложить ограничения на допустимые параметры;
TRACE — включить трассировку;
PROPFIND — получить свойства документа;
PATCH — изменить файл;
PROPATCH — изменить свойства документа;
COPY — скопировать файл;
MOVE — переместить файл;
LOCK — блокировать доступ к файлу со стороны других процессов;
UNLOCK — снять блокировку с файла.

Предполагается, что выражения Directory, расположенные ниже выражения Directory, относящегося к DirectoryRoot, относятся к каталогам, которые являются частью корневого каталога документов (DirectoryRoot). Иными словами, подразумевается, что указанный в выражении Directory подкаталог входит в иерархию каталогов, начинающуюся с DirectoryRoot. Однако выражение Directory можно отнести также и к какому-либо каталогу, который не входит в DirectoryRoot. Для этого перед этим выражением Directory следует разместить одно из трех специальных объявлений. Эти объявления перечисляются далее. Каждое из них принимает два аргумента. Первый аргумент — это аргумент доступа к каталогу — некоторая последовательность символов, которую клиент указывает в качестве составной части URL для того, чтобы получить доступ к каталогу. Второй аргумент — это местоположение каталога в файловой системе. Именно туда Apache будет перенаправлять клиентов. Аргумент доступа к каталогу, как правило, указывается в форме /cgi-bin/, а местоположение каталога — это полный путь к этому каталогу в файловой системе Linux. Местоположение может указываться в форме /home/httpd/cgi-bin/. Специальные объявления это:

- Alias (псевдоним) — позволяет перенаправлять клиента к некоторому специальному месту в файловой системе;

- ScriptAlias — то же, что и Alias, однако предполагается, что каталог обладает разрешением ExecCGI (это разрешение должно быть явно указано в соответствующем выражении <Directory>);

- UserDir — домашний каталог пользователя, доступ к которому, как правило, осуществляется при помощи метки /~имя_пользователя/.

Еще одним полезным выражением, похожим на выражение <Directory>, является выражение <Files>. В выражении <Files> указывается разделенный пробелами список аргументов. По умолчанию в файле httpd.conf содержатся две записи: «~» и «\ .ht». Первая запись соответствует файлам, в именах которых содержится символ тильды (~). Эта запись, как правило, используется программами, которые создают резервные копии файлов. Вряд ли вы захотите, чтобы клиенты обладали доступом к этим устаревшим версиям файлов. Вторая запись является регулярным выражением (regular expression, regex). Оно соответствует всем файлам, в именах которых первые три символа являются символами .ht. Первый символ регулярного выражения — стрелочка вверх (^) — обозначает первый символ имени. Второй символ — обратная косая (\) — является esc-символом. Этот символ сообщает системе, что она не должна выполнять специальную интерпретацию символа, следующего за данным. Иными словами, символ, расположенный сразу же после символа обратной косой, не интерпретируется как спецсимвол, а рассматривается как обыкновенный символ. Сразу же за символом обратной косой стоит символ точки (.). В регулярных выражениях символ точки имеет специальное значение — им обозначают любой символ (подобно символу вопросительного знака ? при работе с командной оболочкой bash). Однако в данном случае имеется в виду именно символ точки, а не любой допустимый символ, поэтому чтобы отключить специальную интерпретацию этого символа, перед ним поставлен esc-сим-вол обратной косой черты. Определив таким образом в выражении <Files> некоторый набор файлов, вы можете разместить в разделе начиная с метки <Files> и заканчивая меткой </Files> директивы Order с объявлениями Allow и/или Deny, благодаря чему вы получаете возможность контролировать доступ к заданным файлам.

В файле httpd.conf используется также объявление AccessFilesName, которое располагается, как правило, до выражения <Files> и указывает имя файла .htaccess, в котором содержится информация об изменениях (если такие изменения разрешены) правила доступа в рамках данного каталога. Если вы меняете это имя файла, вы должны также изменить соответствующее выражение <Files>, в противном случае любой желающий сможет прочитать данный файл (и, если ему разрешена запись в каталог,

модифицировать или заменить данный файл).

Еще одним важным объявлением является директива DirectoryIndex, которая имеет большое значение для каталогов, содержимое которых не разрешается просматривать напрямую. В данном объявлении через пробел перечисляются допустимые имена так называемых индексных файлов. Если клиент обращается к серверу, указывая в URL только лишь имя каталога (а не конкретный файл), система произведет в указанном каталоге поиск файлов, на которые указывает директива DirectoryIndex. Поиск имен файлов будет производиться в том порядке, в котором они указаны в директиве DirectoryIndex. Первый подходящий обнаруженный системой файл будет передан в ответ на запрос пользователя. Если ни одного из указанных в директиве DirectoryIndex файлов обнаружить не удалось, сервер передаст клиенту список файлов в данном каталоге (если это разрешено). Если передача содержимого каталога клиенту запрещена, сервер вернет клиенту сообщение об ошибке. Если вы используете PHP3, выражение DirectoryIndex для вашего сервера может выглядеть следующим образом:

```
Directorylndex index.html  index.php3 index.phtml
```

Кроме того, возможно, вы захотите добавить и расширить ранее закомментированную строку:

```
AddType application/x-httpd-php3 .php3 .phtml
```

Эта строка разрешает использование файлов PHP3, обладающих расширениями либо php3, либо phtml.

Следующие три директивы разрешают использование на сервере CGI и/или HTML-документов, грамматический разбор которых выполняется на стороне сервера. Директива CGI разрешает запуск сценариев CGI из каталогов, которые не объявлены как ScriptAlias. Если каталог объявлен как ScriptAHas, для него не требуется использовать директиву CGI.

```
#AddHandler cgi-script .cgi
#AddType text/html .shtml
#AddHandler server-parsed .shtml
```

Комбинация двух последних директив объявляет тип файлов shtml и предписывает серверу выполнять грамматический анализ HTML-файлов на стороне сервера. Возможность грамматического анализа на стороне сервера добавляет вашему серверу гибкости, однако также увеличивает риск. Как директива CGI, так и директивы shtml снижают уровень защиты сервера и часто используются злоумышленниками для того, чтобы получить доступ к системе. Для некоторых web-узлов эти директивы являются необходимостью, однако если вы можете обойтись без CGI и грамматического разбора на стороне сервера, вы не должны использовать данные директивы.

Чтобы защитить вашу систему от непреднамеренных ошибок в конфигурации, добавьте в нее следующее выражение:

```
<Directory />
Order deny,allow
Deny from all
</Directory>
```

Это выражение запрещает кому-либо намеренный или непреднамеренный просмотр всей вашей файловой системы. Связанные с этим предупреждения содержатся в разделе *«Дополнительные замечания, связанные с безопасностью»* далее в данной главе.

Также будет лучше, если вы отключите доступ к пользовательскому каталогу учетной записи root. Для этого используется следующая простая директива:

```
UserDir disabled root
```

Записи SSL

Если вы компонуете Apache с использованием модуля mod_ssl, в конец файла httpd.conf добавляется раздел, который имеет отношение к SSL.

ВНИМАНИЕ

Прежде чем приступить к использованию вашего сервера SSL, вы обязаны выполнить настройку этого раздела.

Значения некоторых параметров следует оставить такими, какие они есть. Ранее у вас уже была возможность изменить номер порта, через который сервер будет ожидать поступления новых соединений. Данные, которые необходимо заменить в первую очередь, располагаются в разделе <VirtualHost>. Выражение VirtualHost содержит в себе метку _default_, которая указывает на то, что система будет

использовать следующие далее записи, если только для SSL-сервера, к которому клиент обращается по имени, не существует другого раздела VirtualHost, в котором указываются другие параметры. Вам также потребуется изменить номер порта для того, чтобы он соответствовал любым сделанным вами ранее изменениям.

Для данного узла набор параметров General будет выглядеть в точности так же, как и для обычной, не защищенной части HTTP-сервера. Это означает, что защищенный корневой каталог web-документов (DirectoryRoot) совпадает с незащищенным корневым каталогом web-документов.

ВНИМАНИЕ

Использование одного и того же корневого каталога Web-документов как для защищенной, так и для незащищенной части web-сервера — плохая идея. Директива DirectoryRoot защищенного сервера ни в коем случае не должна указывать на тот же самый каталог, что и директива DirectoryRoot незащищенного сервера. Эти два каталога не должны совпадать и не должны обладать общими подкаталогами. Два соответствующих дерева подкаталогов должны существовать полностью отдельно друг от друга, в противном случае создание защищенного web-сервера теряет смысл.

Разделить два дерева подкаталогов можно несколькими методами. Например, для незащищенной части web-сервера можно использовать каталог /home/httpd/ htdocs, а для защищенной части сервера создать параллельную структуру каталогов с другим именем (например secure). В частности, вы можете создать подкаталог secure в каталоге httpd и в качестве защищенного корневого каталога документов использовать каталог /home/httpd/secure. Другой вариант предусматривает создание двух раздельных подкаталогов в каталоге htdocs. Их можно назвать secure и nonsecure или public и private. Вам потребуется внести соответствующие изменения в конфигурацию обычного и защищенного web-серверов.

Для каждого из этих каталогов можно создать отдельный подкаталог cgi-bin. Это не является обязательным условием, однако если ваш незащищенный web-сервер использует множество сценариев CGI, возможно, вы не захотите, чтобы эти сценарии можно было запустить на стороне защищенного сервера.

Значение параметра ServerName может быть одним и тем же для обоих серверов, так как для обращения к защищенному серверу клиент будет использовать префикс https, а для обращения к незащищенной стороне будет использоваться обычный префикс http.

Возможно, будет более удобным, если вы будете использовать для защищенной и незащищенной стороны вашего сервера раздельные журналы ошибок и трансферов, однако, опять же, это не является обязательным и может быть реализовано на ваше усмотрение.

Многие параметры, о которых говорилось ранее, могут применяться также и в отношении защищенной стороны вашего сервера. Я надеюсь, что читатели смогут самостоятельно освоить использование этих параметров.

ВНИМАНИЕ

Если ваш защищенный и незащищенный web-серверы работают на одном и том же компьютере и один из этих серверов был взломан, можете считать, что второй сервер столь же уязвим. Будьте чрезвычайно осторожны в выборе сценариев CGI и параметров, действующих на незащищенной стороне вашего web-сервера.

В отношении Apache можно использовать множество других, более совершенных возможностей, однако я не буду описывать их в данной книге. К таким возможностям относятся директивы ограничения используемых шифров (таких как idea) и директивы определения уровня кодирования, который требуется от клиентов (например, запрещение 40-битных шифров), и т. п. Более подробно об этом можно узнать из электронной документации Apache.

Первые запуск и обращение к Apache

После того как вы внесли в конфигурацию сервера описанные ранее изменения и разместили в защищенном дереве каталогов по крайней мере один файл index, html, вы можете приступить к проверке механизмов доступа к web-содержимому. И если файл index.html защищенного сервера отличается от файла index.html незащищенного сервера, вы сможете со всей очевидностью обнаружить любые возникающие проблемы.

Для начала запустите Apache в качестве незащищенного сервера. Для этого можно использовать команду apachectl start. После этого запустите web-браузер и обратитесь по адресу http://localhost/, чтобы убедиться в том, что web-сервер функционирует. Если вы выбрали порт, номер которого отличается от номера по умолчанию 80 (например 8080), вы должны указать номер используемого сервером порта в составе URL (например, http://localhost:8080/). Если браузер отобразит корректную индексную HTML-

страницу, значит, половина вашего сервера работает нормально. Если система сработает некорректно, обратитесь к файлам журналов для того, чтобы понять, в чем проблема. Если все работает нормально, остановите сервер командой apachectl stop.

Теперь запустите web-сервер Apache с поддержкой SSL. Для этого необходимо использовать команду apachectl startssl. Если при создании серверного сертификата вы указали ключевую фразу, при запуске сервера Apache в режиме SSL система попросит вас ввести эту фразу. После того как сервер Apache начал работу, прежде всего обратитесь к незащищенной web-странице так, как это было описано в предыдущем абзаце, — это необходимо для того, чтобы убедиться, что незащищенная часть сервера, как и прежде, работает нормально. После этого в строке адреса вашего браузера введите URL: https://localhost/. Обратите внимание на присутствие буквы «s» в названии протокола https. Если вы используете порт, отличающийся от порта по умолчанию 443 (например, 8443), вы должны указать номер этого порта в составе URL (например, https://localhost:8443/). Если все работает так, как должно, на экране перед вами появится диалоговое окно New Site Certificate (сертификат нового узла). Это диалоговое окно появляется на экране потому, что ваш браузер не распознает авторитетный сертифицирующий орган, который является источником данного сертификата. Используя появляющиеся на экране диалоговые окна, сообщите системе всю необходимую информацию (например, срок использования сертификата) и примите сертификат. Если вы создали сертификат для некоторого конкретного имени сетевого узла, на экране появится сообщение о том, что указанный вами сетевой узел (localhost) не соответствует имени узла, которое содержится внутри сертификата. Это нормально. Если вы обратитесь к серверу, используя в точности то же имя, которым вы пользовались при создании серверного сертификата (о создании серверного сертификата было рассказано ранее в данной главе), предупреждение о несоответствии имен на экране не появится. Убедитесь в том, что сервер обращается к правильному корневому каталогу web-документов. Если все работает нормально, вы можете приступить к использованию вашего защищенного web-сервера.

Использование файлов .htaccess

Файл .htaccess может содержать множество разнообразных параметров и может использоваться для множества различных целей. Прежде всего, этот файл может использоваться для ограничения доступа к некоторому каталогу. Если вы хотите, чтобы доступом к каталогу обладали только лица, обладающие определенными именами и паролями, вы можете реализовать это с использованием файла .htaccess.

Прежде всего вы должны отредактировать файл /etc/httpd/conf/httpd.conf. Добавьте в этот файл выражение <Directory>, в котором содержится объявление Allow/Override Authconfig. Этим вы разрешаете использование в отношении данного каталога специального механизма авторизации. Не забудьте выполнить команду apachectl restart для того, чтобы приказать серверу заново прочитать содержимое конфигурационных файлов.

После этого создайте файл .htaccess. При создании необходимо добавить в файл .htaccess четыре строки, показанные в листинге 20.5. В первой строке содержится ключевое слово AuthName и имя. Имя может быть любым, однако оно должно быть по смыслу связано с той информацией, доступ к которой вы защищаете. Параметр AuthType всегда имеет значение Basic. Параметр AuthUserFile должен указывать на файл, который вы намерены использовать для аутентификации клиентов. Файл аутентификации клиентов создается с использованием htpasswd. В последней строке содержится директива require valid-user.

Листинг 20.5. Файл .htaccess

```
AuthName  "foo"
AuthType Basic
AuthUserFile   /путь/к/.htpasswd
require valid-user
```

Файл .htaccess может использоваться также для того, чтобы расширить или изменить любые из директив файла /etc/httpd/conf/httpd.conf для данного каталога. Для этого вы должны.внести в раздел <Directory> для данного каталога объявление AllowOverride Options. Формат файла .htaccess в точности совпадает с форматом файла httpd.conf, и фактически любые записи, которые используются в файле httpd.conf, могут быть использованы также в файле .htaccess.

Дополнительные замечания, связанные с безопасностью

Некоторые выражения, которые допускается использовать в файле httpd.conf, не обсуждались ранее, так как они не используются в конфигурационном файле по умолчанию. Эти выражения имеют отношение

к выражениям Directory и Files. К этим дополнительным выражениям относятся выражения DirectoryMatch, FilesMatch, Location и LocationMatch. Синтаксис этих выражений совпадает с синтаксисом выражений Directory и Files, однако добавка Match указывает на то, что в составе этих выражений используются аргументы, являющиеся регулярными выражениями (regular expression, regex). Аргументы, являющиеся регулярными выражениями, отличаются от обычных аргументов. Дополнительно об использовании этих аргументов рассказывается в электронной документации.

ПРИМЕЧАНИЕ

*Термин гедех обозначает regular expression — регулярное выражение. В Linux регулярные выражения используются для описания некоторого символьного шаблона. В рамках этого шаблона допускается использовать специальные символы (такие как * и ?) для того, чтобы указать неопределенный символ или набор символов. При помощи регулярных выражений вы можете специфицировать строки, начинающиеся с определенного набора символов, заканчивающиеся определенным набором символов, содержащие в себе определенный набор символов и т. п. Более подробно о регулярных выражениях и методах их использования можно узнать из книги, посвященной программированию sed, awk или Perl. Регулярные выражения — это чрезвычайно мощный и очень гибкий механизм, однако для новичков овладение этим механизмом может оказаться непростым делом.*

Необходимо отметить, что выражения Location и LocationMatch соответствуют не файлам и каталогам, а URL. Также необходимо отметить порядок, в котором обрабатываются все эти выражения. Любое выражение Directory обрабатывается в первую очередь. Если следующее за ним выражение Files содержит в себе конфигурационные значения, которые противоречат заданным в рамках выражения Directory, директивы Directory будут перекрыты директивами Files. Иными словами, выражение Files обладает более высоким приоритетом. В этом есть смысл, так как файл является более специфичным элементом файловой системы, чем весь каталог. Выражение Location перекрывает собой как выражения Directory, так и выражения Files. Такое распределение приоритетов менее очевидно, так как местоположение URL является виртуальной концепцией, в отличие от каталогов и файлов. Выражение Location обеспечивает большую гибкость, однако повышает сложность конфигурирования. Применив выражение Location, вы непреднамеренно можете перекрыть директивы, определенные с использованием Directory и Files. Используя выражения Location, будьте внимательны и осторожны.

Замечания, связанные с suEXEC

Программа suEXEC — это оболочка, которая позволяет запускать программы от лица пользователя, отличающегося от пользователя, от лица которого запущен web-сервер. При нормальном функционировании все программы запускаются на уровне привилегий пользователя, от лица которого запущен web-сервер. В случае использования конфигурации, описанной ранее в данной главе, три четверти программ работают на уровне привилегий пользователя nobody. В некоторых ситуациях это может быть неприемлемым. Например, если пользователи обращаются к индивидуальным базам данных, запись в которые со стороны посторонних пользователей недопустима. В этом случае вы можете компилировать Apache с использованием suEXEC и включить конфигурацию userdir. Эта возможность может быть использована в ситуациях, когда применение файлов .htaccess оказывается неудобным или неприемлемым.

Для настройки suEXEC необходимо некоторое время. Потребуются дополнительные конфигурационные директивы, среди которых:

```
--enable-suexec
--suexec-caller=UID
--suexec-docroot=DIR
--suexec-logfi1e=FILE
--suexec-userdir=DIR
--suexec-uidmin=UID
--suexec-gidmin=GID
--suexec-safepath=PATH
```

Данная возможность конфигурации обладает рядом недостатков. Однако используя ее, вы можете с большой гибкостью указывать (при помощи выражений VirtualHost) пользователя, от лица которого должна работать та или иная программа. К сожалению, компиляция с включением suEXEC и использование этого механизма в записях VirtualHost налагают ограничения на расположение каталогов VirtualHost — все они должны быть расположены в рамках иерархии основного корневого каталога документов DocumentRoot. Такое положение вещей не всегда можно считать приемлемым.

Наибольшую опасность при использовании suEXEC представляет указание небезопасного пути (PATH) для каталогов, содержащих в себе бинарные файлы. Каждый из бинарных файлов должен запускаться от лица указанного вами пользователя — это выглядит замечательно с точки зрения системы, однако может оказаться бедствием для пользователя, в особенности если бинарный файл является «троянским конем».

Ядро Linux 2.4.x и khttpd

В состав ядра Linux 2.4.x (которое на момент написания данной книги до сих пор находится в стадии разработки) входит работающий на уровне ядра демон HTTP (khttpd). Этот демон не является полноценной заменой web-сервера Apache, однако он служит для улучшения его работы. Разработчики демона исходили из предположения, что большинство web-страниц являются статическими, поэтому доступ к ним можно обеспечить с использованием небольшого, быстрого, безопасного демона, работающего на уровне ядра. Демон khttpd ожидает поступления запросов через порт 80 (привилегированный порт). Запросы, связанные с обработкой простых статических web-страниц, обрабатываются демоном khttpd. Если демон khttpd не в состоянии обработать некоторый запрашиваемый клиентом документ (например, документ, содержащий код PHP3, или документ, предусматривающий сложный грамматический анализ), такой запрос переадресуется другому демону HTTP (например Apache), который ожидает поступления запросов через порт 8080. После этого более мощный и совершенный web-сервер (такой как Apache) выполняет обработку подобных запросов. Перенаправление запросов осуществляется от демона khhtpd из порта 80 узлу localhost на порт 8080 (или любой другой указанный вами порт).

Демон khttpd можно скомпоновать либо в виде модуля, либо в виде демона, встроенного в ядро. В файле /etc/re.d/rc.local можно указать, запускается ли он в процессе начального запуска ОС или его запуск осуществляется позднее. Если этот демон скомпилирован в виде модуля, каталог khttpd не появится в рамках дерева /proc до тех пор, пока данный модуль не будет загружен. Как только модуль khttpd загружается в память, в файловой системе появляется каталог /proc/sys/net/ khttpd. В этом каталоге присутствуют следующие файлы:

```
clientport
documentroot
dynamic
logging
maxconnect
perm_forbid
perm_required
serverport
sloppymime
start
stop
threads
unload
```

Каждый из этих файлов может быть настроен в соответствии с вашей конкретной конфигурацией. В первом файле содержится клиентский порт (clientport) -это порт, в который демон khttpd передает запрос (как клиент) в случае, если обработка этого запроса выходит за рамки операции простого копирования файла в сеть. По умолчанию используется клиентский порт 80, через который web-сервер Apache ожидает поступления запросов. Это означает, что по умолчанию демон khttpd настроен как вспомогательное средство, а не как основной демон обработки запросов HTTP. Чтобы сделать демон khttpd основным web-сервером, а Apache — вспомогательным, измените клиентский порт на 8080 (или любой другой неиспользуемый порт) и прикажите серверу Apache ожидать поступления запросов через этот порт. Также внесите рекомендуемые изменения в файл serverport (см. далее).

Второй файл содержит в себе местоположение каталога DocumentRoot. По умолчанию корневым каталогом web-документов для демона khttpd является каталог /var/www. Есго следует изменить при помощи команды echo "/home/httpd/htdocs" > /proc/sys/net/httpd/documentroot.

Предназначение третьего файла менее понятно, чем первых двух. Файл dynamic содержит динамические строки, поиск которых будет осуществлять демон khttpd. По умолчанию в этом файле содержится следующий текст: Dynamic strings are : -cgi-bin- -..---. Возможно, если вы используете PHP, вам потребуется добавить в этот файл php3 (echo php3 > /proc/sys/net/khttpd/dynamic).

Четвертый файл, logging, указывает, должно ли осуществляться документирование сведений в журналах с использованием syslog. В этом файле может содержаться либо 0, либо 1. По умолчанию в файле содержится 0. Пятый параметр ограничивает максимальное количество одновременных подключений. По умолчанию в этом файле содержится 1000 (этого должно быть достаточно для любых, даже самых крупных

узлов, однако для узлов с недостаточной емкостью каналов связи значение этого параметра можно снизить до 100 или меньше).

По умолчанию в файле perm_forbid содержится весьма разумное значение 16969. Это число является маской, определяющей, какими характеристиками должен обладать файл. Характеристики могут быть следующими (читаем числа в направлении слева направо): FIFO (именованный канал — named pipe, на что указывает 1 в первом разряде), SUID или SGID (4 или 2 соответственно, а комбинация равна значению 6), чтение, запись, исполнение (7) или только запись (2) для владельца — комбинированное значение равно 9, разрешение на чтение (4) и запись (2) для группы — комбинированное значение равно 6. Если вы меняете это значение, убедитесь в том, что вы хорошо понимаете, как оно обрабатывается. Еще одним оправданным и разумным значением является значение 16999. Если вы внесете в этот файл какое-либо значение, вы можете нарушить защиту вашей системы.

Файл perm_required обрабатывается приблизительно так же, как и файл perm_ forbid, однако в данном случае для того, чтобы быть предоставленным пользователю, файл должен быть доступен только для чтения всем пользователям. Если файл не открыт для чтения всем пользователям, он не будет обрабатываться.

В файле serverport содержится номер порта, через который демон khttpd ожидает поступления клиентских запросов. По умолчанию демон khttpd настроен в качестве вспомогательного сервера, поэтому в файле serverport содержится значение 8080. Если вы намерены сделать khttpd основным web-сервером, а Apache — вспомогательным, вы должны внести в этот файл необходимые изменения, кроме того, вы должны соответствующим образом модифицировать файл httpd.conf сервера Apache. Если вы используете Apache в качестве вспомогательного web-сервера (рекомендуется), измените в конфигурации Apache объявление BindAddress с * на 127.0.0.1. В результате сервер Apache будет принимать запросы только от демона khttpd.

Файл sloppymime может содержать одно из двух значений: 1 или 0. Если этот файл содержит значение 1, любой неизвестный тип MIME будет рассматриваться как тип text/html и будет обрабатываться демоном khttpd. Если файл sloppymime содержит значение 0, любой неизвестный тип MIME будет передан web-серверу пользовательского режима (Apache).

Следующие два файла — start и stop — по умолчанию содержат в себе значение 0. Если в файл start заносится 1, файл stop автоматически устанавливается равным О, при этом демон khttpd приступает к обслуживанию поступающих запросов. Если в файл stop вносится значение 1, файл start автоматически становится равным О и демон khttpd прекращает обработку запросов.

Файл threads по умолчанию содержит 2. Это значение определяет количество серверных программных потоков, которые могут обрабатываться одним центральным процессором. Как правило, это значение должно быть равно 1. Вы можете увеличить его только в случае, если ваш узел является очень крупным Web-узлом (настолько крупным, что все активные файлы не вмещаются в оперативную память).

Последний файл unload подготавливает khttpd к выгрузке из памяти. Прежде чем выгрузить модуль из памяти, необходимо записать значение 1 в файл stop (при этом в файл start автоматически будет записано значение 0, что блокирует запуск дополнительных программных потоков khttpd). Программные потоки, которые в данный момент уже функционируют, продолжат свою работу, и вы должны либо дождаться, пока они завершат функционирование, либо послать им сигнал SIGHUP (kill -HUP 'идентификатор PID демона khttpd'). После этого вы можете подготовить модуль khttpd к выгрузке, для чего необходимо занести 1 в файл unload. Теперь вы можете без опасений отдать команду rmmod khttpd. При загрузке модуля в память в файл unload автоматически будет внесено значение 0.

Заключение

В данной главе вы узнали о том, как осуществляется конфигурирование и компоновка сервера Apache-1.3.9 с поддержкой SSL и PHP3. Вы также узнали о том, как выполняется настройка Apache. Я рассказал вам о некоторых рискованных с точки зрения безопасности конфигурациях. Напомню некоторые опасные варианты конфигурации:

- защищенный и незащищенный каталоги DocumentRoot совпадают;

- там, где необходимо использовать вложение файлов, вместо IndudesNOEXEC используется просто Includes;

- там, где необходимо использовать символические ссылки, вместо SymLinksIfOwnerMatch используется FollowSymLinks;

- пренебрежение использованием безопасных CGI-каталогов, определяемых при помощи объявления ScriptAlias;

- использование выражений <Location>, которые перекрывают собой конфигурацию, заданную при

помощи выражений <Directory> и <File>;

- использование небезопасной директивы ExecCGI;

- использование директивы Alias, которая делает доступным для внешних клиентов корневой каталог вашей системы;

- отказ от использования директивы AllowOverride AuthConfig и файлов .htaccess для обеспечения более надежной защиты важных областей вашего web-сервера;

- отказ от отключения домашнего каталога пользователя root;

- отказ от использования выражения Directory, запрещающего доступ к каталогу «/».

Я также рассказал о демоне уровня ядра khttpd, который в состоянии обслуживать простые запросы HTTP и который будет включен в состав ядра Linux 2.4.x.

21 Использование оболочки Secure Shell и сетей VPN

В данной главе рассматриваются следующие вопросы:
- что такое Secure Shell (SSH);
- компоновка и установка SSH;
- конфигурирование SSH;
- использование SSH;
- что такое FreeS/WAN;
- компоновка и установка FreeS/WAN;
- конфигурирование FreeS/WAN;
- расширение сети WAN;
- что такое OpenSSH.

Одной из наиболее сложных с точки зрения безопасности проблем, с которыми сталкиваются в наше время пользователи и сетевые администраторы, является обеспечение безопасного способа удаленного доступа к расположенным дома или в офисе системам. Сам по себе удаленный доступ к таким системам не является проблемой. В состав каждой системы Linux входят клиент и сервер telnet. Однако использование telnet означает, что имя пользователя и пароль передаются через канал связи в незашифрованном виде. Это обстоятельство не является проблемой в случае, если для передачи данных используется выделенная телефонная линия. Однако если данные передаются далее, за сервер, который обеспечивает соединение между телефонной линией и Интернетом, любой пользователь tcpdump или какой-либо другой утилиты, прослушивающей сеть, сможет перехватить пользовательское имя и пароль.

Чтобы иметь возможность в любой точке страны получать электронную почту, многие путешественники пользуются услугами общенациональных интернет-провайдеров, которые во многих городах обеспечивают защищенный доступ к почтовым ящикам своих клиентов с использованием локальной телефонной сети. Если же использовать подобное соединение для доступа к домашней или офисной сети, данные могут передаваться через множество различных сетей, и каждый, кто обладает доступом к этим сетям, сможет перехватить передаваемые через сеть пакеты. Пакет содержит имя пользователя и пароль, а в заголовке пакета указывается адрес назначения. Обладая подобными данными, даже ребенок, мечтающий стать хакером, сможет проникнуть в чужую систему. Именно по этой причине в системе OpenLinux по умолчанию пользователь root не имеет права удаленного подключения к системе. Недостаток данного подхода заключается в том, что пользователь может подключиться к системе, используя непривилегированную учетную запись, а затем при помощи su получить привилегии root — фактически это то же самое, что и удаленное подключение с использованием учетной записи root

Однако если подключение к системе осуществляется с использованием шифруемого соединения, получение пользовательского имени и пароля из перехваченных пакетов становится фактически невозможным. При этом для того, чтобы «вставить» себя между вами и вашим сервером и реализовать так называемую атаку man-in-the-middle (человек в середине), злоумышленник должен будет приложить немалые весьма нетривиальные усилия. Атака подобного рода является достаточно сложной, ее сможет реализовать только хорошо подготовленный, обладающий немалыми знаниями взломщик.

В рабочей среде Linux существует множество инструментов и программ, которые позволяют организовать безопасный обмен данными между системами. Мы с вами уже рассмотрели один из подобных механизмов в прошлой главе, где речь шла о web-сервере Apache, обладающем поддержкой технологии SSL. Конечно, подобное решение может оказаться несколько ограничивающим, однако пользователи могут запускать защищенные с использованием SSL web-серверы, связанные с непривилегированными портами с использованием файлов .htaccess для того, чтобы просматривать и загружать (через HTTP или FTP) документы, расположенные в их домашних каталогах.

Однако чтобы наделить пользователей более широкими возможностями, а именно позволить им работать с системой так, как они работают с ней при помощи telnet (то есть работать с системой через сеть фактически так же, как будто они работают с локальной консолью), вам потребуется некий эквивалент telnet. В последующих разделах я расскажу о двух альтернативных способах организации такого взаимодействия с учетом требований безопасности. Первый способ основан на использовании программы

под названием Secure Shell (SSH), а второй предусматривает создание полноценной виртуальной частной сети (Virtual Private Network, VPN) с использованием программного средства FreeS/WAN.

Как уже отмечалось в предыдущей главе, на прилагаемом к данной книге компакт-диске вы не найдете ни одного из этих программных продуктов. Это сделано для того, чтобы избежать нарушения действующих в США ограничений на экспорт кодирующих технологий. Следует отметить, что любой системный администратор, который работает на уровне привилегий root через незащищенную сеть и не использует эти программные средства, периодически будет иметь дело с несанкционированным проникновением в подконтрольные ему системы. Чтобы избежать этого, необходимо в обязательном порядке организовать должную защиту передаваемых через сеть данных с использованием одного из двух упомянутых средств. Программы можно получить по следующим адресам: ftp://ftp.cs.hut.fi/ pub/ssh/ssh-1.2.27.tar.gz и ftp://ftp.xs4all.nI/pub/crypto/freeswan/freeswan-l.l.tar.gz.

Secure Shell

Программа Secure Shell (название переводится как *«защищенная командная оболочка»)* предназначена для организации защищенного удаленного доступа к системе через сеть в режиме, напоминающем рабочий сеанс telnet. В настоящее время существует две разновидности Secure Shell (в дальнейшем SSH): версия 1 и версия 2. Каждая из разновидностей поддерживается ее автором. Различие между этими разновидностями состоит в наборе возможностей (версия 2 обладает некоторыми дополнительными весьма полезными возможностями) и в лицензировании. В общем и целом версия 1 распространяется абсолютно бесплатно для некоммерческого использования. Однако если вы намерены использовать версию 2, вы в любом случае должны приобрести лицензию. В данном тексте я буду рассказывать о версии 1, а конкретнее, о пакете ssh-1.2.27.

Компоновка и установка SSH

Загрузив SSH, определите, где вы намерены установить эту программу (местоположение файла не имеет значения, вы можете разместить его например в каталоге $HOME). После этого раскройте архив при помощи команды:

```
tar xzvf ssh-1.2.27.tar.gz
```

При этом будет создан каталог ssh-1.2.27, в который вы должны перейти, чтобы продолжить.

Для тех, кто плохо владеет процедурами компоновки программных пакетов, отмечу, что пакет SSH использует новые файлы GNU autoconf. Эти файлы являются упрощенной альтернативой прямого редактирования файлов Makefile и обеспечивают простой метод формирования разнообразных конфигураций, а также проверки корректности настройки программного пакета. Механизм GNU autoconf еще недостаточно хорошо отлажен, поэтому при установке некоторых пакетов возникают проблемы, связанные с тем, что этот механизм не может обнаружить в системе установленного в ней программного обеспечения. Однако в среде OpenLinux файл ssh-1.2.27 autoconf работает без сбоев.

Для начала отдайте команду ./configure --help. По этой команде на экран будет выведен достаточно длинный перечень параметров, позволяющих скомпоновать пакет с использованием самых разнообразных конфигураций. В данном тексте будут затронуты лишь некоторые из них. Как правило, значения параметров по умолчанию являются вполне подходящими, однако конкретно в вашей ситуации вы можете изменить некоторые из них. В верхней части экрана показаны префиксы каталогов для установки. По умолчанию установка осуществляется в каталоге /usr/local, и в большинстве случаев это вполне приемлемо, если только вы не используете для этой цели каталог /opt. Далее перечисляются типы сетевых узлов. В случае если вы компонуете программу в системе, в которой она будет установлена, типы узлов вам не понадобятся. В последнем разделе перечисляются разнообразные возможности и пакеты. Этот список следует изучить внимательнее. К наиболее важным параметрам относятся:

- --with-x — если вы хотите добавить библиотеки для разработки X11 (неплохая идея);

- --without-idea — если вы находитесь в Европе и не можете использовать алгоритмы кодирования IDEA;

- --without-rsh — если вы хотите запретить SSH переходить в режим взаимодействия незащищенной оболочки rsh в случае, когда SSH не может обнаружить сервера ssh (еще одна неплохая идея);

- --with-secured=/путь/к/secureid — если ваша система использует карту Security Dynamics SecurID;

- --with-kerberos5 — если вы используете Kerberos 5 (Kerberos 4 не поддерживается);

- --with-libwrap — если вы желаете использовать оболочку TCP (то есть TCP Wrappers и файл /etc/hosts.allow);

- --with-socks --with-socks4 --with-socks5 — если вы хотите обеспечить поддержку брандмауэра SOCKS;

- --with-rsaref — если вы желаете/должны удовлетворить требованиям патента RSA (США).

При желании вы можете выбрать и другие параметры. Выберите тот набор параметров, которые вы намерены включить или отключить. Я рекомендую вам создать для этой цели исполняемый сценарий, подобный тому, текст которого приведен в листинге 21.1.

Листинг 21.1. Возможная конфигурация SSH

```
./configure  --with-x \
             --without-rsh \
             --with-rsaref
```

Если вы желаете или обязаны использовать RSAref, вы должны следовать инструкциям, описанным в данном абзаце. Электронная справка о конфигурации подсказывает вам, что вы можете сообщить SSH о том, где располагаются библиотеки RSAref, однако этот раздел сценария некорректен. В каталоге, в котором вы выполняли команду .configure -help, создайте подкаталог rsaref2 (mkdir rsaref2). Это имя изменять нельзя — оно должно быть именно таким, как показано здесь. После этого перейдите в подкаталог rsaref2 и выполните команду tar xzvf/путь/к/ rsaref20.tar.Z. После этого перейдите обратно в каталог конфигурации и запустите ваш сценарий конфигурирования.

После выполнения конфигурационного файла скорректируйте любые неточности (это, как правило, отсутствие библиотек и неправильные пути к файлам). После того как конфигурационный сценарий будет выполнен без ошибок от начала до самого конца, вы можете выполнить команду make. Выполнение этой команды может занять некоторое время. Когда ее функционирование завершится, перейдите на уровень привилегий root (su) и выполните команду make install. По этой команде все будет установлено и подготовлено к использованию, включая создание закрытых системных ключей.

После того как программа SSH будет установлена в системе, необходимо убедиться в том, что бинарный файл sshd запускается в процессе начального запуска системы. Проще всего добиться этого при помощи следующей команды:

```
echo "/usr/local/sbin/sshd" > /etc/rc.d/rc.local
```

По умолчанию серверный демон Secure Shell будет запускаться с использованием ключа размером 768 бит (стандартный изначальный размер для RSA). Если вы хотите использовать ключ большего размера, добавьте в командную строку sshd параметр -Ь 1024. Число 1024 можно заменить любым удобным для вас количеством битов в ключе. Имейте в виду, что применение ключа большего размера требует большей вычислительной мощности, при этом может замедлиться скорость передачи данных через канал, однако уровень защиты существенно возрастает. Использовать ключи с размером меньшим, чем 768, не рекомендуется, так как на существующем уровне технологий злоумышленники могут легко дешифровать ключи длиной в 512 бит, и даже ключ размером 768 нельзя расценивать как неуязвимую защиту на все сто процентов. При каждом запуске демона sshd требуется некоторое заметное время для генерации ключа, поэтому запуск с использованием inetd не рекомендуется. Чтобы запустить демон sshd без перезагрузки, просто наберите команду запуска sshd в командной строке.

ПРИМЕЧАНИЕ

Добавление соответствующей записи в файл /etc/services не обязательно, однако будет неплохо, если вы зарегистрируете порт с номером 22 как порт, связанный с ssh. Для этого в файл /etc/services необходимо добавить следующие строки:

```
ssh             22/tcp
ssh             22/udp
```

В процессе установки SSH в каталоге /etc создаются следующие файлы: - sshd_config — конфигурация по умолчанию для серверного демона SSH; - ssh_config — конфигурационный файл по умолчанию для SSH;

- ssh_host_key — закрытый ключ для данного узла; правом доступа к нему обладает только пользователь root, для всех остальных пользователей доступ должен быть закрыт;

- ssh_host_key.pub — открытый ключ для данного узла;

- ssh_random_seed — создается каждый раз при запуске sshd (правом доступа должен обладать только пользователь root и никто другой);

В файл ssh_host_key в процессе установки записывается 768-битный ключ. Если вы хотите увеличить размер этого ключа до, скажем, 1024 бит, вы должны выполнить команду ssh-keygen -b 1024. После генерации ключа запишите его в файл /etc/ssh_host_key. В процессе генерации ключа будут созданы файлы ssh_host_key и ssh_ host_key.pub. Когда система предложит вам ввести ключевую фразу, *не вводите ее.* Если вы укажете ключевую фразу, каждый раз при запуске sshd вам придется вводить ее заново, а это очень неудобно, так как в большинстве случаев запуск sshd происходит в процессе начальной загрузки системы наряду с другими программами.

После того как вы выполните все вышеописанные шаги на каждой из систем, на которых вы хотите установить SSH, и запустите демон sshd, вы можете собрать все файлы ssh_host_key.pub и скопировать их содержимое в один большой файл под именем /etc/ssh_known_hosts с режимом доступа chmod 644, который следует записать на каждый из сетевых узлов. Благодаря этому вы сможете избежать подделки IP-адресов и обеспечить больший уровень безопасности. На этом все процедуры, выполняемые на уровне привилегий root, завершены.

ВНИМАНИЕ

Если злоумышленник получил возможность воспользоваться учетной записью root на одной из ваших систем, вы больше не можете доверять ключам SSH. Все ключи следует заменить новыми, в противном случае вы не можете быть уверенными в том, что системы корректно идентифицируют себя.

Использование SSH

Теперь вы сможете, как обычный пользователь, работать с любым удаленным узлом, на котором установлена программа SSH и на котором у вас есть своя учетная запись. Вы подключаетесь к системе примерно так же, как это происходит при использовании telnet, однако оболочка SSH передает на удаленную систему ваше пользовательское имя. Система запрашивает у вас пароль, который соответствует этому пользователю. Если пользовательское имя, которое вы хотите использовать на удаленной системе, отличается от вашего локального пользовательского имени, вы можете использовать ключ -l для того, чтобы указать другое пользовательское имя. В отличие от telnet, программа SSH не накладывает никаких ограничений на то, можете ли вы использовать для подключения учетную запись root. Это происходит потому, что как только устанавливается соединение TCP, первое, что делает SSH, это обмен открытыми ключами и создание защищенного шифруемого туннеля. После этого все, что передается через этот туннель, включая имя подключающегося пользователя, шифруется.

При первом запуске SSH эта программа создает каталог с именем .ssh, в котором размещается файл random_seed. Если вы ранее не выполнили необязательную процедуру создания файла /etc/ssh_known_hosts или подключились к системе, которая не упоминается в файле /etc/ssh_known_hosts, то в каталоге .ssh также будет расположен файл known_hosts (в этом случае вначале система сообщит вам, что соответствующей записи в файле known_hosts не существует, на вопрос, надо ли создавать такую запись, следует ответить утвердительно). Программа SSH автоматически копирует файл ssh_host_key.pub в файл known_hosts, в дальнейшем каждый раз при создании соединения будет осуществляться проверка. Если ключ в файле known_hosts не соответствует ключу, переданному удаленным узлом, программа предупредит вас о том, что полученный ключ не существует и что кто-то пытается реализовать атаку типа man-in-the-middle (вклинивание между системами). Программа SSH чрезвычайно подозрительна на этот счет, однако в этом случае она спросит у вас, хотите ли вы все же продолжить создание соединения. Если вы уверены в том, что ключ на удаленной системе действительно изменился, вы можете избежать вывода предупреждения. Для этого необходимо удалить соответствующий ключ из файла known_hosts. В этом случае при следующей попытке подключения программа SSH сообщит вам, что в файле known_hosts отсутствует соответствующая удаленному узлу запись. При этом вам будет предложено продолжить создание соединения. Если вы ответите утвердительно, программа скопирует новый ключ с удаленной системы в файл known_hosts.

Чтобы упростить работу с SSH, любой постоянный пользователь этого механизма связи может создать свой собственный идентификационный файл. Предположим, что пользователь регулярно подключается к нескольким разным удаленным системам с использованием разных пользовательских имен и паролей. Конечно же, для подключения к удаленным системам с использованием различных пользовательских имен можно использовать аргумент командной строки -l, однако если пользователь имеет дело с множеством различных систем, ему будет сложно запомнить множество сложных и отличающихся друг от друга паролей, из-за этого процедура подключения может усложниться. В этом случае, чтобы упростить дело, пользователь может создать свой собственный идентификационный файл. Для этого необходимо воспользоваться командой ssh-keygen (с необязательным указанием ключа -Ь), по которой в подкаталоге $HOME/.ssh будут созданы файлы идентификации identity и identity.pub. При этом система предложит пользователю ввести ключевую фразу. Содержимое сгенерированного таким образом файла identity.pub необходимо скопировать в файл $HOME/.ssh/authorized_keys на удаленной системе (в случае необходимости этот файл следует создать). Теперь, когда пользователь подключается к удаленной системе, оболочка SSH будет запрашивать его ввести не пароль пользователя удаленной системы, а ключевую фразу для файла идентификации. Таким образом, для подключения ко всем разнообразным системам, на которых файл authorized_keys содержит в себе данные из файла identity.pub для некоторого пользователя, этому пользователю требуется запомнить всего лишь один пароль.

При создании файлов идентификации пользователь вовсе не обязан указывать ключевую фразу. Он может либо указать ее, либо оставить соответствующее поле пустым. В этом отношении необходимо

учесть связанные с этим соображения безопасности. Право чтения-записи файла $HOME/.ssh/identity принадлежит только владельцу этого файла, то есть пользователю, которого идентифицирует этот файл. Если учетная запись этого пользователя становится доступной для злоумышленника и если при этом пользователь не использует ключевую фразу, злоумышленник автоматически получает возможность доступа ко всем удаленным системам, на которых содержится идентификационная информация для данного пользователя. В этом случае чтобы восстановить защиту, пользователь будет вынужден генерировать новые идентификационные файлы и внести соответствующие изменения во все удаленные системы, с которыми он взаимодействует. При отсутствии ключевой фразы пользователь получает возможность организовать удаленное подключение без собственного участия. Иными словами, он сможет запускать от своего имени автоматические сценарии взаимодействия с удаленными системами, и эти сценарии смогут подключаться к удаленным системам без участия пользователя (так как при отсутствии ключевой фразы никакого пароля при удаленном подключении не требуется). Это удобно, однако при этом повышается вероятность того, что удаленные системы будут взломаны с использованием локальной учетной записи этого пользователя.

ПРИМЕЧАНИЕ

Если вы хотите упростить процесс подключения пользователя к множественным удаленным системам, вы можете воспользоваться для этой цели специальным графическим программным средством под названием sshbuddy. Это программное средство запоминает имя удаленного сервера и соответствующее ему имя пользователя. Таким образом, доступ к удаленной системе может быть осуществлен при помощи всего двух щелчков мыши.

Конфигурация SSH и SSHD

В состав программного пакета SSH входят две конфигурационные программы: одна для клиента (ssh_config) и одна для сервера (sshd_config). Когда любая из этих программ начинает работу, она прежде всего анализирует конфигурационные параметры из командной строки, затем — из индивидуальных пользовательских файлов .ssh, а затем из системных конфигурационных файлов. Если на одном из этих этапов настраивается значение некоторого параметра, все последующие настройки значения этого параметра игнорируются.

Файл /etc/ssh_config, содержимое которого показано в листинге 21.2, содержит значения параметров по умолчанию. Эти значения используются системой в случае, если ни один из параметров нигде не переопределяется.

Листинг 21.2. Фрагмент файла /etc/ssh_config, демонстрирующий значения по умолчанию

```
# Host *
  ForwardAgent yes
  ForwardX11 yes
  RhostsAuthentication yes
  RhostsRSAAuthentication yes
  RSAAuthentication yes
  TISAuthentication no
  PasswordAuthentication yes
  FallBackToRsh yes
  UseRsh no
  BatchMode no
  StrictHostKeyChecking no
  IdentityFile ~/.ssh/identity
  Port 22
  Cipher idea
  EscapeChar ~
```

Не обращайте внимание на то, что все строки этого листинга являются комментариями. Именно эти значения будут использоваться программой SSH в случае, если конфигурация не модифицируется при помощи командной строки, пользовательских конфигурационных файлов или других записей в системных конфигурационных файлах. Конфигурацию можно модифицировать для каждого узла по отдельности. Если вы хотите определить значения конфигурационных параметров для некоторого конкретного узла, вы должны внести соответствующие записи в раздел Host перед разделом Host, относящимся ко всем остальным узлам (под заголовком all other).

О некоторых параметрах следует рассказать особо. Параметр ForwardXll разрешает автоматическое и прозрачное перенаправление дисплейных сообщений X11 обратно на локальный узел. Подобного эффекта можно добиться, выполнив на локальном узле команду xhost +remotehost, а затем на удаленном узле присвоив переменной окружения DISPLAY значение localhost (export DISPLAY=CLIENTHOST:0.0). Работа в подобном режиме зависит от того, найдет ли программа SSH на удаленном узле приложение xauth (для некоторых систем, не являющихся системами Linux, это может оказаться проблематичным). Если запрашивается перенаправление X11, однако программа SSH не разрешает этого, на экране появится

соответствующее сообщение. Если перенаправление X11 разрешено, соединения X11 будут осуществляться через защищенный шифруемый туннель. Таким образом, если вы воспользуетесь командой ssh foo, а затем на узле foo запустите xterm, информация, отображаемая xterm, будет отображаться на вашей локальной системе (предполагается, что вы подключились к рабочему сеансу X). При этом запускать оболочку X на узле foo вовсе не обязательно, достаточно чтобы для вас были доступны клиентские программы. Следует иметь в виду, что через медленное соединение (например, аналоговый модем) работа в подобном режиме может оказаться нежелательной.

Обратите внимание, что по умолчанию программа SSH использует шифр IDEA. Если вместо этого вы желаете использовать тройной шифр DES, раскомментируйте строку Cipher и замените значение idea на значение 3des, или запустите ssh с аргументом -c 3des..

Конфигурационный файл по умолчанию для серверного демона SSH показан в листинге 21.3.

Листинг 21.3. Файл конфигурации по умолчанию /etc/sshd_config

```
Port 22
ListenAddress 0.0.0.0
HostKey /etc/ssh_host_key
RandomSeed /etc/ssh_random_seed
ServerKeyBits 768
LoginGraceTime 600
KeyRegenerationlnterval 3600
PermitRootLogin yes
IgnoreRhosts no
StrictModes yes
QuietMode no
X11Forwarding yes
X11Displa/Offset 10
FascistLogging no
PrintMotd yes
KeepAlive yes
SyslogFacility DAEMON
RhostsAuthentication no
RhostsRSAAuthentication yes
RSAAuthentication yes
PasswordAuthentication yes
PermitEmptyPasswords yes
UseLogin no
# CheckMail no
#PidFile /u/zappa/.ssh/pid
#AllowHosts *.our.com friend.other.com
#DenyHosts lowsecurity.theirs.com *.evil.org evil.org
#Umask 022
#SilentDeny yes
```

Прежде всего обратите внимание на то, что этот конфигурационный файл не содержит в себе закомментированных строк (за исключением нескольких строк в конце). Это потому, что все эти значения нужны серверу для нормального функционирования. В отличие от клиентской части, сервер SSH не содержит в себе встроенных в код конфигурационных значений по умолчанию — вся конфигурация в обязательном порядке читается из файла. Многие из этих значений легко можно изменить. Некоторые из них влияют на то, каким способом пользователи взаимодействуют с системой и что они принимают от сервера. Эти значения можно изменить в соответствии с политиками локальной системы. Эффекты модификации различных параметров могут быть самыми разными. Я не буду подробно рассказывать об этом, так как лишь немногие из этих параметров напрямую связаны с безопасностью. Исключением являются параметры, имеющие отношение к RHosts. Если вы не хотите позволять пользователям подключаться к системе с использованием файлов .rhosts в стиле rsh, отключите все эти параметры. В большинстве ситуаций использование файлов rhosts — это плохая идея. Подключение, основанное на файлах identity и authorized_keys, обеспечивает более высокий уровень безопасности.

FreeS/WAN

Принцип действия программного механизма FreeS/WAN несколько отличается от пакета SSH. Пакет FreeS/WAN предназначен для формирования защищенного шифруемого туннеля между двумя закрытыми частными сетями, при этом шифруемый туннель пролегает через открытые, публично доступные каналы связи (как правило, Интернет). Набор операций, выполняемых с использованием SSH, ограничен действиями, которые вы можете выполнить при помощи telnet (это не относится к версии 2 пакета SSH). В отличие от SSH, программный механизм FreeS/WAN осуществляет шифровку всего трафика между двумя

сегментами WAN.

Компоновка и установка FreeS/WAN

Чтобы установить FreeS/WAN, вы должны встроить его в ядро Linux. Для этой цели вам прежде всего потребуется чистый, никоим образом не модифицированный исходный код ядра. Исходный код ядра, входящий в комплект OpenLinux (включая исходный код, записанный на компакт-диск, прилагаемый к данной книге), для этой цели не подходит: компания Caldera использует модифицированный исходный код ядра, который нельзя скомпоновать вне рабочей среды RPM. Возможно, в дальнейших версиях ситуация изменится, однако если вы в любом случае будете использовать чистый, нетронутый код ядра, у вас будет меньше проблем. Получить такой код можно по адресу http://www.us.kernel.org или вы можете использовать любое зеркало этого web-узла. Создайте каталог (mkdir linux-2.2.14) и убедитесь в том, что ссылка linux указывает на этот каталог. Распакуйте исходный код ядра, перейдите в этот каталог, выполните компоновку и установку ядра в соответствии с требованиями вашей системы. Убедившись в том, что ядро работает (компонуется, устанавливается и загружается), вернитесь в каталог /usr/src.

ПРИМЕЧАНИЕ

Если при загрузке нового ядра у вас возникли проблемы, убедитесь в том, что используемый вами файл /etc/lilo.conf корректен, и вновь запустите lilo. Убедитесь в том, что загрузчик lilo видит ваше ядро.

Находясь в каталоге /usr/src, распакуйте архив FreeS/WAN. После этого перейдите в подкаталог freeswan-1.1 и отдайте одну из следующих команд:

make menugo (соответствует make menuconfig)
make xgo (соответствует make xconfig)
make ogo (соответствует make config)
make oldgo (соответствует make oldconfig)

В результате выполнения одной из этих команд исходный код ядра будет исправлен и начнется компиляция и сборка ядра. Выберите сетевые параметры из категории Networking Options, которые вы намерены установить (не сбрасывайте параметр Kernel/User netlink socket). Если вы намерены использовать данную систему как выделенный маршрутизатор, имеет смысл установить параметр IP: optimize as router not host (оптимизировать как маршрутизатор, а не как обычный узел), благодаря этому ваша система будет работать несколько быстрее. Если же настраиваемая вами система должна совмещать выполнение функций маршрутизатора WAN и рабочей станции, устанавливать данный параметр не следует, так как при этом будут возникать некоторые ошибки пакетов.

Вы можете использовать также IP: advanced router options (дополнительные параметры маршрутизатора) и выбрать некоторые из дополнительных параметров. Если вы выбрали эту возможность, в процессе начальной загрузки вы должны отключить rp_filter. Для этого в файл /etc/rc.d/rc.local необходимо добавить следующую строку:

```
echo "0" > /proc/sys/net/ipv4/conf/all/rp_filter
```

Ближе к концу раздела Networking Options вы можете обратить внимание на некоторые новые параметры. Первый параметр имеет отношение к шифрующему протоколу IPSEC и называется IP Security Protocol (FreeS/WAN IPSEC). Соответствующий код может быть либо встроен в ядро, либо скомпилирован как отдельный модуль. Включив этот параметр, вы получите возможность воспользоваться несколькими другими параметрами, имеющими отношение к IPSEC. В большинстве случаев нет необходимости изменять значения по умолчанию. Если вы хорошо представляете себе предназначение и смысл этих параметров, можете настроить их в соответствии с вашими предпочтениями, однако вы ни в коем случае не должны включать параметр IPSEC: Enable Insecure algorithms (IPSEC: разрешение на использование незащищенных алгоритмов). Если вы включите этот параметр, теряется смысл использования зашифрованного канала связи.

Когда вы закончите настройку, сохраните новую конфигурацию ядра. Вы должны сохранить новую конфигурацию ядра вне зависимости от того, внесли ли вы в нее какие-либо изменения, в противном случае необходимые для использования FreeS/WAN изменения не будут внесены в исполняемый код ядра. После того как вы сохраните конфигурацию и завершите работу с утилитой настройки, автоматически начнется процедура компоновки ядра. Однако в процессе компоновки не будут выполнены все необходимые действия. Поэтому когда компоновка завершится, вы должны выполнить команду cd ../linux, а затем выполнить команду make modules_install. После этого следует переместить ядро, запустить lilo, чтобы загрузчик распознал новое ядро, и, наконец, перезагрузить систему с использованием нового ядра.

В процессе сборки ядра в системе будут установлены также несколько утилит и страницы электронной документации man pages.

Конфигурирование FreeS/WAN

Чтобы рассказать вам о настройке FreeS/WAN, я начну с самой простой конфигурации — требуется обеспечить защищенный канал обмена данными между двумя системами, подключенными к одному и тому же сетевому кабелю. В дальнейшем эта конфигурация будет расширена и усовершенствована. На каждом этапе конфигурирования FreeS/WAN вы должны будете проверять конфигурацию, чтобы убедиться в том, что она работает корректно.

В процессе загрузки ядра и инициализации системы на консоли, скорее всего, появится несколько сообщений, связанных с IPSEC. До тех пор пока вы не настроите IPSEC, вы должны игнорировать эти сообщения. Первое, что вы должны сделать, это отключить IPSEC на всех ваших системах до тех пор, пока вы не будете готовы соединить их.

Для выполнения первого этапа конфигурирования я буду использовать две системы: сетевой узел HostA с IP-адресом 192.168.0.1 и сетевой узел HostB с IP-адресом 192.168.0.2. Обе эти системы подключены к одному сетевому кабелю. Тестовое соединение между ними будет называться HostA2HostB (если бы я хотел быть действительно оригинальным, я назвал бы его foo-test). Каждая из этих систем использует для соединения с сетью Ethernet интерфейс eth0.

На узле HostA уберите файл /etc/ipsec.conf по умолчанию (по умолчанию в этом файле располагаются значения, подразумевающие слишком «мягкую» безопасность) и вместо него используйте следующие значения:

```
config setup
interfaces="ipsec0=eth0"
#(later you can change the above to interfaces="ipsec0=eth0 ipsec1=ppp0"
klipsdebug=all #(or none if you like)
plutodebug=all #(again, none if you like) conn HostA2HostB
HostA=192.168.0.1
HostB=192.168.0.2
keyingtries=0 # this is actually a very large number
```

Обратите особое внимание на отступы. Отступы являются важным атрибутом большинства файлов, связанных с ipsec. Скопируйте этот файл на узел HostB. После этого отредактируйте файл ipsec.secrets следующим образом:

192.168.0.1 192.168.0.2 "между кавычками следует разместить 256 случайно выбранных битов, полученных при помощи команды 'ipsec ranbits > tmpfile'"

ВНИМАНИЕ

Вы должны обеспечить надежную защиту файла ipsec.secrets. Этот файл должен принадлежать пользователю root, а правом его чтения должен обладать только пользователь root. Любой, кто получит в свое распоряжение этот ключ, сможет проникнуть в ваш защищенный туннель и читать любые ваши зашифрованные сообщения.

Чтобы случайным образом сгенерировать биты для файла ipsec.secrets, вы можете воспользоваться командой ipsec ranbits. Обе системы (192.168.0.1 и 192.168.0.2), упомянутые в строке файла ipsec.secrets (в эту строку можно добавить дополнительные системы), должны обладать идентичными копиями этого файла (по крайней мере должны совпадать биты 256-битного ключа). Убедитесь в том, что 256 бит размещаются между кавычками.

После того как на обе системы скопированы оба конфигурационных файла ipsec (с корректным набором разрешений на доступ к файлу /etc/ipsec.secrets), воспользуйтесь командой modeprobe для того, чтобы добавить в систему модуль ipsec (если вы скомпоновали ipsec в виде модуля), и выполните команду /etc/rc.d/ init.d/ipsec start на каждой из систем.

Чтобы убедиться в том, что ipsec функционирует, можно проверить несколько признаков. Во-первых, воспользовавшись утилитой ipconfig, вы можете обнаружить, что в системе появился новый интерфейс ipsec0 со всеми соответствующими параметрами. Кроме того, в таблице маршрутизации возникли несколько новых записей, связанных с интерфейсом ipsec0. Наконец, вы можете получить некоторую диагностическую информацию при помощи самой утилиты ipsec. Для этого достаточно использовать команды ipsec look и ipsec tncfg.

Расширение сети

Если взглянуть в файл ipsec.conf по умолчанию, можно обнаружить несколько дополнительных переменных, таких, как, например, переменная nexthop (следующий участок ретрансляции). В нашем конфигурационном файле соответствующие переменные могут обладать именами HostAnexthop или HostBnexthop. Если две системы, обменивающиеся зашифрованной информацией, физически разделены несколькими участками ретрансляции (несколькими хопами) Интернета (или любой другой сети) — иными

словами, если адрес шлюза одной системы не совпадает с адресом шлюза другой системы, — тогда переменная nexthop должна содержать IP-адрес шлюза для каждой из этих машин. Если для связи с Интернетом вы используете PPP, у вас в распоряжении будет локальный и удаленный IP-адрес соединения «точка-точка». Локальный адрес будет принадлежать сетевому узлу HostA (надеюсь, вы придумаете более осмысленное имя), а удаленный адрес — это значение переменной HostAnexthop. Ваш узел HostB (а в конфигурационном файле по умолчанию — rightnexthop) содержит IP-адрес nexthop другого узла. И конечно же, IP-адрес каждой из систем должен быть упомянут в файле ipsec.secrets.

На текущий момент мы с вами сформировали туннель, напрямую соединяющий две системы. Это весьма ограниченная конфигурация, однако вместе с тем она самая простая из всех возможных. Если вы хотите расширить туннель таким образом, чтобы включить другие сетевые узлы, расположенные за двумя системами, участвующими в созданном вами соединении, вы должны добавить в конфигурацию еще несколько переменных.

Прежде всего вам потребуется сетевая маска для подсети, расположенной за шлюзом. Эту сетевую маску следует присвоить переменной HostAsubnet= в формате *сеть/битовая_маска,* например, 192.168.0.0/24. Эта запись указывает на то, что узел HostA выполняет функции шлюза для всех систем с IP-адресами в диапазоне 192.168.0. Соответственно в конфигурации каждой из этих систем узел HostA должен быть указан в качестве шлюза.

ВНИМАНИЕ

Если вы используете переменную subnet, шлюзовая система исключается из указанного вами диапазона. Иными словами, трафик, исходящий от шлюзовой системы или адресованный шлюзовой системе, не будет передаваться через туннель. Чтобы исправить ситуацию, вы должны либо настроить отдельный туннель, в котором не будет переменной subnet, либо указать 192.168.0.1/32.

Также следует определить, выполняет ли шлюз функции брандмауэра (является ли он маскирующим узлом или нет). Если да, вы должны добавить переменную firewall и присвоить ей значение yes: HostAfirewall=yes.

Добавление дополнительных сетей

Чтобы увеличить количество подсетей в вашей защищенной сети WAN (то есть чтобы соединить защищенными каналами три и более подсетей), необходимо расширить файлы ipsec.conf и ipsec.secrets. Наиболее сложной является задача обеспечения уникальности IP-адресов в рамках вашей защищенной сети. Например, если в состав вашей сети WAN входит 50 немаршрутизируемых подсетей (например, из диапазона адресов 192.1б8.x.x), необходимо добиться того, чтобы каждый из узлов, входящих во все эти подсети, обладал бы уникальным IP-адресом. Вторая сложная задача — придумывание пар имен для осмысленного именования виртуальных соединений.

Например, допустим, вы хотите соединить три подсети. Три шлюзовых узла будут именоваться rnoe, larry и curly. Узел moe будет обладать файлом ipsec.conf с двумя разделами conn, содержимое которого может быть таким, как показано в листинге 21.4.

Листинг 21.4. Фрагмент файла ipsec.conf без раздела config setup

```
conn   moe-larry
       moe=192.168.0.1
       moenexthop=Internetgatewaymaewest
       moesubnet=192.168.0.0/24 #remember, moe is not secure to larry
       moefirewall=yes
       larry=192.168.1.1
       larrysubnet=192.168.1.0/24
       larrynexthop=Internetgatewaymaeeast
       larryfirewall=yes
       keyingtries=0
conn    moe-curly
       moe=192.168.0.1
       moenexthop=Internetgatewaymaewest
       moesubnet=192.168.0.0/24 #remember, moe is not secure to larry
       moefirewall=yes
       curly=192.168.2.1
       curlysubnet=192.168.2.0/24
       curlynexthop=Internetgatewaymaesouth
       curlyfirewall=yes
       keyingtries=0
```

На узле larry можно использовать точную копию показанного в листинге раздела conn moe-larry, а на узле curly — точную копию раздела conn moe-curly. При этом каждый из узлов larry и curly должен

обладать разделом conn larry-curly, в котором будет описываться соединение между этими двумя узлами.

Файл ipsec.secrets должен содержать два дополнительных набора секретов. Узел тое должен обладать общим ключом с узлом larry и еще одним общим ключом с узлом curly, однако ключ, используемый для обмена данными между узлами larry и curly, узлу тое не требуется (вы, конечно, можете добавить соответствующий ключ на узел тое, однако этот ключ просто не будет использоваться). Подобное правило обмена ключами действует и в отношении двух других серверов IPSEC.

И конечно же, если вы хотите обеспечить защищенную передачу трафика, исходящего от одного из шлюзовых узлов (например, тое) и адресованного другому шлюзовому узлу (например, larry), вам потребуется определить еще одно соединение (например, moe-larry-gates), которое формирует еще один дополнительный простой туннель. Для этого туннеля необходимо либо не определять переменные moesubnet и larrysubnet, либо определить данный туннель специально для соответствующего узла (192.168.0.1/32 и 192.168.1.1/32).

OpenSSH

На момент написания данной книги пакет OpenSSH находится в стадии разработки. Автор этого пакета намерен разработать программное средство OpenSSH, которое можно будет использовать в качестве полноценной замены пакета SSH. Отличие состоит в том, что по сравнению с SSH лицензирование пакета OpenSSH будет более открытым и его развитие будет продолжено (в настоящее время осуществляется поддержка ssh-1.2.27, однако развитие этого пакета не осуществляется, — вместо него разрабатывается пакет версии 2.0.x, в отношении которого действует коммерческая схема лицензирования).

Программное средство OpenSSH наряду с обычными возможностями SSH позволит воспользоваться защищенными версиями протоколов FTP и RCP. Разработчики отказались от использования любого криптографического кода, с которым могут быть связаны патентные проблемы (например, IDEA или RSA), однако при этом им удалось обеспечить надежную криптографическую защиту.

Заключение

В данной главе я рассказал вам о том, как настроить, скомпоновать и установить программный пакет SSH — программное средство для защищенного удаленного администрирования систем и взаимодействия между системами. Вы узнали о том, как осуществляется настройка и использование данного программного средства. После этого я рассказал вам о том, как скомпилировать пакет FreeS/WAN и включить его в состав ядра Linux, а также выполнить настройку этого программного механизма для того, чтобы обеспечить надежное соединение WAN через публичную сеть, или просто соединить зашифрованным каналом две системы.

Часть IV
Аудит системы безопасности

22

Конфигурирование syslog

В данной главе рассматриваются следующие вопросы:
- демоны syslog и klog;
- настройка файла syslog.conf;
- что такое каналы протоколирования и приоритеты;
- как узнать об имеющихся каналах протоколирования и приоритетах;
- конфигурирование syslogd;
- параметры syslogd;
- настройка параметров в процессе начальной загрузки.

Программный механизм под названием syslog, осуществляющий документирование (или, по-другому, *протоколирование)* сообщений о функционировании системы в журналах, может стать для вас бесценным источником полезных сведений о вашей системе. Все службы с ограниченным доступом обладают средствами взаимодействия с syslog. Однако зачастую сетевые администраторы пренебрегают документированием сведений в журнале. Зачастую они знают, что такой механизм существует, однако они недостаточно хорошо понимают, как он работает и как его настроить для конкретной ситуации.

В данной главе я расскажу вам о том, что такое syslog, что он делает, как он работает и как вы сможете приспособить его для решения ваших задач. В вашей системе syslog изначально настроен с использованием конфигурации по умолчанию, однако редактирование этой конфигурации для многих кажется черной магией, поэтому многие игнорируют широчайший набор возможностей, которыми может наделить сетевого администратора механизма syslog. Надеюсь, что после того как вы ознакомитесь с данной главой, ситуация изменится в лучшую сторону.

В операционной среде Linux одновременно функционируют два механизма протоколирования сведений в журналах. Первый из них является системным (system logger), второй используется для слежения за ядром (kernel logger). В обычных условиях механизм слежения за ядро передает все сообщения системному механизму протоколирования. В результате сообщения, имеющие отношение к ядру, смешиваются с ситемными сообщениями. При желании вы можете настроить механизм слежения за ядром (klog) таким образом, чтобы он направлял все сообщения в отдельный файл, не совпадающий с syslog, однако эта настройка должна осуществляться в момент запуска. Необходимо изменить файл /etc/syslog/daemons/syslog следующим образом. Вместо строки

```
OPTIONS_KLOG="-k /boot/System.map-'uname  -r ' "1
```

необходимо разместить в файле строку

```
OPTIONS_KLOG="-k /boot/System.map-'uname -r'   -f /var/log/kmesg"
```

Ключ -f и путь к файлу указывают на то, в каком файле следует записывать документируемые сообщения.

СОВЕТ

Изучите файлы, расположенные в каталоге /etc/sysconfig/daemons/, так как все демоны, запускаемые в процессе начального запуска, конфигурируются в этом каталоге.

Почему протоколируемые сообщения ядра по умолчанию передаются демону syslog, а не записываются в отдельный файл? Дело в том, что демон klogd не обладает возможностью удаленного ведения журнала. Если в одной сети располагаются несколько статических систем, зачастую удобно использовать одну из этих систем в качестве центрального места хранения и ведения всех журналов для всех систем, вместо того чтобы содержать журналы на разных компьютерах. Более подробно об этой возможности будет рассказано далее.

Базовая конфигурация syslog

Чтобы понять, что именно протоколируется, когда и куда заносится соответствующая информация, необходимо владеть несколькими базовыми концепциями, имеющими отношение к протоколированию сведений в журналах. Механизм протоколирования оперирует тремя основными понятиями: *канал протоколирования* (facility), *приоритет сообщений* (priority) и *местоположение журнала* (logging location).

Канал syslog — это специальный канал для внутренних сообщений демона syslogd. Специальные определения каналов протоколирования содержатся в файле /usr/ include/sys/syslog.h. Фрагменты этого файла показаны в листинге 22.1. Комментарии переведены на русский язык.

Листинг 22.1. Фрагмент файла /usr/include/sys/syslog.h, в котором определяются каналы протоколирования syslog

```
/* коды каналов протоколирования */
fdefine LOG_KERN (0<<3) /* kernel messages - сообщения ядра*/
fdefine LOG_USER (1<<3) /* random user-level messages • сообщения пользовательского режима */
#define LOG_MAIL (2<<3) /* mail system - почтовая систена */
#define LOG_DAEMON (3<<3) /* system daemons - системные демоны */
продолжение^
#define LOG_AUTH (4<<3) /* security/authorization messages */
                                /*сообщения безопасности/авторизации */
#define LOG_SYSLOG (5<<3) /* messages generated internally by syslogd */
                                  /* внутренние сообщения syslogd */
#define LOG_LPR (6<<3) /* line printer subsystem - подсистема печати LPR */
#define LOG_NEWS (7<<3) /* network news subsystem - сетевая подсистема */
#define LOG_UUCP (8<<3) /* UUCP subsystem - подсистема UUCP */
#define LOG_CRON (9<<3) /* clock daemon - демон счетчика времени */
#define LOG_AUTHPRIV (10«3) /* security/authorization messages (private) */
                                     /*сообщения безопасности/авторизации (закрытые) */
#define LOG_FTP (11<<3) /* ftp daemon - демон ftp*/
        /* остальные коды вплоть до 15 зарезервированы для системного использования */
#define LOG_LOCALO (16<<3) /* зарезервировано для локального использования */
#define LOG_LOCAL1 (17<<3) /* зарезервировано для локального использования */
#define LOG_LOCAL2 (18<<3) /* зарезервировано для локального использования */
#define LOG_LOCAL3 (19<<3) /* зарезервировано для локального использования */
#define LOG_LOCAL4 (20<<3) /* зарезервировано для локального использования */
#define LOG_LOCAL5 (21<<3) /* зарезервировано для локального использования */
#define LOG_LOCAL6 (22<<3) /* зарезервировано для локального использования */
#define LOG_LOCAL7 (23<<3) /* зарезервировано для локального использования */
#define LOG_NFACILITIES 24 /* текущее количество каналов протоколирования */
#define LOG_FACMASK Ox03f8 /* маска для извлечения кода канала */
        /* канал pri */
#define LOG_FAC(p) (((p) & LOG_FACMASK) >> 3)
#ifdef SYSLOG_NAMES CODE facilitynames[] =
{
{    "auth", LOG_AUTH },
{    "authpriv",  LOG_AUTHPRIV },
{    "cron",  LOG_CRON },
{    "daemon",  LOG_DAEMON },
{    "ftp",  LOG_FTP },
{    "kern", LOG_KERN },
{    "Ipr",  LOG_LPR },
{    "mail",  LOG_MAIL },
{    "mark",   INTERNAL_MARK },   /* INTERNAL -  канал внутреннего пользования*/
{    "news",  LOG_NEWS },
{    "security",  LOG_AUTH },  /* DEPRECATED • устаревшее */
{    "syslog".  LOG_SYSLOG }.
{    "user",  LOG_USER },
{    "uucp",  LOG_UUCP },
{    "local0",  LOG_LOCALO },
{    "local1". LOG_LOCAL1 },
{    "local2",  LOG_LOCAL2 },
{    "local3",  LOG_LOCAL3 },
```

```
{    "local4",   LOG_LOCAL4 },
{    "local5",   LOG_LOCAL5 }.
{    "Iocal6",  LOG_LOCAL6 },
{    "local7",   LOG_LOCAL7 },
{    NULL,   -1 }
}:
#endif
```

Информация из этого файла включается разработчиком в состав каждой службы, которая должна взаимодействовать с механизмом syslog. Доступность механизма протоколирования является базовым правилом для всех служб с ограниченным доступом и сетевых служб. В первой части листинга 22.1 содержатся выражения #define, при помощи которых определяются числовые коды для каждого из каналов протоколирования. В разделе facilitynames каждому коду канала ставится в соответствие символьное имя этого канала. Имена каналов, содержащиеся в массиве facilitynames, используются для конфигурирования демона syslog. Вы должны знать, какой из каналов используется той или иной службой для протоколирования.

Многие имена каналов говорят сами за себя, к тому же предназначение каналов объясняется в комментариях, сопровождающих определения кодов каналов. Для каналов группы local (с порядковыми номерами от 0 до 7) никаких объяснений нет. Эти каналы используются службами, у которых нет своих собственных каналов. Многие службы используют каналы, не соответствующие именам этих служб. Например, служба telnet не имеет собственного канала, однако для протоколирования эта служба использует каналы auth и authpriv, так как любой пользователь, подключающийся к системе через telnet, должен пройти через процедуру аутентификации, основанной на анализе пары «пользовательское_имя/пароль».

С другой стороны, программа pppd, работающая на уровне привилегий root (благодаря чему она обладает возможностью модифицировать таблицы маршрутизации, читать привилегированные файлы и т. п.), не обладает специальным связанным с ней каналом. Поэтому она протоколирует сообщения в канале daemon, а если ее компиляция была выполнена с использованием параметра debug, протоколирование отладочных сообщений осуществляется в канале local2. Версия pppd, входящая в состав OpenLinux, по умолчанию компилирована с использованием параметра debug. Однако отладочные сообщения по умолчанию не записываются в local2. Этот режим необходимо включить явно. Для этого необходимо внести в файл /etc/ppp/options строку kdebug 1 или разместить эту команду в командной строке, которая вызывает pppd.

Если программа, использующая механизм протоколирования, не обладает собственным связанным с ней специальным каналом, разработчик может выбрать один из каналов по собственному усмотрению. В некоторых случаях выбор канала можно изменить, перекомпилировав программу.

Вторым элементом системы протоколирования является *приоритет* протоколируемых сообщений. Каждое сообщение обладает уровнем приоритета, в зависимости от которого это сообщение либо вносится в журнал, либо отбрасывается. Далеко не все возможные сообщения для каждого канала вносятся в журнал. Приоритеты также определяются в файле /usr/include/sys/syslog.h, как показано в листинге 22.2.

Листинг 22.2. Фрагмент файла /usr/include/sys/syslog.h, в котором определяются приоритеты сообщений syslog

```
/* приоритеты (в порядке убывания важности)
 */
#define LOG_EMERG 0 /* дальнейшее использование системы невозможно */
#define LOG_ALERT I /* необходимо немедленно принять меры */
#define LOG_CRIT 2 /* критическое состояние */
#define LOG_ERR 3 /* состояние ошибки */
#define LOG WARNING 4 /* предупреждение */
#define LOG_NOTICE 5 /* состояние нормальное,  но произошло нечто важное */
#define LOGJNFO 6 /* информационное сообщение */
#define LOG_DEBUG 7 /* отладочное сообщение */

#define LOG_PRIMASK 0x07 /*маска для извлечения приоритета (для внутреннего использования)*/
  /* извлечение приоритета */
#define LOG_PRI(p)   ((p)  & LOG_PRIMASK)
#define LOG_MAKEPRI(fac,  pri)   (((fac) <<3)   |   (pri))
#ifdef SYSLOG_NAMES
#define INTERNAL_NOPRI 0x10 /* приоритет "no priority",  то есть "без приоритета" */
  /* mark "facility" */
#define  INTERNAL_MARK  LOG_MAKEPRI(LOG_NFACILITIES,   0) typedef struct _code {
char *c_name;
int c_val; } CODE:

CODE prioritynames[] =
```

```
{
{ "alert",  LOG_ALERT },
{ "crit",  LOG_CRIT },
{ "debug",  LOG_DEBUG },
{ "emerg",  LOG_EMERG },
{ "err",  LOG_ERR },
{ "error", LOG_ERR }, /* DEPRECATED - устаревшее */
{ "info",  LOG_INFO },
{  "none",   INTERNAL_NOPRI },   /* INTERNAL */
{ "notice",  LOG_NOTICE },
{ "panic", LOG_EMERG }, /* DEPRECATED - устаревшее */
{ "warn", LOG_WARNING }, /* DEPRECATED - устаревшее */
{ "warning", LOG_WARNING },
{ NULL, -1 }
}:
#endif
```

Здесь можно видеть коды и соответствующие им символьные имена для восьми уровней приоритета, которые могут использоваться службами для оповещения механизма syslog о собственном состоянии^ В листинге 22.2 эти уровни перечислены в порядке убывания важности.

В составе OpenLinux используется стандартный конфигурационный файл /etc/syslog.conf. Этот файл содержит в себе записи, относящиеся к наиболее часто конфигурируемым службам, и указывает, какие файлы журналов, содержащиеся в каталоге /var/log, соответствуют этим службам. Стандартный конфигурационный файл показан в листинге 22.3. Комментарии переведены на русский язык.

Листинг 22.3. Файл /etc/syslog.conf, используемый в OpenLinux по умолчанию

```
#Все сообщения ядра протоколируются прямо на консоль.
#Если вы будете протоколировать на консоль какие-либо другие данные,
#       вы рискуете заполнить весь экран информацией,
kern.* /dev/console
#Протоколировать все на уровне info и выше за исключением сообщений,
#связанных с электронной почтой,  новостями,
#а также закрытых аутентификационных сообщений в файле messages
*.info;news,mai1,authpriv,auth.none •/var/log/messages
#       Протоколировать отладочные сообщения
*.debug;news,mail,authpriv,auth.none -/var/log/debug
#Файл для закрытых сообщений аутентификации обладает ограничениями на доступ
authpriv.*;auth.* /var/log/secure
#Канал auth в двух предыдущих правилах является устаревшим,
#       однако он до сих пор используется.
#Все сообщения, связанные с почтой, протоколируются в одном месте
mail.* /var/log/mail
#Демон innd блокирует /var/log/news
#(вместо того, чтобы использовать /var/log/news.d)
#поэтому
#news.* /var/log/news.all
#Сохраняем сообщения об ошибках uucp и news уровня err и выше
#в специальном файле.
#uusp,news.err /var/log/spooler
#Сообщения уровня emergency получают все,
#кроне того, они протоколируются на другом компьютере.
 *.emerg *
#*.emerg @loghost
```

Строки, начинающиеся с символа #, являются комментариями и игнорируются демоном syslog. Комментарии помогают понять, какие именно сведения протоколируются и в каких файлах размещаются соответствующие записи.

Записи файла syslog.conf обладают следующим форматом:

```
канал.приоритет       путь/к/файлу/журнала
```

В самом начале записи указывается канал, в который поступают сообщения. Далее, после символа точки указывается уровень приоритета. Это минимальный допустимый уровень приоритета протоколируемых сообщений. Иначе говоря, если не указано что-либо иное, то в соответствии с данной записью протоколируются сообщения, уровень приоритета которых либо равен, либо выше указанного уровня. После этого указывается полный путь к файлу журнала. Все сообщения, связанные с указанным каналом, уровень приоритета которых либо равен, либо выше указанного уровня, будут записываться в указанный файл.

Приведенный формат используется лишь для самых простых записей конфигурационного файла

syslog, однако во многих случаях используются файлы более сложного вида. Конечно, вы можете выделить для каждой используемой вами комбинации каналов и приоритетов отдельную запись, однако зачастую удобнее скомбинировать несколько записей, имеющих отношение к одному и тому же файлу журнала, в одну запись. Для начала определите, чего у вас меньше: каналов, которые вы намерены протоколировать, или каналов, которые вы не намерены протоколировать. Использовать следует наиболее короткий из этих списков. Каналы в списке разделяются символом запятой.

ПРИМЕЧАНИЕ

В одной записи может содержаться несколько определений. В этом случае последующие определения перекрывают собой предыдущие. Тогда определяемые в них комбинации каналов/приоритетов либо добавляются к предыдущим определениям, либо исключаются из них. Благодаря этому появляется возможность определять явно заданные исключения. Учитывая то, что символом звездочки (*) обозначаются либо все каналы, либо все приоритеты, вы можете формировать короткие и весьма эффективные правила.

Например, если вы хотите протоколировать в файле /var/log/lnm сообщения из каналов lpr, news и mail с уровнем приоритета info или выше, вы можете использовать для этой цели следующую запись:

```
Ipr,news,mail.info                    /var/log/lnm
```

Эта запись аналогична следующей:

```
Ipr.info;.news.info;.mail.info                    /var/log/lnm
```

СОВЕТ

Символами-разделителями являются символы запятой и точки с запятой. Запятая используется для разделения имен каналов. Если в списке указываются разделенные запятыми уровни приоритетов, все уровни, за исключением последнего, игнорируются. Символ точки с запятой используется для разделения полных определений вида канал/приоритет.

С другой стороны, если вы хотите протоколировать все сообщения, за исключением сообщений cron и news в файле /var/log/messages, вовсе не обязательно перечислять в записи все каналы, за исключением каналов cron и news. Достаточно воспользоваться исключающим определением:

```
*.*;news,cron,authpriv.none      /var/1og/messages
```

Первое определение (*.*) предписывает протоколировать абсолютно любые сообщения. Однако далее в строке располагается специально определенный список каналов news,cron,authpriv с уровнем приоритета попе. Уровень приоритета попе является специальным уровнем, указывающим демону syslogd на то, что для данных каналов ни один из уровней не представляет интереса и не должен протоколироваться. Таким образом, в рассматриваемой записи первое определение (*.*) разрешает протоколирование всех сообщений, однако второе определение (news,cron,authpriv.none) отменяет протоколирование сообщений любого приоритета для каналов news, cron и authpriv.

ВНИМАНИЕ

Канал authpriv является чрезвычайно важным для безопасности системы. Сообщения из этого канала должны записываться только в хорошо защищенный (доступный для чтения записи только на уровне привилегий root) файл, так как эти сообщения содержат в себе пароли.

Совместно с приоритетами допускается использование двух дополнительных специальных символов. Первым специальным символом является восклицательный знак (!), который обозначает «отрицание», то есть все уровни приоритета, за исключением указанных. Второй специальный символ — это знак равенства (=), который обозначает единственный указанный уровень приоритета и никаких других. Например, если вы хотите, чтобы сообщения из всех каналов с уровнем приоритета debug (и никаким другим) записывались бы в файл /var/log/debug, вы можете воспользоваться следующей записью:

```
*.=debug                    /var/log/messages
```

Если вы также хотите, чтобы в этот же самый файл протоколировались сообщения с уровнем приоритета notice, после приведенной ранее записи добавьте запись:

```
*.!notice                    /var/log/debug
```

Если при всем при этом вы также желаете, чтобы все сообщения за исключением сообщений info записывались бы в файл /var/log/noinfo, вы можете добавить в конфигурацию также следующую запись:

```
*.!=info                    /var/log/noinfo
```

В данном случае оба специальных символа (! и =) использовались для того, чтобы исключить из всего возможного набора сообщений только сообщения с уровнем info. Имейте в виду, что если вы используете оба символа, вы должны располагать их так, как показано в примере, то есть символ

восклицательного знака должен располагаться на первом месте.

До сего момента в рассматриваемых нами примерах записей указывался конкретный файл, в который предписывалось вносить протоколируемые сообщения. По умолчанию после каждой записи в файл демон syslogd выполняет команду sync (синхронизация) для того, чтобы сохранить содержимое буферов из памяти на диск. Благодаря этому в случае сбоя системы по последним содержащимся в журналах записям вы сможете нащупать причину возникновения проблемы. Если после записи в файл журнала протоколируемого сообщения не осуществить сброс содержимого буферов на диск, может случиться так, что запись о событии, произошедшем непосредственно перед сбоем, не будет физически записана в файл на диске.

Таким образом, сброс буферов на диск — важная составляющая процесса протоколирования сообщений. Однако в связи с этим вы должны учитывать также один немаловажный нюанс. Если протоколируемые сообщения пересылаются на один центральный сервер протоколирования и если при этом сразу же после получения каждого из сообщений будет выполняться синхронизация буферов с диском, производительность сервера может заметно снизиться и другие работающие на нем процессы замедлят функционирование. Чтобы решить проблему, необходимо отключить выполнение немедленной синхронизации сразу же после записи сообщения в файл, однако сделать это следует лишь для тех сообщений, которые не считаются особенно важными. Чтобы явно отключить немедленную синхронизацию, следует поставить перед именем файла символ дефиса (-). Вот пример подобной записи из стандартного файла syslog.conf, входящего в комплект поставки Caldera OpenLinux:

```
*.debug;news,mail,authpriv,auth.none -/var/log/debug
```

Эта запись указывает на то, что выполнение немедленной синхронизации сразу после записи отладочного сообщения в файл /var/log/debug не является обязательным.

Существует еще одна спецификация, которую можно использовать в качестве цели при протоколировании сообщений. Эта спецификация имеет формат @имя_узла. В качестве имени узла указывается имя вашего центрального протоколирующего сервера. При этом все сообщения, к которым относится данная запись, направляются указанному узлу в порт UDP с номером 514.

ПРИМЕЧАНИЕ

Узел, принимающий протоколируемые сообщения, должен быть настроен на прием таких сообщений, иначе они будут отбрасываться.

Сообщения, принимаемые через сеть центральным протоколирующим сервером, будут обрабатываться работающим на нем демоном syslogd в соответствии с расположенным на этом сервере файлом /etc/syslog.conf. Сообщения без указания места назначения (отсутствует определение канал.приоритет или соответствующая сообщению пара канал.приоритет исключена) будут отбрасываться.

Вы можете использовать, например, следующую простую запись:

```
*. *                @центральный - протоколирующий - сервер
```

При этом центральный протоколирующий сервер берет на себя все заботы, связанные с тем, какие сообщения протоколировать, и куда их записывать. Однако имейте в виду, что при этом на сеть создается дополнительная нагрузка.

ВНИМАНИЕ

Центральный протоколирующий узел должен блокировать неавторизированные клиенты, посылающие ему сообщения syslog. Мало того, журналы должны располагаться в файловой системе, которая не является корневой. Благодаря этим мерам вы исключите возможность атаки типа Denial of Service, нацеленной на то, чтобы заполнить файлы журналов и остановить работу протоколирующего сервера.

Наконец в качестве цели для протоколируемых сообщений можно указать список пользователей. Имена пользователей в списке должны разделяться символом запятой. Как правило, подобным образом обрабатываются только сообщения с приоритетом emergency. Если в момент поступления сообщения пользователь не подключен к системе, сообщение ему передано не будет. Символом звездочки (*) обозначаются все пользователи. При этом используется команда wall, которая осуществляет широковещательную передачу сообщения всем подключенным пользователям в локальной сети. Сообщения с уровнем приоритета emergency зарезервированы системой для безвыходных ситуаций, поэтому, скорее всего, вы захотите настроить syslog таким образом, чтобы сообщение этой категории было принято вами вне зависимости от того, какую пользовательскую учетную запись вы использовали для подключения.

Демон syslogd

Настройка файла syslog.conf — это только половина всего дела. Некоторые настройки можно выполнить только при помощи командной строки при запуске демона syslogd. Ранее мы с вами уже использовали эту возможность: в ходе создания тюрьмы с измененным корнем для сервера DNS мы использовали ключ -a и указывали путь к дополнительному файлу журнала. Однако это был лишь один из множества возможных аргументов командной строки.

Конфигурационным файлом по умолчанию для демона syslogd является файл /etc/syslog.conf. Этот файл можно переопределить при помощи ключа -f, за которым следует указать полный путь к альтернативному конфигурационному файлу.

Чтобы приказать демону syslogd выполнять функции центрального сервера протоколирования, следует использовать ключ -r. По этой команде демон syslogd начинает принимать сообщения от клиентов. Имейте в виду, что syslogd принимает протоколируемые сообщения через порт UDP с номером 514, поэтому syslogd должен работать на уровне привилегий root, так как только на этом уровне можно связать порт с номером ниже 1024. Также следует учитывать, что по умолчанию syslogd не осуществляет пересылку сообщений, принимаемых им от клиентов, на другие сетевые узлы. Иначе говоря, если в файле /etc/syslogd.conf в качестве цели для сообщения, принятого от клиента, указан удаленный сетевой узел, сообщение не будет передано этому узлу, если только не был использован ключ -h.

Благодаря ключу -h сообщения syslog можно передавать от клиента к серверу, а затем еще одному серверу. В нормальных условиях эта возможность вряд ли когда может потребоваться, однако если некоторые из ваших систем находятся вне зоны, защищенной при помощи брандмауэра, а протоколирующий сервер расположен за брандмауэром, и при этом брандмауэр не осуществляет перенаправление UDP-порта с номером 514, то вы можете использовать брандмауэр в качестве ретранслятора сообщений syslog. Такую конфигурацию нельзя назвать идеальной, однако она вполне работоспособна.

Если вы осуществляете протоколирование на центральном сервере, сообщения будут вноситься в журнал с указанием полных доменных имен компьютеров, с которых они поступили. В результате некоторые записи могут оказаться чрезмерно длинными. Проблему можно решить при помощи ключа -l. После этого ключа указывается перечень разделенных запятыми имен сетевых узлов, для которых syslog будет вносить в журнал только сокращенное имя сетевого узла (без добавления полного имени домена). Однако есть и другой метод: вы можете использовать ключ -s, после которого следует указать перечень разделенных запятыми имен доменов. При внесении сообщений в журнал имена этих доменов в журнал вноситься не будут.

Два эти ключа работают по-разному, хотя и'близки по назначению. По команде -l демон syslog будет заносить в журнал только короткое имя узла. По команде -s демон syslog отбрасывает от полного доменного имени узла некоторую часть. Предположим, что требуется внести в журнал доменное имя roadrunner.marketing. acme.com. При использовании команды -l и указании узла roadrunner в журнал попадет только краткое имя этого узла — roadrunner. Если же использовать команду -s acme.com, от полного доменного имени сетевого узла будет отсечено имя домена acme.com, и в журнал попадет только оставшаяся часть имени, а именно roadrunner.marketing. Эта возможность может оказаться полезной в случае, если в вашей сети используются субдомены и в некоторых из них встречаются дублирующиеся имена сетевых узлов.

Еще одним полезным параметром командной строки является ключ -p. При помощи этого ключа вы можете изменить сокет (имя или местоположение), через который демон syslogd осуществляет запись. Однако используя эту возможность, вы не сможете повысить защиту вашей системы. По умолчанию используется сокет с именем /dev/log. Существует очень мало причин, по которым может потребоваться скрыть данный канал (pipe). Любой пользователь системы, обладающий возможностью запустить утилиту netstat, при желании сможет обнаружить этот канал. Это видно в листинге 22.4, где показан сокращенный вывод утилиты netstat.

Листинг 22.4. Фрагмент вывода команды netstat -a, в котором показан канал syslog

```
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags Type State I-Node Path
unix 0 [ ACC ] STREAM LISTENING 12556 /dev/log
unix 1 [ ] STREAM CONNECTED 12560 /dev/log
```

Правом записи в этот сокет и правом чтения из него обладает любой пользователь, поэтому скрывать его не имеет смысла. Основная цель злоумышленников — удаление записей из журнала. По этой причине правом записи в журналы должен обладать только пользователь root (или пользователь, от лица которого работает syslog).

ПРИМЕЧАНИЕ

Для повышения уровня безопасности демон syslogd можно запустить от имени непривилегированного пользователя,

однако в этом случае его нельзя будет использовать в качестве центрального механизма протоколирования для всей сети. Дело в том, что UDP-порт с номером 514, используемый для приема сообщений syslog через сеть, может быть связан только на уровне привилегий root.

Совместно с демоном syslogd можно использовать и многие другие параметры командной строки, однако здесь я расскажу вам еще только об одном ключе. Ключ -m используется для добавления в файлы журналов пометок о времени. В числе прочих механизмом syslog используется канал mark, который на самом деле не является обычным каналом. Этот канал используется только для того, чтобы добавлять в журналы пометки о времени. Эти пометки многим могут показаться раздражающими, однако на самом деле, проанализировав их или обнаружив их отсутствие, вы сможете сделать вывод о том, что с файлами журналов кто-то производил несанкционированные операции. По умолчанию пометки о времени добавляются в файлы журналов через каждые 20 минут, однако длительность этого промежутка можно изменить при помощи ключа -m, к которому в качестве аргумента необходимо добавить желаемую длительность временного интервала. Если установить временной интервал равным нулю, пометки о времени не будут добавляться в файлы журналов.

Что искать в файлах журналов?

После того как система протоколирования настроена, возможно, у вас возникнет вопрос, а что именно следует искать внутри файлов журналов? Другой, более грамотный вопрос: какую именно информацию следует получать от системы протоколирования для обеспечения приемлемого уровня защиты и как следует настроить систему протоколирования, чтобы получить от нее эту информацию? В большинстве случаев с точки зрения безопасности вас будут интересовать действия локальных пользователей, а также удаленные подключения к вашей системе.

Конечно же, система протоколирования является емким источником информации о функционировании самых разнообразных работающих в системе служб. Зачастую при помощи журналов syslog можно выполнить отладку конфигурации, а также получить сведения о неправильно работающих процессах. Зачастую сообщения, протоколируемые syslog, весьма детально описывают возникшую в системе проблему. Однако все это в первую очередь относится не к безопасности, а к системному администрированию, поэтому я не буду затрагивать всех этих вопросов в данной книге.

С точки зрения безопасности вам прежде всего необходимо знать, используют ли локальные пользователи команду su или приглашение на подключение к системе для того, чтобы получить доступ на уровне привилегий root. По содержимому журналов вы можете определить, когда некто подключается к системе в качестве обычного пользователя, а затем при помощи команды su переходит на уровень привилегий root. Чаще всего для получения доступа к системе на уровне привилегий root используются социальные контакты и умение общаться с людьми, то есть так называемая «социальная инженерия». Именно этот метод использовался печально известным взломщиком Кевином Митником (Kevin Mitnick) для получения несанкционированного доступа к системам. Чтобы улучшить защиту, ограничьте подключение на уровне root к локальным терминалам, полностью запретите подключение через telnet. В OpenLinux такой порядок вещей установлен в конфигурации по умолчанию.

Что касается внешних подключений, вы должны реагировать на попытки сканирования вашей системы в поисках открытых портов. Чтобы попытаться взломать вашу систему, злоумышленник должен вначале узнать, какими службами он может воспользоваться для этой цели. Таким образом, он будет пытаться подключиться к вашей системе через различные порты. Эти попытки вы сможете обнаружить, просмотрев содержимое журналов. Вы увидите, что некто последовательно подключается к разнообразным портам вашей системы. Чуть позже, возможно, поступает запрос о статусе сервера, при этом клиент не пытается использовать данный сервер. Это может указывать на то, что взломщик пытается определить разновидность и версию серверной программы, чтобы точнее подобрать инструменты, которые потребуются ему для взлома. Получив эти сведения и выполнив быстрый поиск уязвимых мест, злоумышленник может приступить к взлому. Сама попытка может выглядеть в журналах syslog как соединение с клиентом, через которое получены некорректные данные. Возможно, это попытка организовать переполнение буфера. Некорректные данные — это бессмысленный набор символов, среди которых можно обнаружить как бы случайно оказавшийся там полный путь к некоторому исполняемому файлу.

ССЫЛКА

Сканирование сетей обсуждается в главе 25 «Использование средств наблюдения за сетью».

В лучшем случае может произойти одно из двух:

а) возникает сбой серверной программы, выводится содержимое некоторого

участка оперативной памяти и программа завершает работу;

б) программа принимает неправильные данные, отказывается от их обработки, заносит их в журнал и продолжает нормальное функционирование.

В любом из этих случаев можете считать, что вам повезло, — замысел злоумышленника провалился. Однако может произойти и другое: в момент своего аварийного завершения программа запускает командную оболочку на уровне привилегий root, к которой получает удаленный доступ злоумышленник. Слишком легко? Именно по причине такой легкости значительная часть Интернета в последнее время превратилась в настоящий детский сад, заполненный малолетними хулиганами.

ССЫЛКА

В главе 23 о просмотре и анализе файлов журналов рассказывается подробнее.

Заключение

В этой главе я рассказал вам о том, как настраивается механизм протоколирования данных о работе системы. Вы познакомились с форматом записей файла /etc/syslog.conf. Вы также узнали о том, какими бывают каналы и уровни приоритета и откуда они берутся. Вы узнали о том, как составлять спецификации протоколируемых сообщений, и о том, как указывать разнообразные цели для записи этих сообщений.

Затем я рассказал вам о параметрах командной строки демона syslogd. Вы также узнали о том, как настроить центральный сервер протоколирования и предотвратить атаку Denial of Service, направленную на этот сервер.

23 Просмотр и анализ журналов syslog

В данной главе рассматриваются следующие вопросы:
- типы файлов журналов и их местоположение;
- разрешения на доступ к файлам журналов;
- как работает механизм протоколирования;
- использование dmesg;
- просмотр журналов в формате, отличающемся от ASCII.

В предыдущей главе я рассказал вам о том, как управлять механизмом syslog и как настроить его таким образом, чтобы он вносил в файлы журналов именно ту информацию, которая вам нужна. В данной главе я подробнее расскажу вам об этих файлах журналов.

Большая часть файлов журналов являются простыми текстовыми файлами ASCII. Для просмотра таких файлов вы можете воспользоваться любым текстовым редактором. Также вы можете просто вывести их содержимое на экран или в окно /terminal при помощи утилиты cat. Однако как только вы приступите к анализу содержимого любого из этих файлов, вы обнаружите множество однообразных сообщений, большую часть которых вам захочется игнорировать. Они вносятся в журнал на всякий случай, однако если попытаться просмотреть их все в поисках чего-либо ненормального, у вас могут возникнуть сложности.

Требуется некий способ анализа базовой информации о вашей системе для того, чтобы оперативно определять, что является нормальным, а что вызывает подозрения. При работе с файлами журналов вы имеете дело с сотнями или с тысячами (а возможно, и более того) записей ежедневно. Очевидно, что для упрощения этой работы удобнее использовать некий сценарий, позволяющий отбирать нужные вам записи и скрывать от ваших глаз ненужные записи (после того как вы поймете, что они вас не интересуют).

ВНИМАНИЕ

Если вы обнаруживаете что-либо странное, например запись, содержащую следующий текст: H^N^L^K^^N^H^V^L^^?^?^?/bin/sh^^?^?^?^?^?^?^?^?^?^?имейте в виду, что в отношении вашей системы была предпринята попытка взлома (возможно, успешная, а возможно — нет). Немедленно примите меры, так как комбинация символов /bin/sh в середине бессмысленного набора символов — это не простая случайность.

Файлы, расположенные в каталоге /var/log

Как отмечалось в предыдущей главе, каталог /var/log является местоположением по умолчанию для системных файлов журналов. В большинстве случаев, если некоторый файл отсутствует, механизм syslog автоматически создает его. Однако существует пара файлов, которые не создаются демоном syslog автоматически -их следует создать вручную (об этом будет рассказано позже).

Если syslog автоматически создал некоторый отсутствующий ранее файл журнала, это еще не означает, что данный файл был создан правильно. Вы должны внимательно изучить содержимое каталога /var/log. Необходимо проанализировать, какая информация записывается в каждый из файлов и кто должен обладать правом просмотра этой информации.

Если в отношении файлов журналов назначены неправильные наборы разрешений, достаточно выполнить простую операцию chmod. Некоторые из файлов могут использоваться опытными пользователями или пользователями, осуществляющими компиляцию и запуск своих собственных программ, которые могут выводить отладочную информацию или какие-либо результаты работы в syslog. Для таких пользователей, возможно, будет необходимо разрешить просматривать некоторые из файлов журналов. Для каждой отдельной сети и для каждого отдельного узла решение о просмотре тех или иных файлов журналов следует принимать по отдельности. Если вы используете центральный протоколирующий сервер, будет лучше, если вы ограничите к нему доступ и запретите подключение к нему обычных пользователей.

Некоторые из файлов журналов не должны быть доступны для широкой публики. Любой из файлов, в который попадают сообщения из канала authpriv.*, может содержать в себе пароли. В листинге 23.1

показан пример записи файла secure, в которой содержится пароль.

Листинг 23.1. Фрагмент файла /var/log/secure

```
Oct 3 16:22:54 volcan login[1549]:  FAILED LOGIN 1 FROM (null) FOR nypassword.  User not known to the underlying authentication module
```

Показанная в листинге запись сообщает, что некто с именем mypassword пытался подключиться к системе, но получил отказ, так как пользователя с таким именем в системе не существует. Что же произошло? На самом деле пользователь по ошибке ввел вместо имени свой пароль, естественно, система не узнала его. Подобное часто случается с людьми, которые привыкли работать с Windows. Операционная система Windows запоминает имя последнего подключавшегося к ней пользователя и по умолчанию предлагает имя этого пользователя в приглашении на подключение к системе. Таким образом, пользователь привыкает к тому, что в большинстве случаев для того, чтобы войти в систему, ему надо ввести только пароль. Все это — остаточный менталитет идеологии «одна система, один пользователь», на базе которой была построена большая часть систем Windows. Пользователи привыкают сообщать системе только пароль, подразумевая, что система уже знает, кто они есть. В нашем случае пользователь вводит свой пароль вместо имени и получает отказ — первая попытка подключения проваливается. Однако запись, которая следует сразу же за записью, указанной в листинге 23.1 сообщает нам, что пользователь root успешно подключился к системе. Таким образом, в первой из рассматриваемых записей содержится пароль, а в следующей — имя пользователя, обладающего данным паролем. Не надо быть гением, чтобы сообразить, какой пользовательской учетной записи соответствует этот пароль. Любой человек, обладающий возможностью просмотра данного файла журнала, получит в свое распоряжение открытые ворота, пропускающие его в систему.

СОВЕТ

Если вы используете программные средства, осуществляющие автоматическую смену файлов журналов, убедитесь в том, что используемый вами программный механизм создает файл-замену с корректным набором разрешений на доступ.

Содержимое каталога /var/log, который должен быть зарезервирован исключительно для системы протоколирования, должно выглядеть приблизительно так, как показано в листинге 23.2.

Листинг 23.2. Содержимое каталога /var/log

```
drwxr-xr-x 2 root      root    1024 Oct 27 00:00 atsar
drwxr-x--- 2 root      root    1024 Oct 25 12:42 httpd
-rw-r--r-- 1 root root  1702 Aug 5 09:06 kdm
-rw-r--r--  1 root     root    146876 Oct 27 06:50 lastlog
-rw ------- 1 root     root    180264 Oct 27 09:34 mail
-rw -------  1 root     root    323304 Oct 25 11:58 mail.0.gz
drwxr-xr-x 2 majordom majordom 1024 Jul 27 19:55 majordomo
-rw -------  1 root     root    1425209 Oct 27 09:39 messages
-rw -------  1 root     root    1667546 Oct 25 12:05 messages.0.gz
drwxr-xr-x 3 news      news    1024 Aug 5 08:52 news.d
drwxr-xr-x 2 root      root    1024 Jul 28 00:00 samba.d
-rw -------  1 root     root    331161 Oct 27 09:38 secure
-rw ------- 1 root     root    0 Aug 5 09:04 spooler
drwxr-x--- 2 root      root    1024 Jul  14 20:08 squid
drwxr-xr-x 2 uucp      uucp    1024 Jul  27 20:53 uucp
-rw-rw-r-- 1 root      utmp    1276032 Oct 27 09:37 wtmp
-rw-r--r--  1 root     root    4193 Oct 25 11:16 xdm-errors
-rw -------  1 root     root    52626 Oct 27 08:57 xferlog
```

Обратите внимание, что часть файлов открыты для всеобщего обозрения. Любой пользователь обладает возможность прочитать их содержимое. На данной системе эти файлы используются некоторыми пользователями и не содержат никаких данных, критичных для безопасности системы. Отдельные файлы, в особенности те, которые расположены в подкаталогах, создаются программами, не являющимися syslog. В некоторых случаях они принадлежат другим пользователям. Как правило, если программа должна создать файл журнала, этот файл должен принадлежать пользователю, от имени которого работает данная программа. Если бы файлы и подкаталоги принадлежали пользователю root, некоторые программы, такие как uucp, majordomo и т. п., не смогли бы осуществить запись в файл или подкаталог, если только этот файл или подкаталог не является доступным для записи для всего окружающего мира. А это не самая лучшая идея.

Как работает система протоколирования

Работая с syslogd, вы должны понимать, что происходит, когда syslogd начинает работу. Демон syslogd начинает работу на самых ранних стадиях загрузки системы. Это происходит потому, что механизм syslog должен как можно раньше приступить к протоколированию происходящих в системе событий.

Прежде всего syslog читает содержимое файла /etc/syslog.conf (информация, содержащаяся в этом файле, описывается в предыдущей главе). После этого syslog создает сокет (по умолчанию /dev/log), через который будет осуществляться запись, после чего сокет соединяется со всеми файлами журналов, упоминаемыми в файле /etc/syslog.conf (или в другом конфигурационном файле, указанном в командной строке).

Когда syslogd готов приступить к записи сообщений, он читает содержимое кольцевого буфера ядра. Благодаря этому в журнал попадают все те сообщения, которые отображаются на экране в процессе загрузки системы и рассказывают о том, что было обнаружено системой. Демон syslogd не может начать работу до того, как загрузка ядра будет полностью завершена, поэтому до начала работы syslogd ядро сохраняет сообщения в буфере. Сообщения заносятся в буфер до тех пор, пока буфер не заполнится, после этого для того, чтобы занести в буфер новое сообщение, ядро стирает наиболее старое хранящееся в буфере сообщение.

Чтобы снизить вероятность взлома или какого-либо нарушения работы системы, вы можете запустить syslogd от лица непривилегированного пользователя. Однако прежде чем сделать это, вы должны учесть, что если syslogd работает на более низком, чем root, уровне привилегий, этот демон не сможет связать стандартный порт syslog с номером 514. Однако это вовсе не значит, что в этом случае вы не сможете использовать syslogd в качестве центрального протоколирующего сервера. Чтобы обеспечить прием протоколируемых сообщений через сеть, вы должны связать с syslogd порт с номером выше 1024, а затем перенаправить все UDP-паке-ты, поступающие через порт 514, в тот новый порт с непривилегированным номером. Для этого необходимо изменить определение службы syslog в файле /etc/services и использовать редиректор (например, ipchains или что-либо подобное) для перенаправления UDP-порта 514 в новый непривилегированный порт syslog. Если вы приказали демону syslogd открыть новый нестандартный порт (с использованием ключа -r) и он не может этого сделать, syslogd завершит работу, выдав сообщение об ошибке.

Изменив уровень привилегий, на котором работает syslogd, вы также должны убедиться в том, что все файлы каталога /var/log, запись в которые разрешена только пользователю root, доступны для записи пользователю, от имени которого работает демон syslogd. По умолчанию запись в подкаталоги и файлы /var/log разрешена только пользователю root. Вы должны изменить права на владение файлами журналов, в противном случае syslogd не сможет осуществить запись в эти файлы. После того как syslogd начал работу, файлы журналов постоянно находятся в активном состоянии. Иными словами, они постоянно открыты. В этом можно убедиться, воспользовавшись командой lsof, которая отображает список открытых файлов. В листинге 23.3 показаны выдержки из вывода команды lsof. Здесь вы видите записи, имеющие отношение к протоколированию, то есть содержащие в себе буквосочетание Log.

Листинг 23.3. Фрагмент вывода команды lsof, показывающий только записи, имеющие отношение к syslog и klog

```
COMMAND    PID   USER   FD      TYPE  DEVICE  SIZE NODE   NAME
syslogd   1040   root   cwd      DIR     3,1  1024  2      /
syslogd   1040   root   rtd      DIR     3,1  1024  2      /
syslogd   1040   root   txt      REG     3,1 27772 184444   /usr/sbin/syslogd
syslogd   1040   root   Ou     unixOxc7fecccO   118 22643    /dev/log
syslogd   1040   root   1u     unixOxc21d7000   118 22645    /home/dns/dev/log
syslogd   1040   root   2u     unixOxc5166360   118 65006    /dev/log
syslogd   1040   root   4w       CHR     4,0   123 388  /dev/tty0
syslogd   1040   root   5w       REG    3,11 524040 75780    /var/log/messages
syslogd   1040   root   6u     unixOxc2577960 52739 52   /dev/log
syslogd   1040   root   7u     unixOxc68bbOOO   118 19313    /dev/log
syslogd   1040   root   8u      inet     612     UDP *     :syslog
syslogd   1040   root   9u     unixOxc21d7960    61 5    /dev/log
syslogd   1040   root   10u    unixOxc214d680    65 1    /dev/log
syslogd   1040   root   11u    unixOxc214dccO    66 3    /dev/log
syslogd   1040   root   12u    unixOxc2c7a680 56748 55   /home/dns/dev/log
syslogd   1040   root   13u    unixOxc2577640    19 81   /dev/log
syslogd   1040   root   14w      REG     3,1 333293 75968    /var/log/secure
syslogd   1040   root   15w      REG     3,1 191389 75781    /var/log/mail
syslogd   1040   root   16w      REG     3,1      0 75970    /var/log/news.all
syslogd   1040   root   17w      REG     3,1      0 75971    /var/log/spooler
klogd     1043   root   cwd      DIR     3,1  1024  2      /
klogd     1043   root   rtd      DIR     3,1  1024  2      /
klogd     1043   root   txt      REG     3,1 19932 184443   /usr/sbin/klogd
klogd     1043   root   0r       REG     0,2      0 5      /proc/kmsg
```

```
klogd    1043    root    1u              unixOxc21d76      40 614   socket
```

Проанализировав самую правую колонку, можно обратить внимание, что несколько процессов подключены через сокет /dev/log. Используется также сокет /home/dns/dev/log, который мы с вами создали в главе 14, об этом сокете я не буду рассказывать подробно. Запись *:syslog показывает, что syslogd связывает порт syslog для приема сообщений от других серверов. Этот порт можно легко заметить в разделе UDP вывода команды netstat -an.

В листинге 23.3 можно легко заметить записи, относящиеся к каждому из файлов журналов. В частности, показаны файлы messages, mail, news.all, secure и spooler.

Демон syslogd находится в постоянном контакте с этими файлами, поэтому их нельзя просто так переместить, переименовать или стереть. Если активный файл журнала перемещается или переименуется, перемещенный или переименованный файл продолжает принимать сообщения. Иными словами, если вы переименовываете файл messages в файл messages.0, а затем выполняете команду touch messages, файл messages.O будет продолжать увеличиваться в размерах. Иначе говоря, в момент запуска syslogd между этим демоном и файлом messages создается канал передачи данных, в момент переименования файла канал между демоном и этим файлом сохраняется. Файл можно редактировать, однако канал передачи данных все равно соединен с этим файлом до тех пор, пока syslogd не будет перезапущен.

Чтение файлов журналов

Просмотр содержимого большинства файлов журналов (по крайней мере тех, которые записываются демоном syslogd) не представляет труда. Типичная запись в файле журнала syslog выглядит следующим образом:

```
Oct 3 10:56:18 volcan sshd[5915]: log: Generating 768 bit RSA key.
```

Формат всегда один и тот же. В первой колонке указывается дата и время внесения в журнал записи. Время является локальным временем сервера. Во второй колонке указывается имя сервера, который сделал эту запись. Часто в этой колонке указываются полные доменные имена внешних по отношению к протоколирующему серверу сетевых узлов, однако формат указываемых здесь имен можно изменить при помощи ключа командной строки (см. предыдущую главу). В рассматриваемом случае во второй колонке указано имя локальной системы — volcan. Следующая колонка содержит имя и идентификатор PID демона, создавшего эту запись. Идентификатор процесса PID (Process ID) указывается не всегда. В нашем случае создателем записи является демон sshd с идентификатором PID, равным 5915. Далее идет собственно сама запись, то есть текст сообщения, переданного демоном. В нашем случае запись показывает, что демон защищенной командной оболочки sshd (Secure Shell Daemon) с идентификатором PID 5915 сгенерировал ключ шифрования длиной 768 бит для обеспечения соединения через сеть.

Таким образом, пользователь root может свободно читать любой журнал, созданный демоном syslogd. Для этого не требуется никаких специальных программ. Однако некоторые программы, предназначенные для чтения сообщений, все же существуют. Первая такая программа называется dmesg. На самом деле программа dmesg не читает содержимое файла журнала. Вместо этого данная утилита читает кольцевой буфер ядра. Кольцевой буфер — это обычный буфер, который спроектирован таким образом, что сразу же после его заполнения новые данные начинают записываться в его начало и далее, затирая при этом информацию, которая располагалась в начале буфера до этого. Таким образом, добавление новых данных в буфер происходит по кольцу, переходя из конца в начало буфера.

Утилита dmesg возвращает последние 8192 байт из кольцевого буфера ядра (в документации указывается число 8196, однако на самом деле это не так). Это соответствует размеру кольцевого буфера в старых ядрах версий 2.0.x. Начиная с версии 2.2.13 размер кольцевого буфера увеличился и стал равным 16384 байтам. Чтобы увидеть все содержимое буфера целиком, можно воспользоваться ключом -s за которым указывается количество отображаемых байт кольцевого буфера. Например, вы можете воспользоваться командой:

```
dmesg  -s16284
```

По этой команде на экран будет выведено полное содержимое кольцевого буфера. Если вы полагаете, что вам требуется более емкий буфер, вы можете модифицировать значение параметра LOG_BUF_LEN в файле /usr/src/linux/kernel/printk.c. Увеличивая размер буфера, вы должны сделать его кратным значению 8192.

СОВЕТ

Утилита dmesg читает содержимое кольцевого буфера ядра, а не содержимое файла /var/log/ messages, поэтому даже если система загружается с использованием строки init=/bin/sh, вы все равно сможете ознакомиться с диагностическими сообщениями, хранящимися в кольцевом буфере.

Совместно с утилитой dmesg можно использовать еще один из двух ключей (нельзя использовать эти два ключа одновременно). Любой из них можно использовать совместно с ключом -s. Во-первых, вы можете использовать ключ -c, который очищает кольцевой буфер. Во-вторых, вы можете использовать ключ -n<#>, при помощи которого вы можете сообщить утилите dmesg наименьший допустимый приоритет сообщений, выводимых на консоль. Например, если вы хотите отобразить только сообщения уровня приоритета emergency, вы можете указать -n 1. По мере увеличения этого числа разрешенный уровень приоритета понижается, и соответственно, все большее количество сообщений будет выводиться на экран. Пример вывода утилиты dmesg показан в листинге 23.4.

Листинг 23.4. Фрагмент вывода утилиты dmesg

```
ppp:  channel  pppO closing.
ppp0 released
ppp0:  ccp closed
hdb: ATAPI 24X CD-ROM drive,  120kB Cache
Uniform CDROM driver Revision:  2.56
VFS: Disk change detected on device ide0(3.64)
ISO 9660 Extensions:  Microsoft Joliet Level  3
ISOFS:  changing to secondary root
Uniform CD-ROM driver unloaded
PPP:  ppp line discipline successfully unregistered
CSLIP:  code copyright 1989 Regents of the University of California
PPP:  version 2.3.7 (demand dialling)
PPP line discipline registered.
registered device ppp0
```

Обратите внимание на то, что ни одно из этих сообщений не содержит в себе даты, времени или имени системы. Здесь вы видите только текст сообщения. Все остальные сведения добавляются демоном syslogd после того, как этот демон получает эти сообщения от ядра.

Файлы utmp, wtmp и last log

Файлы журналов, создаваемые демоном syslogd, а также некоторыми другими демонами, использующими свои собственные журналы (например, uucp, httpd и т. п.), являются легко читаемыми файлами в формате ASCII. Однако существуют и другие файлы журналов, которые содержат в себе не просто ASCII-текст, а бинарные данные. К этой категории относятся три файла: lastlog, utmp и wtmp. В этих файлах содержится информация, связанная с подключением пользователей к системе.

Если вы не обнаружите в вашей системе файлов utmp или wtmp, не удивляйтесь. Дело в том, что если эти файлы перемещены или удалены из системы или просто никогда не были созданы, система просто не осуществляет добавление в них записей. Иными словами, при отсутствии этих файлов механизм протоколирования сведений в них отключается. Такая возможность специально предусмотрена для тех, кто не желает пользоваться протоколированием сведений в этих файлах. Если же вы хотите использовать эти файлы, но в вашей системе они отсутствуют, значит, вы должны самостоятельно создать их. Для этого достаточно воспользоваться простой командой touch имя_файла, отданной в корректном каталоге. Файлы, о которых идет речь, содержат в себе информацию о подключении пользователей к системе и используются несколькими программами. Корректным каталогом для файла wtntp является каталог с именем /var/log, а корректным каталогом для файла utmp является /var/run.

ПРИМЕЧАНИЕ

Если файлы utmp и wtmp не будут созданы, ничего страшного не произойдет, однако имейте в виду, что при этом протоколирование сведений в них (эти сведения имеют отношение к использованию пользовательских учетных записей) будет отключено. Такой режим можно использовать для небольшой системы, к которой пользователи, как правило, не подключаются.

Файл utmp содержит в себе сведения о текущих подключениях к системе. Файл wtmp содержит исторические сведения о подключениях к системе. В файле lastlog содержится информация о последнем успешном подключении пользователя к системе, когда это произошло и из какого места пользователь подключился. Все эти файлы используются утилитами last, Lastlog, uptime, utmpdump, w и who. Файлы utmp, wtmp и lastlog содержат в себе данные в бинарном формате, поэтому их нельзя просмотреть напрямую обычным текстовым редактором, однако для просмотра их содержимого можно использовать одну из вышеупомянутых утилит. Например, если вы не хотите, чтобы пользователи обладали возможностью использовать утилиту who, вы можете изменить режим доступа к файлу utmp таким образом, чтобы он перестал быть доступным для чтения всем пользователям.

Несмотря на то, что в файлах utmp и wtmp содержится информация о подключении пользователей к системе, далеко не все программы обращаются к этим файлам. Некоторые программы, такие как xdm (или kdm), не могут или не должны создавать записи в файле utmp, однако при этом они создают записи в файле wtmp. Это объясняется способом, при помощи которого пользователь подключается к системе с использованием xdm или kdm. Ни одна из программ диспетчера экрана не обладает контролирующим TTY, поэтому процесс login не может составить для них корректную запись для файла utmp. Это означает, что программы xdm и kdm функционируют подобно службе FTP, которая вносит записи только в файл wtmp.

ПРИМЕЧАНИЕ

Контролирующий терминал процесса — это терминал, с которого был запущен этот процесс. Именно на контролирующий терминал направляются все выводимые этим процессом сообщения (если только не было выполнено специальное перенаправление стандартного потока вывода и/или стандартного потока вывода ошибок).

Программы, использующие utmp и wtmp, выводят на экран информацию о подключениях, перезапусках и т. п. В листинге 23.5 показан сильно урезанный вывод утилиты last.

Листинг 23.5 Вывод команды last, урезанный для краткости

```
david pts/2 volcan.pananix.c Thu Oct 28 08:50  - 08:54 (00:03)
david pts/1 volcan.pananix.c Thu Oct 28 07:52 still  logged in
david pts/1 volcan.pananix.c Wed Oct 27 22:38 - 22:55 (00:17)
david pts/2 Wed Oct 27 20:49 - 08:50 (12:00)
root tty2 Wed Oct 27 20:48 - 21:12 (00:23)
david ftp volcan.pananix.c Wed Oct 27 19:15 -  19:16 (00:01)
reboot system boot 2.2.13 Sun Oct 24 08:46 (4+00:18)
root tty1 Sun Oct 24 08:31 - crash (00:14)
```

Информация для этого листинга извлекается из файла wtmp. В каждой записи слева направо перечисляются: имя пользователя, контролирующий терминал (или ftp или system boot — загрузка системы), узел (если узел не является локальным, или версия ядра в случае, если указано system boot), дата и время подключения и отключения (или смещение GMT если указано system boot). Обратите внимание: в последней записи в завершающей колонке содержится метка crash (сбой). Это отнюдь не указывает на то, что система зависла. На самом деле это признак того, что было выполнено контролируемое завершение работы системы (shutdown). Однако сеанс подключения был прерван, так как изменился уровень запуска (runlevel), поэтому считается, что произошел ненормальный выход из сеанса.

Вывод (в сокращенной форме) утилиты lastlog показан в листинге 23.6. Здесь отображена информация, извлеченная из файла /var/log/lastlog, которая сопоставлена с информацией из файла /etc/passwd.

Листинг 23.6 Сокращенный вывод команды lastlog

```
Username Port From Latest
root tty2 Wed Oct 27 20:48:59 1999
bin **Never logged in**
daemon **Never logged in**
majordom **Never logged in**
postgres **Never logged in**
mysql **Never logged in**
Silvia ttyl Sat Aug 7 22:55:54 1999
david pts/2 volcan.pananix.c Thu Oct 28 08:50:18 1999
dns **Never logged in**
```

Утилита uptime использует информацию из файла utmp для формирования данных о подключениях, кроме того, она обращается к /proc для получения информации о процессах. Вывод утилиты uptime выглядит следующим образом:

```
10:11am up 2 days, 19:29, 1 user, load average: 0.00, 0.01, 0.00
```

Единственным полем, для формирования которого используется файл utmp, является поле, в котором указывается количество пользователей, в данном случае 1.

Программа utmpdump может обрабатывать любой файл, имя которого указано в командной строке, однако эта программа в состоянии интерпретировать только файлы /var/run/utmp и /var/log/wtmp, так как эти файлы обладают сходным форматом. Сокращенная интерпретация файла utmp показана в листинге 23.7, а сокращенная интерпретация файла wtmp — в листинге 23.8.

Листинг 23.7 Содержимое utmp

```
Utmp dump of /var/run/utmp
[8]  [00651]  [bw ][][][]  [0.0.0.0 ]  [Sun Oct 24 08:46:17 1999 EST]
[1]  [20021]  [~~ ]  [runlevel]  [- ] [ ]  [0.0.0.0 ]  [Sun Oct 24 08:46:17 1999 EST]
[8]  [00899]  [15 ][][][]  [0.0.0.0 ]  [Sun Oct 24 08:46:40 1999 EST]
```

```
[8] [09264] [P001] [ ] [pts/1 ] [ ] [0.0.0.0 ] [Wed Oct 27 22:55:41 1999 EST]
[7] [13696] [P001] [david ] [pts/1 ] [volcan.pananix.com] [0.0.0.0 ] [Thu Oct 28 07:52:07 1999 EST]
```

Файл utmp содержит информацию только о текущих сеансах. Можно заметить, что в листинге присутствуют записи об очень старых сеансах. Если сравнить содержимое wtmp и utmp, можно обнаружить, что в utmp присутствуют некоторые устаревшие записи. Они сохранились потому, что соответствующие сеансы были завершены некорректно. Если просматривать каждую запись слева направо, в первой колонке можно увидеть численный идентификатор. Это одноразрядное число, которое не имеет большого значения. Вторая колонка содержит PID процесса. Если вы произведете поиск в системе перечисленных здесь идентификаторов PID, вы обнаружите, что активным является только самый последний, 13696. В третьей колонке может содержатся одно из следующих значений: ~~, bw, число или буква и число. Эти метки обозначают соответственно: изменение уровня запуска или перезагрузку, процесс bootwait и либо номер TTY, либо комбинацию буквы и числа для PTY. Четвертая колонка может быть либо пустой, либо содержать имя пользователя, reboot (перезагрузка) или runlevel (уровень запуска) в зависимости от доступной информации. В пятой колонке указывается контролирующий TTY или PTY, если эти данные известны. В пятой колонке указывается имя удаленного узла. Если подключение осуществляется с локального узла, в этом поле ничего не указывается. В седьмой колонке указывается IP-адрес удаленной системы. В последней колонке содержится дата и время записи. Обе таблицы — utmp и wtmp — содержат одни и те же сведения.

Листинг 28.8. Сокращенный вывод wtmp Utmp dump of /var/log/wtmp

```
[2] [00000] [~~ ] [reboot ] [- ] [2.2.12 ] [0.0.0.0 ] [Fri Aug 27 20:40:16 1999 EST]
[8] [00631] [bw ][][][] [0.0.0.0 ] [Fri Aug 27 20:40:16 1999 EST]
[1] [20021] [~~ ] [runlevel] [~ ] [ ] [0.0.0.0 ] [Fri Aug 27 20:40:16 1999 EST]
[5] [00879] [15 ][][][] [0.0.0.0 ] [Fri Aug 27 20:40:16 1999 EST]
[7] [02976] [/3 ] [Silvia ] [pts/3 ] [ ] [0.0.0.0 ] [Fri Aug 27 23:35:42 1999 EST]
[8] [02753] [1 ] [ ] [tty1 ] [ ] [0.0.0.0 ] [Fri Aug 27 23:40:10 1999 EST]
[7] [13368] [ ] [david ] [ftpd13368 ] [volcan.pananix.com] [0.0.0.0 ] [Thu Oct 07 08:09:30 1999 EST]
```

В листинге 23.7 записи таблицы utmp расположены в хронологическом порядке. В листинге 23.8 записи таблицы wtmp также расположены в хронологическом порядке, однако этот порядок противоположен, то есть самые старые записи располагаются в конце таблицы.

Последний листинг этой главы — листинг 23.9 — показывает вывод команд w и who. Эти утилиты также используют для отображения информации сведения из файла utmp и системы /proc.

Листинг 23.9. Вывод команд w и who

```
[root@chiriqui]# w
  10:35am up 2 days,  19:52,  1 user,  load average: 0.00, 0.00,  0.00
USER TTY  FROM LOGIN@  IDLE JCPU  PCPU   WHAT
david pts/1 volcan.pananix.c 7:52am O.OOs 0.25s 0.02s w
[root@chiriqui]# who
david pts/1 Oct 28 07:52 (vo1can.pananix.com)
```

Команда w выводит несколько больший объем информации о системе и о пользователе, чем команда who. Однако данные, выводимые командой who, удобнее использовать в рамках сценариев. Обе команды показывают, что пользователь david подключен к псевдотерминалу 1 с 7 часов 52 минут утра 28 октября с удаленного узла с именем volcan. Команда w выводит сведения о том, какую нагрузку создает данный пользователь на локальную систему. Если этот пользователь ранее открывал другие сеансы, которые на текущий момент являются закрытыми, расход ресурсов для тех сеансов показан не будет. Команда w также показывает командный процесс, запущенный пользователем david, а также в первой строке сведения о времени, в течение которого система продолжала работу.

В предыдущих листингах показан основной режим работы каждой из утилит. Следует учитывать, что каждая из рассмотренных утилит обладает множеством ключей, при помощи которых можно изменить ее поведение. В зависимости от того, какие сведения протоколируются в журналах, из этих журналов вы можете почерпнуть большой объем информации о состоянии системы. Однако при этом ни в коем случае не следует забывать об одном очень важном обстоятельстве: нельзя полностью доверять журналам.

ВНИМАНИЕ

Любой, кто без вашего спроса получил доступ к системе на уровне привилегий root, может и, скорее всего, попытается отредактировать журналы так, чтобы скрыть свое присутствие и свои действия. Журналам можно доверять только в случае, если они выводятся напрямую на принтер или на любой другой не стираемый носитель информации.

Я хотел бы еще раз подчеркнуть это. Файлы журналов нельзя считать исчерпывающими и в полной мере аккуратными и точными. Содержимое этих файлов можно модифицировать, в них можно по своему усмотрению добавлять новые записи, кроме того, существующие в них записи можно стирать.

Содержащаяся в журналах информация зависит от того, какие демоны работают в системе, записывают ли они информацию в свои собственные журналы или передают ее демону syslog.

СОВЕТ

Протоколирование на системах повышенной важности (таких как брандмауэр) надежнее организовать таким образом, чтобы протоколируемые сообщения выводились напрямую на принтер. В этом случае злоумышленник не сможет модифицировать содержимое журналов, и вы получите возможность своевременно обнаружить его присутствие.

Заключение

В данной главе я рассказал о содержимом некоторых стандартных файлов журналов, существующих в вашей системе. Эту информацию нельзя считать точной и исчерпывающей, однако благодаря ей вы можете получить множество полезных сведений о функционировании системы. Теперь вы должны лучше понимать, как работает система протоколирования и как используются связанные с ней программы. Я также рассказал о некоторых недостатках системы протоколирования syslog.

24

Использование средств наблюдения за защитой сети

В данной главе рассматриваются следующие вопросы:
- защита вашей системы;
- начальная установка системы;
- повторяющиеся задачи.

Одной из наиболее сложных и обременительных задач, встающих перед администраторами-новичками (в особенности теми из них, кто никогда ранее не имел дела с какими-либо разновидностями Unix), является задача обслуживания системы и поддержки должного уровня безопасности. Зачастую манера, в которой они обеспечивают должную защиту системы, напоминает рассказ нескольких слепых о слоне. Некоторые из администраторов знают о том, что очень важно ежедневно проверять системные журналы, другие знают, что надо постоянно следить за целостностью системных файлов, третьи осведомлены, что следует периодически проверять, какие из программ связывают порты на уровне привилегий root. Однако многие забывают, что самое важное — это объединить все эти операции.

Некоторые из задач следует выполнять чаще других. На небольшой однопользовательской системе, подключающейся к Интернету время от времени, правила фильтрации пакетов которой настроены на блокирование всех привилегированных портов, многие операции могут выполняться раз в неделю или даже раз в месяц. Если точно такая же система соединена с Интернетом 24 часа в сутки семь дней в неделю, если к ней ежедневно подключается большое количество пользователей, те же самые операции по обслуживанию системы и обеспечению безопасности следует выполнять ежедневно.

Подытоживая эти рассуждения, следует отметить, что все зависит от многих факторов. Решения о том, насколько часто следует выполнять те или иные процедуры, принимает администратор системы. Эти решения должны быть продуманными и аргументированными.

Когда система загружена

Лучше всего начать обслуживание еще до того, как система будет подключена к сети. В этот период времени, то есть уже после того как система установлена и загружена, но еще до того как вы подключитесь к сети и позволите другим пользователям входить в систему (либо через сеть, либо при помощи консоли), вы можете быть в определенной степени уверены в том, что система работоспособна и не содержит в себе «троянских коней» и т. п.

Подразумевается, что вы доверяете используемому вами комплекту Linux. Такие компании, как Caldera и Red Hat, прикладывают массу усилий для того, чтобы установить и настроить систему наиболее безопасным образом. Например, если вы устанавливаете RPM-пакет netkit-telnet, сервер (демон telnet) и клиент устанавливаются с использованием наиболее безопасной конфигурации. При этом сервер активизирован, то есть способен отвечать на поступающие запросы.

Если вы нуждаетесь в использовании клиента telnet, но при этом вам не нужен сервер telnet, вы должны отключить сервер telnet. Однако будьте уверены, что разрешения на доступ к файлам и права на владения настроены должным образом.

Прежде всего следует провести инвентаризацию системы. В пакет dailyscript (название переводится как «ежедневный сценарий») входит программа с именем check-package (название переводится как

«проверка пакетов»), которая в процессе установки dailyscript устанавливается в /etc/cron.d/lib таким образом, чтобы запускаться ежедневно. Если вы установите и запустите этот пакет, в результате первого запуска на экране появится сообщение об ошибке. Это происходит потому, что база данных для выполнения сравнения еще не сформирована. При повторном запуске вы не обнаружите никаких изменений, однако сообщение об ошибке на экран выводиться не будет. Теперь каждый раз при запуске этого сценария на экран будет выводиться информация об обнаруженных изменениях. В полностью установленной системе Caldera для завершения работы сценария требуется около 15 минут. Если вы не желаете устанавливать dailyscript, вы можете выполнить проверку пакетов самостоятельно. Для этого достаточно воспользоваться двумя командами: rpm -qa и rpm -Va. Результат выполнения каждой из этих команд следует записать в отдельный файл. В результате вы получите два файла. Если вы захотите сравнить текущее состояние вашей системы с изначальным ее состоянием, вы должны снова выполнить две упомянутые команды, после чего результат выполнения каждой из команд следует сравнить с содержимым двух файлов, которые вы сгенерировали, когда система была только что установлена. Если все в порядке (обнаруженные отличия не вызывают у вас подозрений), вы можете сохранить два новых полученных файла для последующих проверок. Именно такую процедуру сравнения выполняет сценарий check-packages. Этот сценарий записывает полученные файлы на жесткий диск. Возможно, для большей надежности будет лучше сохранить эти файлы на гибком диске и защитить этот диск от записи.

Также рекомендуется сохранить в отдельный файл вывод команды:

```
find / -perm +600 -print
```

Благодаря этому вы получите список всех файлов SUID/SGID (каталоги и исполняемые файлы). Если вы будете периодически выполнять эту команду и сравнивать результат ее выполнения с изначальным файлом, вы сможете обнаружить, какие новые файлы SUID/SGID появились в вашей системе.

Помимо этого рекомендуется проверять файл /etc/passwd на наличие пользователей с идентификаторами UID или GID, равными нулю:

```
grep ":0:"  /etc/passwd
```

Идентификаторами UID и GID, равными нулю, должны обладать только пользователь root и несколько системных учетных записей.

Кроме того, вы должны проверить, что в системе не существует ни одной учетной записи с пустым паролем:

```
awk -F: ' { print $2 ":" $1 } ' /etc/shadow | grep "V - | sed "s/://g" -
```

Просмотрите содержимое файла/etc/inetd.conf и закомментируйте все ненужные службы, а затем перезапустите (или сделайте SIGH UP) демон inetd. После этого можно приступить к «укреплению» вашей системы. Если в процессе просмотра /etc/ inetd.conf вы обнаруживаете службу, смысла которой вы не понимаете, закомментируйте ее. Если после завершения такой чистки в файле/etc/inetd.conf осталось что-либо помимо ftp, pop3 и swat, значит, возможно, вы оставили там нечто лишнее.

Наконец, следует отредактировать файл /etc/aliases. Убедитесь в том, что запись ближе к концу файла изменена таким образом, что почта, адресованная пользователю root, перенаправляется какому-либо обычному пользователю. Не забудьте запустить newaliases.

Перекомпоновка ядра

Следующий шаг — перекомпоновка ядра. Для выполнения этой задачи лучше всего подключиться к Интернету, загрузить последнюю версию «чистого» исходного кода ядра и откомпилировать его. Скорее всего, в том, что вы используете ядро Caldera, нет ничего ужасного, однако, возможно, в комплект OpenLinux входит не самая свежая версия ядра, кроме того, в исходный код этого комплекта внесены модификации и исправления, поэтому подчас такое ядро сложно компоновать. Если ранее вы не имели опыта компоновки ядра Linux, будет лучше, если вы будете использовать для этой цели именно «чистые» исходные файлы. При этом вы за короткое время получите в свое распоряжение новое ядро, которое будет оптимизировано под интересующие вас задачи, и поэтому оно будет более эффективным и более быстрым. Кроме того, благодаря использованию «чистых» исходных кодов вы подготовите свою систему к установке пакета FreeS/WAN (если, конечно, вы нуждаетесь в использовании этого пакета), как об этом рассказывается в главе 21. К сожалению, очень многие имеющиеся на рынке комплекты Linux используют в своем составе модифицированное ядро и модифицированные исходные файлы. Это относится не только к Caldera, но и к Red Hat, Mandrake, SuSE и многим другим комплектам. Компании, занимающиеся распространением Linux, намеренно хотят предложить вам за ваши деньги более широкие возможности, однако пакет ядра является наиболее важной частью системы и к нему следует относиться с особенным вниманием. Если вы хотите быть уверенными в том, что ядро откомпилировано должным образом, используйте «чистые» исходники.

После того как вы скомпонуете ядро и загрузите его (в идеале — с поддержкой ipchains), вы можете

приступить к блокированию портов, которые, возможно, используются локальным узлом localhost, но которые вы не собираетесь делать доступными для внешних пользователей. На этом же этапе вы можете сделать некоторые из служб доступными для узлов вашей внутренней сети, и эти службы не обязательно должны быть внешними службами.

Установка и настройка ipchains

Если вы не установили ipchains, самое время сделать это. Настройку ipchains следует начать с блокирования портов с номерами ниже 1024. Лучше всего заблокировать все или по крайней мере большую часть этих портов. Затем, если в этом есть необходимость, вы можете приступить к открыванию портов. Открывать их следует по одному — по мере того как они становятся необходимыми. Все открываемые вами порты следует использовать для обеспечения работы относительно безопасных демонов. Имейте в виду, что наибольшее количество проблем возникает при использовании демонов telnet и imap. Если вы не чувствуете в себе уверенности в том, что вы сможете обнаружить злоумышленника, пытающегося проникнуть в систему через эти порты, лучше не открывать их для внешнего мира. В качестве вторичной защиты можно использовать tripwire.

ССЫЛКА
Более подробно об ipchains рассказывается в главе 16.

Ежедневные/еженедельные/ежемесячные процедуры обеспечения безопасности

В главах 22 и 23 вы узнали о механизме протоколирования syslog. Помимо проверки файлов SUID/SGID, целостности установленных файлов (rpm -qa и rpm -Va), поиска пользователей с идентификаторами UID/GID, равными нулю, и проверки пустых паролей вы должны уделять время анализу файлов журналов. Файлы журналов подскажут, что именно происходило в системе в то время, пока она была включена.

Если вы решили использовать пакет dailyscript (модифицированный для OpenLinux), возможно, вы захотите запускать этот сценарий один раз в день. Сценарий производит множество проверок корректности системы, он просматривает журналы в поисках аномалий, а также запускает сценарии проверки RPM (check-package) и осуществляет проверки журналов почты. После этого сценарий отправляет по почте пользователю root результаты своей работы (надеюсь, в вашей системе создан почтовый псевдоним учетной записи root, благодаря чему почта, адресованная root, перенаправляется обычному пользователю). Содержимое конфигурационного файла dailyscript показано в листинге 24.1.

ВНИМАНИЕ
Если вы используете программу, которая периодически выполняет смену файлов журналов, убедитесь в том, что сценарий dailyscript выполняется и завершает свою работу до того, как программа смены файлов журналов перемещает их на новое место.

Во-первых, сценарий dailyscript использует утилиту last (которая читает содержимое wtmp) для проверки списка пользователей, подключенных к системе. Если файл wtmp отсутствует, значит, этот раздел доклада dailyscript будет пустым. Некоторые записи также могут оказаться ошибочными. Например, там может указываться still logged in (до сих пор подключен) в то время, как на самом деле соединение с системой было завершено некорректно. Соединение, на самом деле являющееся удаленным, может показываться как удаленное или как локальное. В докладе будет показано, кто подключен к системе и в какое время произошло подключение.

Во-вторых, сценарий dailyscript просматривает файл /var/log/messages в поисках подозрительных сообщений, то есть таких, которые не соответствуют общему описанию (образцу, заданному при помощи регулярного выражения — regex). Благодаря этому вы получаете возможность просмотреть все странные сообщения из файла журнала /var/log/messages. После нескольких первых запусков вы поймете, что получаемый в результате такого поиска набор сообщений требует сокращения. Большинство исходящих сообщений ppp и chat не являются аномальными. Вы можете повлиять на критерий отбора подозрительных сообщений, разместив выбранные слова в конфигурационном файле /etc/dailyscript.conf в строке ALWAYSIGNORE="переменная". В качестве переменной следует указать перечень разделенных пробелами слов (как правило, это имена каналов syslog), которые следует игнорировать при просмотре

журналов. Таким образом, если параметр ALWAYSIGNORE содержит "ppp chat", вы не увидите связанных с ppp и chat сообщений, которые по большей части не представляют для вас интереса (см. листинг 24.1).

Листинг 24.1. Сокращенное содержимое файла /etc/dailyscript.conf

```
TMP2=/var/tmp
SERVICES="syslog named identd pam PAM_pwdb ftpd in.ftpd sshd kernel  talkd in.telnetd in.rlogind login xntpd CRON --"
SCRIPTDIR=/usr/local/sbin/dailyscript
SEDSCRIPTl=$SCRIPTDIR/sed-script-1
SEDSCRIPT2=$SCRIPTDIR/sed-script-2
MAILSTATS=$SCRIPTDIR/todays_stats
MAILLIST=$SCRIPTDIR/smtpstats
LOGFILES="/var/log/messages   /var/log/secure"
ALWAYSIGNORE="chat pppd cardmgr Pluto"
INCOMINGDIRS="/home/ftp/pub/incoming"
PERMDIR=/var/local/dailyscript
MAILLOG=/var/log/mail
```

Файл /etc/dailyscript.conf можно использовать для того, чтобы сообщить сценарию dailyscript, какие данные вас интересуют и в каких сведениях вы не заинтересованы. Очевидно, что информация, которая не была записана в журналы, не может быть подвергнута анализу и выведена в составе доклада. Однако вы можете контролировать то, каким образом осуществляется анализ журналов. Для этого можно изменить значения следующих параметров: LOGFILES сообщает сценарию dailyscript, какие файлы журналов следует анализировать; ALWAYS_IGNORE определяет, какие каналы syslog следует игнорировать; MAILLOG указывает на то, какой файл необходимо использовать для статистики электронной почты.

Единственным недостатком сценария является то, что он предназначен для использования на компьютере, выполняющем протоколирование самостоятельно. Его нельзя использовать для проверки центрального протоколирующего сервера.

Если в вашей сети используется центральный протоколирующий сервер, вы можете модифицировать dailyscript таким образом, чтобы он мог работать с журналами каждой из систем по отдельности. Сценарий dailyscript в том виде, в котором он входит в состав OpenLinux, смешивает в составе одного доклада записи, относящиеся к разным сетевым узлам. Чтобы решить проблему, проще всего создать цикл последовательного перебора систем. Этот цикл следует включить в сценарий сразу же после того, как из журналов извлечены записи текущего дня. Благодаря такой модификации сценарий будет формировать доклад отдельно для каждой системы. Обратите внимание, что файл wtmp не является одним из файлов журналов, записи которого направляются на центральный протоколирующий сервер, поэтому чтобы получить эквивалент доклада dailyscript для каждой из систем, вы должны на каждой из этих систем запустить сценарий, приведенный в листинге 24.2.

Листинг 24.2. Сценарий получения списка подключенных пользователей

```
D=`date -d "1 day ago" +"%b %d"  |  sed 's/ 0/  /' `
mail you@your.org -s "People who logged on to 'hostname' " <<EOF
$(last -ad |  grep $D | grep -v ".* *ftp"  | cut -c-22,34-  )
```

Чтобы модифицировать сценарий dailyscript для формирования отдельных докладов для каждой из систем, необходимо выполнить следующие действия:

1. Непосредственно перед строкой «# Set up temporary directory» разместите следующие две строки:

```
for s in 'echo $SYSTEMS'
do
```

В конце сценария необходимо добавить строку:

```
done
```

2. Далее в файл /etc/dailyscript.conf необходимо добавить следующую строку (между кавычками необходимо разместить имена ваших систем):

```
SYSTEMS="система1   система2"
```

Если вы хотите изменить набор сведений, которые заносятся каждой системой на центральный протоколирующий сервер, лучше всего воспользоваться для этой цели файлом /etc/syslog.conf на каждой из систем. В главах *21 к 22* рассказывалось, как именно следует изменить этот файл, чтобы определить набор интересующих вас сведений, которые следует передавать на центральный протоколирующий сервер. Если вы модифицируете порядок работы syslog, не забудьте выполнить перезапуск syslog.

СОВЕТ

Зачастую после установки утилиты смены журналов некоторые обнаруживают, что никаких журналов не возникает. Это происходит потому, что после перемещения журналов не выполняется перезапуск syslog. Вместо того чтобы перезапускать syslog, можно скопировать файл журнала, а затем выполнить команду cat/dev/null > файл_журнала для того, чтобы обнулить этот файл.

Просмотрев записи журналов, сценарий dailyscript приступает к просмотру записей PAM_pwdb. Осуществляется проверка записей, которые соответствуют подключению пользователей к системе. Эти записи извлекаются из файла /var/log/ secure. Здесь можно обнаружить, кто из пользователей использовал команду su. Также можно проверить подключения к системе с использованием учетной записи root.

Следующим проверяется демон named (предполагается, что вы используете bind). Если на своем узле вы не используете сервер имен, данный раздел доклада будет пустым.

Далее будут перечислены пересылки данных по протоколу FTP (в обе стороны), а также обращения к Identd. Раздел Identd должен быть пустым, если только вы не разрешаете запуск демона identd при помощи inetd. Демон Identd следует использовать только в том случае, если вы абсолютно точно не можете без него обойтись. Во всех остальных случаях этот демон следует отключить, так как он является источником сведений о вашей системе, которым могут воспользоваться взломщики и компьютерные хулиганы. Если вы вынуждены использовать identd, будет лучше, если вы разрешите доступ к нему только со стороны хорошо вам известных узлов, которым вы доверяете.

Название следующего раздела не соответствует действительности. В листинге раздел называется Kernel errors, однако в нем содержатся просто записи, имеющие отношение к ядру (если только вы не изменили допустимый уровень приоритета для данного журнала). Очень часто в данном разделе присутствует огромное количество информационных сообщений. Чтобы контролировать объем записываемой сюда информации, следует должным образом настроить /etc/syslog.conf.

Далее располагается доклад о программах, запускаемых при помощи cron. Код модифицирован таким образом, чтобы исключить atrun (если вы используете atrun и в вашей системе функционирует atd, значит, вам не нужны эти записи), а также ежедневные, еженедельные и ежемесячные запуски cronloop. Все эти программы являются программами, которым вы доверяете, поэтому вам не о чем беспокоиться.

Далее идет достаточно важный раздел — вывод утилиты df. Даже самые добросовестные администраторы могут позабыть о том, что необходимо следить за использованием диска. Сценарий dailyscript выполняет эту функцию.

Далее проверяются ресурсы, экспортируемые системой NFS. Никогда не бывает лишним проверить, что именно экспортирует NFS, хотя это и не так важно. Текущая версия dailyscript не включает в доклад вывод утилиты showmount, однако эту утилиту легко добавить в сценарий, кроме того, она может появиться в последующих версиях сценария.

Следующий раздел содержит в себе статистику, связанную с вашим почтовым сервером. Здесь вы найдете сведения об использовании электронной почты. Если вы обнаружите огромное количество почтовых сообщений, передаваемых некоторым сетевым узлом, это значит, что ваша система находится под атакой распространителей «грязной» почты (спаммеров). В доклад будет включена вся информация, необходимая для того, чтобы закрыть эту дыру.

После статистики, связанной с почтой, будет размещен отчет, генерируемый при помощи check-packages. Этот отчет включается в то же самое почтовое сообщение, однако он формируется вне сценария dailyscript Сценарий check-packages запускается самостоятельно. В листинге 24.3 показано, как приблизительно может выглядеть результат ежедневной автоматизированной проверки вашей системы.

ПРИМЕЧАНИЕ

В вашем случае отчет может оказаться более длинным. Листинг 24.3 был отредактирован мною для того, чтобы сэкономить место. Чтобы регулировать объем содержащейся в докладе информации, следует изменить файл /etc/syslog.conf или добавить каналы syslog в переменную ALWAYS_IGNORE конфигурационного файла /etc/dailyscript.conf.

Листинг 24.3. Сокращенное почтовое сообщение, формируемое сценарием dailyscript

```
************************************************************
General Daily Run -- chiriqui.pananix.com -- Dec 3
************************************************************
People who logged in:
david        22:38 - 11:51  (13:12)
david        20:48    still  logged in
david        20:39 - 21:44  (01:05)
root         15:40 - 15:40  (00:00)
david        15:30 - 15:31  (00:00)    volcan.pananix.com
david        09:22 - 09:45  (00:22)    volcan.pananix.com
root         07:17 - 07:28  (00:11)
david        07:04 - 09:22  (02:18)
************************************************************
Checking System Log Files .
##############################################
###  Unmatched entries in /var/log/messages!  ###
##############################################
```

Dec 3 07:05:08 chiriqui modprobe: modprobe: Can't locate module char-major-108
Dec 3 10:35:13 chiriqui ipsec_setup: Starting FreeS/WAN IPSEC 1.1...
Dec 3 10:35:13 chiriqui ipsec_setup: Loading KLIPS module:
Dec 3 10:35:18 chiriqui ipsec_setup: KLIPS debug 'all1
Dec 3 10:35:18 chiriqui ipsec_setup: KLIPS ipsec0 on eth0 192.168.0.2/255.255.255.192 broadcast 192.168.0.255
Dec 3 10:35:18 chiriqui ipsec_setup: Disabling core dumps:
Dec 3 10:35:18 chiriqui ipsec_setup: Starting Pluto (debug 'all'):
Dec 3 10:35:19 chiriqui ipsec_setup: Enabling Pluto negotiation:
Dec 3 10:35:19 chiriqui ipsec_setup: ...FreeS/WAN IPSEC started
Dec 3 15:40:37 chiriqui ipsec_setup: Stopping FreeS/WAN IPSEC...
Dec 3 15:40:37 chiriqui ipsec_setup: Shutting down Pluto:
Dec 3 15:40:38 chiriquiipsec_setup: Misc cleanout:
Dec 3 15:40:38 chiriqui ipsec_setup: ...FreeS/WAN IPSEC stopped
Dec 3 07:07:35 chiriqui ipop3d[17166]: connect from 127.0.0.1
Dec 3 07:17:00 chiriqui - root[15087]: ROOT LOGIN ON tty1
Dec 3 07:18:32 chiriqui ipop3d[17406]: connect from 127.0.0.1
Dec 3 10:35:18 chiriqui Pluto[18142]: Starting Pluto (FreeS/WAN Version 1.1)
Dec 3 10:35:18 chiriqui Pluto[18142]: | opening /dev/urandom
Dec 3 10:35:18 chiriquiPluto[18142]: | inserting event EVENT_REINIT_SECRET, timeout in 3600 seconds
Dec 3 10:35:18 chiriqui Pluto[18142]: | listening for Whack on /var/run/pluto.ctl, file descriptor 5
Dec 3 10:35:18 chiriquiPluto[18142]: | next event EVENT_REINIT_SECRET in 3600 seconds
Dec 3 10:35:19 chiriquiPluto[18142]: |
Dec 3 10:35:19 chiriqui Pluto[18142]: "received whack message
Dec 3 10:35:19 chiriqui Pluto[18142]: listening for IKE messages
Dec 3 10:35:19 chiriqui Pluto[18142]: | IP interface lo 127.0.0.1 has no matching ipsec* interface -- ignored
Dec 3 10:35:19 chiriqui Pluto[18142]: adding interface ipsecO/ethO 192.168.0.2
Dec 3 10:35:19 chiriqui Pluto[18142]: loading secrets from "/etc/ipsec.secrets"
Dec 3 10:35:19 chiriqui Pluto[18142]: | next event EVENT_REINIT_SECRET in 3599 seconds
Dec 3 10:37:00 chiriqui ipop3d[18186]: connect from 127.0.0.1
Dec 3 11:35:18 chiriqui Pluto[18142]:
Dec 3 11:35:18 chiriqui P1uto[18142]: | *time to handle event
Dec 3 11:35:18 chiriqui Pluto[18142]: | event EVENT_REINIT_SECRET handled
Dec 3 11:35:18 chiriqui Pluto[18142]: | inserting event EVENT_REINIT_SECRET, timeout in 3600 seconds
Dec 3 11:37:10 chiriqui ipop3d[18539]: connect from 127.0.0.1
Dec 3 15:40:30 chiriqui -- root[17534]: ROOT LOGIN ON ttyl
Dec 3 15:40:37 chiriqui PIuto[18142]:
Dec 3 15:40:37 chiriqui PIuto[18142]: | *received whack message
Dec 3 15:40:37 chiriqui Pluto[18142]: shutting down
Dec 3 15:40:37 chiriqui Pluto[18142]: forgetting secrets
Dec 3 15:40:37 chiriqui Pluto[18142]: shutting down interface ipsecO/ethO 192.168.0.2
Dec 3 20:40:02 chiriqui ipop3d[22133]: connect from 127.0.0.1
Dec 3 20:50:02 chiriqui ipopSd[22278]: connect from 127.0.0.1

##

PAM_pwdb Messages:
  Sucessful SU's:
        david -> root
        david -> root

  Successful logins:
        -> root

  Successful graphical logins:

  Authentication Failures:

  Sucessful Logins:
        root logged in 1 time(s)

Syslogd Restarted: 0 Time(s)...
Failed login(s) due to invalid username:
----- named -----
Named had 0 Malformed Response(s)...
Named had 0 Learned Response(s)...
Named loaded 0 zone(s)...
Other Named Errors:
Dec 3 06:38:51 chiriqui named[1069]: ns_forw: sendto([128.9.64.26].53): Network is unreachable

Dec 3 06:38:51 chiriqui named[1069]: ns_forw: sendto([152.158.36.48].53): Network is unreachable
Dec 3 11:50:27 chiriqui named[1069]: Lame server on 'tipwor1d.com' (in 'tipwor1d.com'?): [207.82.198.150].53 'NS2.EXODUS.NET'
Dec 3 16:37:08 chiriqui named[1069]: ns_forw: sendto([206.217.29.220].53): Network is unreachable
Dec 3 18:37:08 chiriqui named[1069]: ns_forw: sendto([206.217.29.220].53): Network is unreachable
Dec 3 20:45:16 chiriqui named[1069]: ns_forw: query(images.sourceforge.net) NS points to CNAME (nsl.sourceforge.net:)
Dec 3 20:50:47 chiriqui named[1069]: "SOURCEFORGE.NET IN NS" points to a CNAME (ns1.sourceforge.net)
Dec 3 20:51:25 chiriqui named[1069]: Lame server on 'www.elxsi.de' (in 'DE'?): [128.63.31.4].53 'ADMII.ARL.MIL'
Dec 3 21:01:54 chiriqui named[1069]: Lame server on 'ns.Germany.EU.net' (in 'eu.NET'?): [198.6.1.81].53 'AUTH01.NS.UU.net'

-------- FTPD ---------

FTP Users Logged in:
  david from volcan.pananix.com
                    Deleted 0 file(s)....
                    Transfered 5 f1le(s)....
                      /root/openssl-0.9.4.tar.gz c
                      /root/Net_SSLeay.pm-1.05.tar.gz c
                      /etc/dhcpd.conf c

Identd Lookups:
          ********** Kernel Errors **********
Dec 3 06:38:46 chiriqui kernel: PPP: ppp line discipline successfully unregistered
Dec 3 07:05:08 chiriqui kernel: CSLIP: code copyright 1989 Regents of the University of California
Dec 3 07:05:08 chiriqui kernel: PPP: version 2.3.7 (demand dialling)
Dec 3 07:05:08 chiriqui kernel: PPP line discipline registered.
Dec 3 07:05:08 chiriqui kernel: registered device pppO
Dec 3 09:05:21 chiriqui kernel: tty02 unloaded
Dec 3 09:05:22 chiriqui kernel: ethO: 3Com 3c589. io 0x300, irq 3. auto xcvr, hw_addr 00:10:5A:8B:OC:FA
Dec 3 09:05:22 chiriqui kernel: 8K FIFO split 5:3 Rx:Tx
Dec 3 09:05:34 chiriqui kernel: PPP: ppp line discipline successfully unregistered
Dec 3 10:35:18 chiriqui kernel: klips_debug:ipsec_init: ipsec module loading, freeswan version: 1.1
Dec 3 10:35:18 chiriqui kernel: klips_debug:ipsec_init: ipsec_init version: RCSID $Id: ipsec_init.c,v 1.34 1999/10/03 18:46:28 rgb Exp $
Dec 3 10:35:18 chiriqui kernel: klips_debug:ipsec_init: ipsec_tunnel version: RCSID $Id: ipsec_tunnel.c,v 1.82 1999/10/08 18:26:19 rgb Exp $
Dec 3 10:35:18 chiriqui kernel: klips_debug:ipsec_init: ipsecjietlink version: RCSID $Id: ipsec_netlink.c,v 1.35 1999/10/08 18:37:34 rgb Exp $
Dec 3 10:35:18 chiriqui kernel: klips_debug: rj_init: version: RCSID $Id: radij.c.v 1.21 1999/10/08 18:37:34 rgb Exp $
Dec 3 10:35:18 chiriqui kernel: FreeS/WAN: initialising PF_KEY domain sockets.
Dec 3 10:35:18 chiriqui kernel: klips_debug:ipsec_tunnel_init: initialisation of device: ipsec0
Dec 3 10:35:18 chiriqui kernel: klips_debug:ipsec_tunnel_init: initialisation of device: ipsecl
Dec 3 10:35:18 chiriqui kernel: klips_debug:ipsec_tunnel_init: initialisation of device: ipsec2
Dec 3 10:35:18 chiriqui kernel: klips_debug:ipsec_tunnel_init: initialisation of device: i psec3
Dec 3 10:35:18 chiriqui kernel: klips_debug:ipsec_tunnel_ioctl: tncfg service call #35312
Dec 3 10:35:18 chiriqui kernel: klips_debug:ipsec_tunnel_attach: physical device eth0 being attached has HW address: 0:10:5a:8b:0c:fa
Dec 3 10:35:18 chiriqui kernel: klips_debug:ipsec_tunnel: ipsec_tunnel_neigh_setup_dev
Dec 3 10:35:18 chiriqui kernel: klips_debug:ipsec_tunnel_open: dev = ipsecO, prv->dev =eth
Dec 3 10:35:18 chiriqui kernel: ipsec_device_event: NETDEV_UP...
Dec 3 15:39:56 chiriqui kernel: ipsec_device_event: NETDEV_DOWN...
Dec 3 15:39:56 chiriqui kernel: ipsec_device_event: NETDEV_DOWN...
Dec 3 15:39:56 chiriqui kernel: klips_debug:ipsec_tunnel_detach: physical device eth0 being detached from virtual device ipsec0
Dec 3 15:39:56 chiriqui kernel: ipsec_device_event: NETDEVJJNREGISTER...
Dec 3 15:40:38 chiriqui kernel: klips_debug:ipsec_callback: skb=0xc2387d20 skblen=48 em_magic=1400332654 em_type=8
Dec 3 15:40:38 chiriqui kernel: klips_debug:ipsec_callback: set ipsec_debug level
Dec 3 15:40:38 chiriqui kernel: klips_debug:ipsec_callback: unset
Dec 3 15:40:38 chiriqui kernel: FreeS/WAN: shutting down PF_KEY domain sockets.
Dec 3 15:40:38 chiriqui kernel: klips_debug:ipsec_cleanup: ipsec module unloaded.
Dec 3 20:37:35 chiriqui kernel: tty02 at Ox03e8 (irq = 3) is a 16550A
Dec 3 20:39:07 chiriqui kernel: CSLIP: code copyright 1989 Regents of the Universityof California
Dec 3 20:39:07 chiriqui kernel: PPP: version 2.3.7 (demand dialling)
Dec 3 20:39:07 chiriqui kernel: PPP line discipline registered.
Dec 3 20:39:07 chiriqui kernel: registered device pppO
Dec 3 21:45:00 chiriqui kernel: PPP: ppp line discipline successfully unregistered
Dec 3 21:45:05 chiriqui kernel: tty02 unloaded

          ****
****

```
All programs executed by cron (except atrun and cronloop):
06:38:46 (/usr/local/bin/webcal_remind.pl >> /var/webcal/message.log 2>&1) by root
06:38:46 (/usr/local/bin/atsal) by root
All Connections (/var/log/secure):
  Connections for in.ftpd:
     3 connections(s) by 192.168.0.1
  Connections for ipop3d:
     33 connections(s) by 127.0.0.1
File-systems...
Filesystem          1k-blocks       Used          Available      Use%     Mounted       on
/dev/hdal           4382509         3355428       800283         81%      /
/dev/hda3           1462974         1179615        207759         85%     /home

NFS Exports

/mnt/cdrom       (ro,no_root_squash)
/home       (rw,no_root_squash)
/usr        (ro,no_root_squash)
/tftpboot/volcan      (rw,no_root_squash)
/opt        (ro,no_root_squash)
/tmp        (rw,no_root_squash)
Mail Queue
Mail queue is empty
**************************************************************
Syslog      Input:                                  Output:                  90th          Msgs     User        Host
File        Msgs   Kbytes    AvgSz     Rcips   Sent    Avg Delay     Percentile          Dferd    Unkn        Unkn
summary     4657   25050     5508      4626    4557    00:00:09.22   00:00:20.00         3        30          0
################### Begin Mail  Information ###################
Total                    messages                    handled:                  4624
Total                    recipients                  handled:                  4668
Total bytes handled: 25.65M
Part I  -- Mail  relayed from:
4657 chiriqui.pananix.com
 ------------------------------------------------------------------------
Part II -- Mail sent from:
 ------------------------------------------------------------------------
4657 chiriqui.pananix.com
 ------------------------------------------------------------------------
Part III -- Mail sent to:                                          Avg delay     Max delay
-------------------------------------------------------------------------------------------------------------------------------------
4522      local host                                               8.95 sees     8.63 mins
99        chiriqui.pananix.com                                     4.74 mins     41.53 mins
1         lesbell.com.au                                           56.00 sees    56.00 sees
1         jordanheart.org                                          3.60 mins     3.60 mins
1         fftw.com                                                 11.10 mins    11.10 mins

############ End Mail information ############
check-packages run on Fri Dec 3 06:39:18 EST 1999
Listing installed packages...
619 packages installed
changes from previous run...
---
---
Checking Packages...
changes from previous run...
---
1d0
<.....U..  /dev/console
---
runtime 848 seconds
```

Дополнительные замечания

Никогда не следует играть в игры, особенно в сетевые игры, на уровне привилегий root. Зачастую игры выжимают из вашего аппаратного обеспечения все, на что оно способно. Разработчики игр больше заинтересованы в производительности, а не в безопасности, поэтому связанные с обеспечением защиты проверки (например, проверки на переполнение буфера) зачастую не выполняются, а возникающие подозрительные ситуации не обрабатываются должным образом.

Если сравнивать с серверными приложениями, между обнаружением уязвимого места в коде игры и исправлением этого зачастую проходит более длительное время. Уязвимые места в наиболее важных

службах, как правило, обнаруживаются быстрее, чем аналогичные уязвимые места в играх. К сожалению, в некоторых играх определенные функции (такие как использование расширений DG для XFree) требуют, чтобы вы запустили игру на уровне привилегий root. Возможно, для вас это не будет проблемой (например, если вы не соединены с сетью или играете в игру, которая не является сетевой), поэтому вы вполне можете пойти на риск, однако этот риск должен быть для вас известным.

После того как вы хорошенько защитили вашу систему, скорее всего, злоумышленники не будут вас беспокоить, так как в Интернете найдутся более легкие цели, чем ваша система. Если вы обнаружите, что некто пытается сканировать вашу систему, возможно, имеет смысл обратиться к вашему провайдеру, чтобы он принял меры. В результате этого в отношении злоумышленников могут быть применены санкции. Если ваши действия не дали результатов, вы можете просто блокировать пакеты, исходящие из блока IP-адресов. Если это не помогает, фильтрацию пакетов может выполнить для вас ваш интернет-провайдер. Имейте в виду, что в случае атаки типа Denial of Service каналы связи вашего провайдера будут перегружены точно так же, как и ваши серверы. Таким образом, провайдер должен быть заинтересован в оказании вам поддержки. Когда я замечаю, что мои системы подвергаются сканированию, как правило, я замечаю также, что осуществляется сканирование систем моего провайдера. И если провайдер также замечает это, в большинстве случаев он предпринимает действия для того, чтобы отключить злоумышленника от Интернета. Дело в том, что в процессе сканирования индивидуальный пользователь Интернета расходует достаточно большую емкость канала.

Заключение

В данной главе я рассказал вам о том, как быстро настроить вашу систему и сохранить на диске несколько файлов для последующих проверок состояния системы. Вы узнали о том, как выполнять проверку безопасности с использованием cron и как отсылать по почте результаты таких проверок. Большая часть материала, представленного в данной главе, наверняка не является для вас чем-то новым, однако я рассказал вам обо всем этом для того, чтобы продемонстрировать, что проверка защищенности системы выполняется относительно несложно, и поэтому не имеет смысла ею пренебрегать.

25 Средства наблюдения за сетью

В данной главе рассматриваются следующие вопросы:
- использование courtney;
- как работает courtney;
- какими возможностями и недостатками обладает courtney;
- использование nmap;
- использование графической оболочки xnmap;
- возможности и особенности xnmap.

Для того чтобы продемонстрировать вам, как именно следует организовать защиту вашей сети от вторжения, в данной главе я собираюсь рассмотреть два программных средства, связанных с безопасностью. Первым из них является программа, предназначенная для обнаружения попытки сканировать вашу систему. *Сканирование системы* — это процесс, который сигнализирует вам о том, что некто пытается получить от вашей системы нечто большее, чем вы предлагаете для всех внешних пользователей. Иными словами, этот некто пытается проникнуть к вам в систему и ищет для этого лазейки. Второе рассматриваемое здесь средство является программой, которую используют взломщики для выполнения сканирования.

Защита сети — это не только просмотр журналов, проверка функционирования брандмауэра и блокирование узлов, для которых доступ внутрь сети запрещен. Защита сети — это высокоинтеллектуальный процесс. Вы должны поставить себя на место злоумышленника, пытающегося проникнуть внутрь. Вы должны приступить к поиску ваших собственных уязвимых мест, вы должны проанализировать эти уязвимые места и принять меры, чтобы обезопасить вашу систему. Для того чтобы узнать о существовании этих уязвимых мест, вы должны искать их — иного способа обеспечить безопасность не существует.

ПРИМЕЧАНИЕ

Для обеспечения безопасности можно воспользоваться множеством хороших программ. К ним относится, в частности, Port Sentry, Snort, IPPL и IPLogger. Однако помните, что протоколирование пакетов — это только часть работы. Вторая часть — это проверка журналов. Перечисленные программы, а также многие другие полезные программные средства можно обнаружить по адресу http://freshmeat.net/.

Программа courtney

Программный пакет courtney — это средство наблюдения за сетью, написанное на Perl в лаборатории Lawrence Livermore Laboratory университета Калифорнии для Департамента энергетики Соединенных Штатов в 1995 году. Программа courtney появилась на свет почти сразу же после того, как авторы программного пакета SATAN решили предоставить свое творение для публичного использования. Программа SATAN (Systems Administrator's Tool to Assess Networks — инструмент системных администраторов для оценивания сетей) была первым программным средством, предназначенным для сканирования систем.

После того как программа SATAN стала доступной для всеобщего использования, многочисленные сетевые узлы стали подвергаться частым атакам злоумышленников. Администраторы узлов, оказавшиеся достаточно проницательными для того, чтобы приступить к просмотру журналов в поисках следов сканирования, стали замечать взаимосвязь между сканированием и попытками взлома.

ПРИМЕЧАНИЕ

Многие администраторы, включая автора этой книги, считают, что сканирование портов — это атака. Единственной целью сканирования портов является обнаружение уязвимых мест системы, а единственной целью поиска уязвимых мест системы является намерение взломать эту систему. К узлам, с которых осуществляется сканирование портов, следует проявлять особенное внимание. Адреса этих узлов следует помечать или блокировать. Сканирование портов должно осуществляться только с разрешения администратора сканируемого узла.

Чтобы запустить пакет courtney, необходимо установить в системе библиотеку libpcap и программу tcpdump. Программа tcpdump входит в стандартный комплект OpenLinux, однако библиотека libpcap там отсутствует. Программа courtney запускает tcpdump и фильтрует вывод этой утилиты при помощи libpcap, сканируя поток в поисках определенных шаблонов. Подробнее о том, как работает courtney, будет рассказано далее.

Программа tcpdump переводит вашу сетевую карту Ethernet в смешанный режим (promiscuous mode), то есть режим прослушивания сети. Такой подход является палкой о двух концах — его можно использовать как во благо, так и для причинения вреда. Благодаря смешанному режиму вы можете читать содержимое всех пакетов, передаваемых через сетевой кабель, вне зависимости от того, адресованы ли эти пакеты вашей системе или нет. Таким образом, если за пределами вашей внутренней сети, в зоне, не защищенной брандмауэром, работают web-сервер, анонимный FTP-сервер, сервер DNS и почтовый сервер, вы можете запустить courtney на любом из них (или вообще на каком-нибудь другом сервере) и наблюдать за атаками, которые направлены против любого из этих серверов.

Однако режим прослушивания сети обладает и недостатками. Карта, работающая в этом режиме, не игнорирует ни одного пакета, передающегося через сеть. Именно так работают злоумышленники, пытающиеся получить доступ к информации, которая им не предназначена. Программа courtney, так как она написана на Perl, может быть легко модифицирована таким образом, чтобы ее можно было использовать для перехвата передаваемых через сеть пользовательских имен и паролей. Из главы 12 вы знаете, что клиент FTP ставит перед пользовательским именем метку USER, а перед паролем — метку PASS, courtney можно настроить на поиск таких пакетов и пересылку их по определенному почтовому адресу. Существуют и другие неприятные обстоятельства. Операционная система Linux больше не уязвима для атак типа big ping, однако эта атака может оказаться действенной в отношении других платформ, на которых можно запустить courtney. Для одной из таких платформ достаточно принять последний пакет big ping -и в результате ядро этой ОС подвисает. Таким образом, на любое программное обеспечение, переводящее сетевую карту в прослушивающий режим, следует смотреть с подозрением. Такое программное обеспечение следует использовать только на безопасных надежных системах. Безопасной является система, к которой обычные пользователи не подключаются ежедневно. Благодаря этому пользователи не могут воспользоваться той информацией, которую извлекает из сети работающая в прослушивающем режиме сетевая карта. Брандмауэр можно считать безопасной системой, однако переключение его сетевых карт в прослушивающий режим не рекомендуется.

Запустив courtney, обратите внимание на список процессов. То, что вы увидите, будет напоминать содержимое листинга 25.1.

Листинг 25.1. Фрагмент списка процессов (ps axww), в котором показаны courtney и tcpdump

```
    pts/1      S  0:00 perl /usr/sbin/courtney
    pts/1      S  0:03 tcpdump -1 ?       (icmp[0] — 8 ) or ?       (port sunrpc) or ?
     ((port (1 or 10 or 100 or 1000 or 5000 or 10000 or 20000 or 30000) or ?(port (6000
    or 6001 or 6002 or       6010 or 6011 or 6012)) ) and ?          (tcp[13] & 18 = 2) )    or ?
(port (tcpmux or ??    echo or ??  discard or ??       systat or ??    daytime or ??    netstat
or ??    charger, or ??    ftp or ??  telnet or ??  smtp or ??    time or ??    whois or ??
domain or??    70 or ??    80 or ??       finger or ?? tftp or ??    login or ??    uucp or ??
printer or ??    shell or ??    exec  or ??    name or   ??    biff or ??    syslog or ??    talk)
and ?      (tcp[13] & 18 = 2)       ) ?
```

Таким образом, видно, что courtney фильтрует все потоки данных, которые видит tcpdump, и сравнивает их со списками стандартных служб. Наименования этих служб видны в составе листинга 25.1. Этот аспект courtney при желании можно модифицировать. Весь код этой программы написан на Perl, поэтому его легко можно изменить. Программа занимает не так уж и много места, однако из соображений экономии я покажу вам лишь наиболее интересные места. Листинг 25.2 показывает часть кода courtney, которую при желании можно модифицировать.

Листинг 25.2. Фрагмент сценария Perl под названием courtney, в котором показаны отслеживаемые службы

```
@assoc_list  =(  'sunrpc',     'icmp',     'ttime1,     'telnet',     'smtp',
                 'ftp',        'whois',    'domain',    'gopher',     'www',
                 'finger',     'exec',     'login',     'shell',      'printer',
                 'uucp',       'tcpmux',   'echo',      'di scard',   'systat'.
                 'daytime',    'netstat',  'chargen',   'tftp',       'name',
                 'biff',       'syslog',   'talk',      'portscan',   'xwindows'  );
```

Если какая-либо из перечисленных здесь служб не упомянута в файле /etc/ services, сценарий courtney завершает работу, выведя на экран сообщение об ошибке. Если по какой-либо причине вы удаляете запись о службе из файла /etc/ services, вы либо должны удалить имя этой службы из массива assoc_list, либо вместо имени указать номер порта для данной службы.

Чтобы обнаружить попытку сканирования, программа Courtney сравнивает адреса-источники узлов, подключающихся к различным портам. Для этого Courtney создает список узлов, пытающихся подключаться к портам. Если за определенный промежуток времени осуществляется слишком большое количество попыток соединения со слишком большим количеством разных портов, Courtney делает вывод о том, что система находится под атакой.

По каким признакам Courtney определяет, что количество соединений слишком большое? Для этого используются два фактора: количество портов, к которым подключается один и тот же узел, и количество портов, подключение к которым осуществлено за некоторый период времени. Эти два фактора показаны в листинге 25.3.

Листинг 25.3. Параметры, при помощи которых courtney обнаруживает атаку

```
$UDATE_INTERVAL =5:  # обновлять информацию о сетевой узле
                     #каждые X минут
$OLD_AGE =7;         # стирать записи об узлах,
                     #если эти записи старше X минут.
$HIGH_THRESHOLD=15; # тяжелая атака    "SATAN"
$LOW_THRESHOLD =9;  # нормальная атака "SATAN"
```

По умолчанию courtney пытается обнаружить узел (адрес-источник), который подключается к более чем девяти службам в течение менее чем семи минут или к более чем 15-ти службам в течение менее чем семи минут. Эти узлы идентифицируются сценарием как узлы, осуществляющие нормальную или тяжелую атаку соответственно.

В системе Linux программа courtney распознает три различных типа сканирования: connect, SYN stealth и FIN stealth. Некоторые из операционных систем не реагируют на эти виды сканирования и даже не видят их (в этом случае courtney не будет их регистрировать). Это связано со способом, который используется ядром для обработки принимаемых пакетов. В системе Linux, в которой используется ядро версии 2.2.x, ядро реагирует на все эти виды сканирования, поэтому courtney будет регистрировать все подобные попытки. Более подробное описание этих типов сканирования содержится в разделе, посвященном программе nmap.

Теперь, когда вы знаете, по каким признакам программа courtney определяет факт нападения на вашу систему, вы можете представить себе, в каких ситуациях courtney не сможет обнаружить действия злоумышленников. Иными словами, вы можете представить себе уязвимые места courtney. Прежде всего, сценарий courtney чувствителен к скорости сканирования. Если сканирование осуществляется в течение нескольких часов (как это бывает при выполнении сканирования одним из способов stealth), более старые записи будут удаляться из таблицы соединений и таким образом, не будут вызывать «тревогу». Например, если атакующий будет проверять шесть портов каждые 15 минут, он сможет без труда избежать обнаружения. Программа courtney не сможет обнаружить злоумышленника также в случае, если он осуществит «сканирование» только одного порта.

Второй недостаток courtney связан с тем, что злоумышленник может изменить адрес-источник отправляемых им пакетов (или использовать для сканирования ftp proxy). В этом случае courtney сообщит вам о попытке сканирования, однако источник атаки будет указан неправильно. Иными словами, вы будете смотреть в одну сторону, в то время как атака будет исходить с другой стороны. Скорее всего, вы блокируете доступ из сети evil.net, в то время как на самом деле сканирование осуществляется из сети archevil.net.

Можно ли преодолеть эти ограничения courtney? Определенно, вы можете менять значения любых присутствующих в исходном коде этой программы переменных, однако при этом вы должны хорошо понимать, к чему это может привести.

Начнем с переменной времени. Наверное, это наиболее безобидная переменная, значение которой вы можете изменить. Как правило, при достаточно быстром сканировании для проверки всех хорошо известных служб достаточно всего нескольких секунд. Несмотря на это, сокращать семиминутный интервал не рекомендуется. Вы можете попробовать увеличить его, однако при этом следует быть осторожным — courtney хранит массивы записей «источник:порт», «прием-ник:порт» и время в оперативной памяти, поэтому при достаточно большой нагрузке на сеть память может быстро заполниться. Не стоит забывать о том, что courtney просматривает весь трафик, передаваемый через сеть, а не только трафик, адресованный локальному узлу. Если вы запускаете courtney в сети, через которую передается большой объем данных, даже если эти данные не адресованы подсети, в которой работает courtney, эта программа все равно будет осуществлять их анализ. Возможно, при помощи courtney вы сможете обнаружить атаки, исходящие из одной удаленной сети и направленные в другую удаленную сеть. Это может произойти в случае, если связанные с этим пакеты данных передаются через каналы вашей подсети (или переданы в вашу подсеть из-за неправильной настройки маршрутизатора Интернета, к которому вы подключены). Если вы увеличите продолжительность промежутка

времени, возможно, это не даст вам ничего, за исключением излишнего расхода памяти.

Помимо переменной, определяющей промежуток времени, вы можете изменить значения переменных, определяющих количество служб, к которым пытается подключиться некоторый удаленный узел. Вы можете присвоить переменной LOW_THRESHOLD меньшее значение, однако в этом случае courtney может поднять ложную тревогу. Представьте, например, что некто, находящийся вдали от дома, подключается к локальному провайдеру и затем пытается соединиться с сервером, работающим в сети своего офиса. Этот пользователь может копировать файлы через FTP в обоих направлениях, использовать POP для получения почты, его web-браузер по умолчанию может сразу же после запуска подключиться к домашней страничке офисного web-сервера, возможно, пользователь воспользуется telnet или ssh для того, чтобы соединиться с офисной сетью. Таким образом происходит подключение к пяти различным сетевым службам. Скорее всего, все эти подключения будут выполнены в течение менее чем пяти минут. Если при этом вы уменьшите порог обнаружения попытки сканирования, программа courtney сообщит вам, что данный пользователь пытается сканировать вашу систему, в то время как на самом деле ничего подобного не происходит. Конечно же, вы можете помимо снижения порогов обнаружения сканирования также удалить некоторые из служб из массива @assoc_list. Если вы хотите наблюдать лишь за одним или двумя портами, к которым не должен подключаться ни один из внешних пользователей, для этой цели лучше использовать tripwire. Если внимательно изучить содержимое массива @assoc_list, можно заметить, что в этом массиве не упоминаются некоторые из стандартных служб, например, POP2, POP3, ШАР и SSH. Возможно, будет лучше, если вы добавите эти службы в массив, особенно с учетом того, что служба ШАР является чрезвычайно привлекательной целью для взломщиков. Если вы добавляете в массив службу SSH, вы должны добавить ее также и в файл /etc/services вашей системы.

При запуске courtney можно использовать параметры командной строки. Наиболее часто используемым параметром является команда, предписывающая Courtney пересылать оповещения об обнаруженных атаках на некоторый почтовый адрес (обычно root). Именно так по умолчанию настраивается сценарий init К этим параметрам относятся:

-i имя_интерфейса — интерфейс, через который осуществляется прослушивание (если не указан, прослушивание осуществляется через интерфейс по умолчанию);

- -d — включение отладочного режима, то есть режима с выводом дополнительной диагностической информации;

-l — отключение протоколирования syslog (протоколируемые данные пересылаются пользователю);

-s — включение вывода информации на экран;) -c — отображение сведений о соединениях;

-m адрес — оповещения об обнаруженных атаках пересылаются по электронной почте на указанный адрес;

-h — электронная справка.

Программа courtney зарекомендовала себя как чрезвычайно полезный инструмент, помогающий системным администраторам обнаруживать сетевые атаки. Программа courtney, в частности, позволяет обнаруживать попытки сканирования портов, производимые с использованием таких программных средств, как nmap, NESSUS, SAINT, GtkPortScan, а также многих других средств сканирования, распространяемых в наши дни через Интернет. Однако при использовании courtney вы должны помнить о том, что эту программу можно обмануть. Используйте ее, если вы того хотите, но при этом не забывайте о ее возможностях и ограничениях.

Программа nmap

Первое публично доступное средство сканирования сетей — программа с названием SATAN — было очень популярно и чрезвычайно эффективно при идентификации открытых портов на подключенных к сети системах. Сегодня программа SATAN утратила былую популярность по ряду причин, среди которых сложности при компиляции в среде Linux, а также проблемы получения работающих исполняемых файлов. На смену SATAN пришли многочисленные аналогичные программные средства, так же как и SATAN, осуществляющие сканирование сетевых систем. Одной из наиболее популярных программ этой категории является программа с именем nmap, написанная человеком по прозвищу Fyodor (fyodor@dhp.com). Эта программа позволяет осуществлять сканирование несколькими разными способами, кроме того, пользователи, у которых возникли вопросы, для решения проблем могут воспользоваться списком рассылки. Web-узел, посвященный программе nmap, располагается по адресу http://www.insecure.org/nmap/.

В данной книге я не собираюсь обсуждать преимущества nmap и сравнивать эту программу с другими популярными в настоящее время средствами сканирования, такими как SAINT (дальнейшее

развитие SATAN), NESSUS, GtkPortScanner и проч. Большинство подобных программ обладают сходными возможностями и выполняют одну и ту же базовую функцию: сканирование сетевых узлов в поисках открытых портов. Чтобы сформировать свое собственное мнение об этих программах, вы можете попробовать каждую из них самостоятельно. Программа nmap является рядовым представителем этой категории программных средств, я выбрал ее из-за того, что эта программа является достаточно простой в использовании.

ВНИМАНИЕ

Программа nmap в состоянии выполнять действия, которые могут привести к деградации производительности сети (Denial of Service) и нарушению работы жизненно важных серверов вашей сети. По этой причине программу nmap следует использовать с осторожностью.

Программу nmap можно запускать от лица любого пользователя системы. Если вы установили nmap, но не хотите, чтобы ею мог воспользоваться любой желающий, вы должны настроить разрешения на доступ таким образом, чтобы ограничить доступ к этой программе. Следует учитывать, что некоторые из возможностей программы доступны только в случае, если она запущена от лица пользователя root. Это связано с тем, что программа выполняет запись в «чистый» сокет (raw socket) или формирует свой собственный сетевой пакет. В общем случае в среде Linux подобные действия разрешается выполнять только привилегированным пользователям.

В комплекте nmap присутствует графическая оболочка этой программы с названием xnmap (разработанная также человеком по прозвищу Foydor), которую могут использовать те, кто не любит работать с командной строкой. Приятной особенностью этой графической оболочки является отображение командной строки nmap, которая будет использована в процессе работы. Автор программы эффективно использует цветовое оформление для того, чтобы отобразить статус пользователя: фоновый цвет диалогового окна становится розовым в случае, если программа запускается на уровне привилегий root, при этом в текстовом окне отображается сообщение о том, что доступны все возможности программы (all options granted); если же программа запускается от лица непривилегированного пользователя, фоновый цвет диалогового окна становится зеленым, а в текстовом окошке выводится сообщение о том, что некоторые из возможностей программы недоступны. Графический интерфейс программы nmap показан на рис. 25.1.

Диалоговое окно xnmap разделено на четыре части: раздел идентификации сетевого узла (Hosts), раздел параметров сканирования (Scan Options), раздел общих параметров (General Options) и раздел текстового вывода (Output). В строке адреса сканируемого узла Host(s) по умолчанию указывается адрес локального сетевого узла 127.0.0.1. В этой строке можно указать список сетевых узлов, идентифицировать которые можно при помощи имени, IP-адреса или диапазона IP-адресов. Диапазон IP-адресов указывается в виде номера сети и сетевой маски, например 123.34.67.00/24. Любое число можно заменить специальным символом звездочки (*), например 123.*.67.00/24. При этом будет выполнено сканирование всех сетей в IP-домене 123, однако в каждой из этих сетей будет сканироваться только подсеть 67. Диапазон адресов можно указывать также в виде начального и конечного значения, например 1-33. Такую форму можно указывать вместо любого из октетов IP-адреса, например 123.1-33.67.00/24. Наконец, интересующие вас численные значения можно указать через запятую, например 123.1,3,7,10.76.00/24. Имейте в виду, что в зависимости от используемой вами командной оболочки вы должны будете для указания символа звездочки использовать esc-последовательность. Вы также можете указать имя сетевого узла и маску.

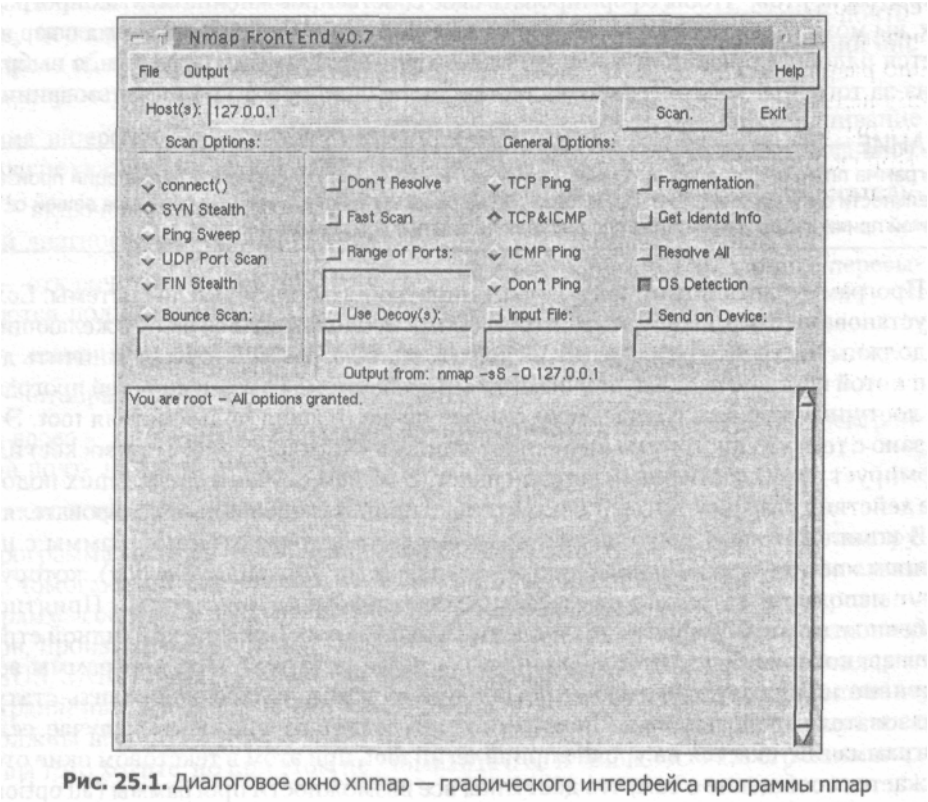

Рис. 25.1. Диалоговое окно xnmap — графического интерфейса программы nmap

Указав в графе Host(s) интересующие вас сетевые узлы, вы можете щелкнуть на кнопке Scan для того, чтобы инициировать сканирование. Помните, что в некоторых случаях сканирование может потребовать достаточно много времени. Если вы хотите остановить процесс сканирования, достаточно будет еще раз щелкнуть на кнопке Scan.

Параметры сканирования nmap

Программа nmap позволяет осуществлять сканирование девятью разными способами. Шесть из них перечислены в разделе Scan Options диалогового окна xnmap. Вариант connect() обеспечивает сканирование самым обычным способом. В этом случае программа nmap подключается к целевому узлу так, как будто для этого используется клиент telnet. При этом формируется соединение TCP, поэтому такое сканирование является сканированием TCP. Попытку сканирования, осуществляемую подобным образом, можно обнаружить даже самыми примитивными средствами обнаружения сканирования.

Способ сканирования SYN stealth работает несколько иначе. Вспомним последовательность создания соединения TCP: клиент посылает пакет по адресу узел: порт. В этом пакете установлен бит SYN. Целевой узел в ответ посылает клиенту пакет, в котором установлены биты SYN и ACK. Клиент отвечает на этот пакет, и процедура создания соединения успешно завершается. Однако в нашем случае вместо того, чтобы ответить на пакет SYN-ACK, программа nmap отправляет целевому узлу пакет с установленным битом RST, при этом соединение немедленно разрывается. Если соединение не было создано, значит, попытку сканирования нельзя обнаружить. Почти. На самом деле подобные попытки сканирования могут быть обнаружены с использованием утилит courtney и Synlogger.

При сканировании способом Ping Sweep программа nmap отправляет целевому узлу запрос ping и ожидает поступления ответа. Если целевая система блокирует запросы ping при помощи IPChains или NetFilter, программа nmap думает, что целевой узел не работает или отсутствует в сети. Следует учесть, что некоторые из методов сканирования, поддерживаемые программой nmap, предусматривают предварительную отправку целевому узлу запроса ping для того, чтобы убедиться, что целевой узел находится в рабочем состоянии. То есть в начале сканирования при помощи запросов ping программа nmap проверяет работоспособность узла и только после этого отправляет ему сканирующие запросы TCP. Как несложно догадаться, если запросы ping блокируются целевой системой, целевой узел не будет отвечать на запросы ping и программа nmap не будет выполнять его сканирование. Позже я расскажу о том, как можно сканировать узлы, блокирующие ping-запросы.

Способ сканирования UDP Port Scan работает несколько иначе. Вместо того чтобы пытаться установить соединение TCP, программа nmap пытается проверить наличие открытого порта UDP, для этого в этот порт посылается пакет UDP нулевой длины. Если сервер ожидает поступления пакетов через этот порт, он примет пакет, и ничего больше не произойдет. Если же сервер не ожидает поступления каких-либо данных через этот порт, целевой узел отправляет клиенту пакет RST (reset). Этот пакет сообщает клиенту, что необходимо разорвать соединение, так как данный порт не является открытым.

При использовании метода FIN Stealth программа nmap формирует специальный пакет, который в нормальных условиях используется клиентом для того, чтобы сигнализировать о разрыве соединения TCP. Помните состояния TCP-соединения, отображаемые утилитой netstat? Передача или получение пакета FIN переключает соединение из состояния FIN_WAIT в состояние FIN_WAIT1 или FIN_WAIT2. Пакет FIN обычно посылается при завершении TCP-соединения для того, чтобы сигнализировать о его разрыве. Но если на момент приема этого пакета никакого соединения не существует? Если сервер ожидает поступления пакетов через данный порт, тогда он просто отбрасывает данный пакет, предполагая, что пакет был неправильно передан или произошла какая-либо другая ошибка. Однако если данный порт закрыт (в системе отсутствует серверная программа, ожидающая поступления данных через этот порт), целевая система отправит клиенту пакет RST.

Последний вариант сканирования, предлагаемый диалоговым окном xnmap, называется Bounce Scan (сканирование с отражением). Этот способ позволяет использовать дыру (и очень большую дыру) в системе безопасности FTP-серверов, используя их в качестве proxy. Таким образом, данный вариант было бы точнее назвать FTP Proxy Scan. К счастью, FTP-сервер, входящий в комплект поставки Caldera OpenLinux, блокирует захват привилегированных портов. Однако если вас мучает любопытство, вы можете попробовать сканировать адрес 127.0.0.1/24 и указать в графе адреса Bounce Scan адрес наподобие 127.1.1.1.

Из шести вариантов сканирования, предлагаемых в диалоговом окне xnmap, только два — connect() и Ping Sweep — могут быть выполнены обычными, непривилегированными пользователями.

Помимо шести способов сканирования, предлагаемых xnmap, существуют еще три, которые можно инициировать из командной строки nmap. Два из них являются разновидностями сканирования FIN stealth. Первый из этих двух способов называется Christmas Tree Scan (сканирование рождественской елки). Он подразумевает, что помимо бита FIN в сканирующем пакете устанавливаются еще два бита: URG и PUSH. Второй способ называется Null Scan — в сканирующем пакете сбрасываются все биты.

Последним методом сканирования, поддерживаемым программой nmap, является RPC Scan — специализированный способ сканирования, позволяющий искать открытые порты RPC. Этот метод можно использовать совместно с другими способами сканирования.

Общие параметры nmap

При помощи раздела General Options диалогового окна xnmap вы можете влиять на поведение программы nmap. Если не учитывать пару исключений, программу nmap можно запустить, не выбирая ни одного из параметров группы General Options. Параметры этой группы позволяют вам модифицировать порядок работы nmap, используемый этой программой по умолчанию. Параметр Don't Resolve (не осуществлять разрешение) предотвращает разрешение имен DNS для сканируемых узлов — это несколько ускоряет процесс сканирования, однако совсем незначительно.

При использовании параметра Fast Scan (быстрое сканирование) для сканирования используется файл служб с именем ../lib/nmap/nmap-services. В этом файле перечисляются 1975 портов TCP и UDP (1022 портов TCP и 953 порта UDP). Благодаря этому сканирование выполняется значительно быстрее, чем при проверке всех портов из диапазона 64 Кбайт. Параметр Range of Ports (диапазон портов) аналогичен параметру Fast Scan, однако при его использовании вы можете самостоятельно определить, какие именно диапазоны портов вас интересуют.

Параметр Use Decoy(s) позволяет выполнять перезапись адреса-источника. Если вы используете этот параметр, у сканируемого вами узла возникает ощущение, что он подвергается атаке одновременно с нескольких узлов. В графе Use Decoy(s) указывается перечень разделенных запятыми сетевых узлов. Сетевые узлы можно идентифицировать как при помощи IP-адреса, так и при помощи имени. В качестве одной из позиций в списке можно указать английское местоимение ME — вместо него программа nmap подставит ваш собственный узел. Если местоимение ME в списке отсутствует, программа nmap располагает ваш узел в случайной позиции списка.

ВНИМАНИЕ

Все сетевые узлы, указываемые вами в списке Use Decoy(s), должны реально существовать и функционировать в сети, в противном случае в результате сканирования в подобном режиме вы станете причиной атаки SYN DoS в отношении сканируемого вами узла. Иными словами, сканируемый вами узел окажется «затопленным» многочисленными SYN-пакетами. Кроме того, использование несуществующих IP-адресов будет способствовать обнаружению реального IP-адреса вашего узла.

К счастью, ядро OpenLinux можно скомпоновать так, чтобы пакеты с подделанным обратным адресом отбрасывались.

Параметры, перечисленные в следующей колонке, взаимосвязаны между собой, (за исключением последнего). Каждый из этих параметров указывает на тип ping-пакета, который используется программой

nmap. Пакеты ICMP являются стандартными «реальными» пакетами ping. При использовании ICMP через сеть отправляется эхо-запрос ICMP и программа ожидает, что в ответ на этот запрос целевой узел вернет эхо-ответ ICMP. Вы можете отключить использование ping-пакетов ICMP или использовать их совместно с ping-пакетами TCP. На самом деле протокол TCP официально не поддерживает стандартного механизма проверки связи, аналогичного механизму ICMP ping. Однако в качестве замены ICMP ping можно использовать одну из особенностей протокола TCP. Дело в том, что при получении TCP-запроса, адресованного закрытому порту, система отправляет обратно клиенту пакет RST в знак того, что интересующий этого клиента порт на самом деле закрыт. Таким образом, если программа nmap отправляет целевому узлу TCP-запрос на подключение и если на этот запрос не получено ответа, это может означать одно из двух: либо порт открыт, либо целевой узел не существует. Однако получение в ответ на запрос пакета RST — это то же самое, что получение от целевого узла эхо-ответа ICMP.

Последним параметром в этой колонке является графа Input File, в которой можно указать имя файла, в котором содержится список сканируемых узлов. Таким образом, вместо того чтобы перечислять узлы в графе Host(s) диалогового окна xnmap или в командной строке утилиты nmap, вы можете внести их в файл. Эта возможность может оказаться полезной в случае, если вы регулярно выполняете сканирование вашей внутренней сети, однако при этом вас интересуют только несколько избранных систем.

В последней колонке раздела General Options диалогового окна xnmap содержатся другие параметры командной строки. Первым параметром является параметр Fragmentation (фрагментация). Этот параметр предписывает программе nmap намеренно выполнять фрагментацию заголовков пакетов в надежде, что в этом случае они смогут быть переданы целевому узлу. Большинство брандмауэров по умолчанию пропускают фрагменты, если только им не предписано обратное. Ядро Linux можно откомпилировать таким образом, чтобы не пропускать фрагменты (параметр CONFIG_IP_ALWAYS_DEFRAG), и этот параметр должен всегда использоваться в системе, которая выполняет функции брандмауэра. Имейте в виду, что данный параметр в ядрах серии 2.4.x больше не используется, так как дефрагментация является встроенным в ядро режимом работы по умолчанию.

Параметр Get Identd Info используется для получения дополнительной информации о системах, на которых запущен демон identd. Если на целевом узле работает этот демон, при помощи данного параметра вы можете получить имя пользователя, от лица которого работает некоторая сканируемая служба. Пользователь может быть как любым непривилегированным пользователем, так и пользователем root. Ранее я уже неоднократно указывал вам на необходимость отключения демона inetd в файле /etc/inetconf, сейчас вы получили еще один аргумент в пользу отключения этого демона.

Параметр Resolve All является параметром, обратным параметру Don't Resolve, однако программа nmap его больше не поддерживает, так как она работает в этом режиме по умолчанию.

Параметр OS Detection (определение операционной системы) предписывает программе nmap попытаться идентифицировать операционную систему, работающую на целевом узле. Для этой цели используется отпечаток стека TCP/IP (TCP/IP fingerprinting). Отпечаток стека — это набор тестовых запросов к целевому узлу и ответов на эти запросы. На основании этой информации программа nmap пытается идентифицировать ОС, работающую на целевом узле. Такой метод нельзя считать на 100 % надежным. Если вы попробуете использовать его для идентификации ОС локального узла и при этом воспользуетесь адресом 127.0.0.1/24, вы получите несколько ответов, на основании которых nmap не сможет идентифицировать ОС вашего узла. Возможно, для корректного определения ОС вам потребуется повторить эту процедуру два или три раза.

Параметр Send on Device (отослать через устройство) позволяет указать сетевую карту, через которую будут отсылаться в сеть сканирующие пакеты. Этот параметр предназначен для компьютеров, оснащенных несколькими NIC.

Некоторые любопытные параметры доступны также из командной строки nmap. Здесь я не буду рассматривать все поддерживаемые параметры, а остановлюсь лишь на самых интересных. Первым таким параметром является параметр -S <IP-адрес>. Этот параметр позволяет вам использовать в качестве адреса-источника вместо адреса вашего узла указанный в параметре IP-адрес. Параметр можно использовать как для обмана целевого узла, так и в случае, если nmap не может самостоятельно определить ваш собственный IP-адрес и вы намерены указать его вручную. В отличие от рассмотренной ранее возможности Use Decoy(s) параметр -S создает у целевого узла ощущение, что сканирование осуществляется только с одного узла, однако адрес этого узла вы задаете самостоятельно, и он может быть ложным.

Еще одним весьма интересным параметром является параметр -g <номер_порта>. Этот параметр позволяет подделать номер порта, из которого исходят сканирующие пакеты, подобно тому как параметр -S подделывает IP-адрес сканирующего узла. Подделав номер порта-источника, вы можете преодолеть защиту некоторых из брандмауэров.

Последним из рассматриваемых здесь параметров является параметр -T, который позволяет настроить временной режим сканирования. При помощи этого параметра вы можете установить уровень сканирования от 0 до 5. Уровни сканирования обозначаются следующими именами: Paranoid

(параноидальный), Sneaky (трусливый), Polite (обходительный), Normal (нормальный), Aggressive (аргессивный) и Insane (бешеный). Сканирование в параноидальном режиме выполняется чрезвычайно медленно и позволяет избежать обнаружения такими программами, как Courtney. В режиме Insane сканирование выполняется параллельно и настолько быстро, насколько это возможно. При этом пропускная способность самого медленного канала связи между источником и целью полностью исчерпывается. Очевидно, что сканирование в таком режиме привлечет к себе внимание администраторов сканируемой системы.

Вывод программы nmap

Непосредственно над окном вывода располагается командная строка, которая будет выполнена в момент щелчка на кнопке Scan в диалоговом окне xnmap. Благодаря этому пользователь графической оболочки xnmap получает возможность освоить работу с nmap из командной строки. Однако полную информацию обо всех параметрах командной строки, последних изменениях и способах сканирования можно получить только из электронной документации man pages. Благодаря использованию nmap вы получаете представление о том, как осуществляется сканирование ваших узлов, и можете проверить, позволяет ли courtney или другая программа обнаружения попыток сканирования зафиксировать сканирование или ее можно обмануть.

В листинге 25.4 показан вывод программы nmap при выполнении сканирования всех портов локального узла с использованием метода SYN Stealth. Такое сканирование в отношении локального узла выполняется существенно быстрее, чем сканирование локальной сети Ethernet.

Листинг 25.4. Программа nmap сканирует узел localhost

```
Starting nmap V.  2.3BETA6 by Fyodor (fyodor@dhp.com,  www.insecure.org/nmap/)
Interesting ports on localhost (127.0.0.1):
Port        State       Protocol    Service
21          open         tcp        ftp
22          open        tcp         ssh
25          open        tcp         smtp
80          open        tcp          http
110         open        tcp         pop-3
111         open        tcp         sunrpc
901         open        tcp         unknown
3306        open        tcp         mysql
6000        open        tcp         Xll
TCP Sequence Prediction: Class=random positive increments
                         Difficulty=599840 (Good luck!)
Remote operating system guess: Linux 2.1.122 - 2.2.12
Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

Время, которое требуется программе nmap для того, чтобы отобразить результаты сканирования на экране, определяется пропускной способностью каналов, количеством сканируемых узлов и тем, используете ли вы режим Fast Scan или нет. Во многих случаях крупные сети эффективнее сканировать из командной строки и запускать процесс сканирования в фоновом режиме.

Вывод программы nmap, показанный в листинге 25.4, вполне соответствует набору служб, предлагаемых целевой системой. Обратите внимание на графу TCP Sequence Prediction (предсказание последовательности TCP). В этой графе указывается, насколько сложно будет имитировать подключение TCP. Для Linux значение этого параметра, как правило, лежит в диапазоне от 500 000 до 2 500 000. Чем больше значение этого параметра, тем выше расхождение. У операционной системы Sun Solaris значение этого параметра обычно еще выше, чем у Linux. Меньшие значения соответствуют меньшему расхождению. В листинге 25.5 показаны примеры значений этого параметра для одного и того же сетевого узла Linux.

Листинг 25.5. Значение параметра TCP Prediction Sequence для узла localhost с 30-секундным интервалом

```
TCP Sequence Prediction:  Class=random positive increments
                          Difficulty=2177514 (Good luck!)
Remote operating system guess:  Linux 2.1.122 - 2.2.12
TCP Sequence Prediction:  Class=random positive increments
                          Difficulty=4453284 (Good luck!)
Remote operating system guess:  Linux 2.1.122 - 2.2.12
TCP Sequence Prediction: Class=random positive increments
                          Difficulty=871237 (Good luck!)
Remote operating system guess:  Linux 2.1,122 - 2.2.12
```

По сравнению с графической оболочкой xnmap командная строка nmap наделяет вас более широкими возможностями. Работая с командной строкой nmap, вы можете указать большее количество параметров в самых разнообразных комбинациях. Если в командной строке вы указываете несовместимую комбинацию

параметров, программа nmap сообщит вам об этом, однако при этом все же попытается выполнить сканирование. В листинге 25.6 показан пример запуска nmap из командной строки.

Листинг 25.6. Программа nmap запускается из командной строки с использованием комбинации параметров, недоступной из графической оболочки xnmap

```
[root@chiriqui /root]# nmap -sFUR local host
Starting nmap V.  2.3BETA6 by Fyodor (fyodorPdhp.com,  www.insecure.org/nmap/)
Interesting ports on localhost (127.0.0.1):
Port            State       Protocol  Service (RPC)
21              open         tcp       ftp
22              open        tcp       ssh
25              open        tcp       smtp
67              open        udp       bootps
80              open        tcp       http
110             open        tcp       pop-3
111             open        udp       sunrpc
111             open        tcp       sunrpc
177             open        udp       xdmcp
514             open        udp       syslog
901             open        tcp       unknown
3306            open        tcp       mysql
6000             open        tcp       X11
Nmap run completed -- 1 IP address (1 host up) scanned in 3 seconds
```

В листинге 25.6 присутствует информация о нескольких портах, о которых не сообщалось ранее, при использовании xnmap. Речь идет о портах UDP — при использовании xnmap вы должны сканировать UDP-порты отдельно. В листинге показан перечень всех портов, как TCP, так и UDP. Обратите внимание, что порт с номером 901 является портом swat (Samba Web Administration Tool). Точно такое же сканирование (все 64 Кбайт портов в режиме TCP/UDP/RCP) в отношении другого узла, подключенного к тому же сегменту Ethernet, потребовало бы

1586 секунд, или более 26 минут.

-

Сравнение nmap и netstat

У многих может возникнуть вопрос: зачем сканировать собственные узлы с использованием nmap, когда ту же самую информацию можно получить при помощи netstat? Команда netstat -an показывает вам все открытые порты локального узла. Отличие состоит в том, что, используя nmap (особенно если эта программа запускается с другого сетевого узла), вы видите то, что видят злоумышленники и взломщики, пытающиеся нащупать уязвимые места вашей системы. Эти сведения могут существенно отличаться от того, что отображает на экране утилита netstat.

Воспользовавшись nmap, вы сможете воочию убедиться в том, насколько эффективным является набор правил вашего брандмауэра, корректно ли выполняется блокирование эхо-запросов ICMP, какие порты действительно заблокированы и о каких портах вы позабыли. Вывод nmap выглядит менее громоздко, чем вывод netstat, его легче читать.

Для более тщательной проверки вашей системы защиты вы можете запускать nmap с локального узла localhost, из внутренней сети и из внешней сети, а затем сравнивать результаты. Сканирование nmap необходимо выполнять регулярно, частота сканирований определяется вашей конкретной ситуацией. Чем выше уровень уязвимости вашей сети и чем больше входящий трафик, тем более часто следует сканировать вашу сеть. Это можно выполнять ежемесячно, еженедельно или даже чаще. Однако выполняя сканирование, следует быть осторожным. Лучше выполнять сканирование медленно, в нерабочее время, возможно на протяжении нескольких вечеров — такой подход более безопасен, чем скоростное сканирование в рабочее время. При скоростном сканировании nmap может создать существенную нагрузку на каналы связи. Программа nmap автоматически замедляет процесс сканирования в случае, если удаленный сервер замедляет отправку ответов на адресуемые ему запросы. Это нормально и ожидаемо. Операционная система Linux на протяжении четырех минут отправляет в сеть не более 80 ответов ICMP Port Unreachable (порт недоступен). Система Sun Solaris отправляет в сеть еще меньшее количество ответов. Однако узлы Microsoft вообще никогда не замедляют свою работу, то есть отправляют в сеть ответы ICMP Port Unreachable настолько быстро, насколько это возможно, а это означает, что, запустив nmap в отношении узла Microsoft, вы можете нарушить работу сервера, сократить доступную пропускную способность каналов (это будет выглядеть как избыток коллизий Ethernet) или просто создать такую нагрузку на сеть, что пользователи сразу же почувствуют неудобства.

Заключение

В данной главе были рассмотрены две программы: одна из них осуществляет обнаружение попыток сканирования системы, а вторая осуществляет собственно сканирование. Вы узнали о том, какие методы используются для обнаружения сканирования и как ведет себя сканирующее программное обеспечение, чтобы получить сведения о сканируемой системе и при этом обмануть программы, следящие за попытками сканирования.

В начале главы я рассказал вам о courtney — программе, осуществляющей обнаружение попыток сканирования. Мы рассмотрели переменные, которые влияют на работу этой программы, и изучили, каким образом значения этих переменных могут быть модифицированы. Я рассказал вам о недостатках courtney и о том, как можно обмануть эту программу.

После этого я рассказал вам о программе nmap — наиболее популярном инструменте сканирования портов. Вы увидели, что сканирование в режиме по умолчанию может быть обнаружено при помощи courtney. Также было рассказано о параметрах работы этой программы, а также о том, как настроить nmap таким образом, чтобы попытка сканирования не была обнаружена при помощи courtney.

26 Где найти сведения о безопасности

В данной главе рассматриваются следующие вопросы:
- где найти сведения, связанные с безопасностью;
- официальные узлы, посвященные безопасности;
- узлы сомнительной направленности.

Что касается информации о безопасности и защите данных, то Интернет можно назвать палкой о двух концах — извлекаемые из него сведения можно использовать как во благо, так и во вред. В данной книге я постарался предоставить читателям базовую информацию о защите систем, ориентированную на начинающих администраторов и администраторов среднего уровня подготовки. Более подробные сведения, связанные с безопасностью, можно почерпнуть из специальных книг, которые, как правило, посвящаются рассмотрению какой-то конкретной более узкой области. В подобных книгах, как правило, рассматриваются теоретические основы, функционирование и практическое использование тех или иных аспектов безопасности, причем уровень детализации материала может оказаться чрезмерным очень для многих, за исключением, может быть, лишь наиболее опытных экспертов в области компьютерной безопасности. И даже самые опытные эксперты подчас находят этот материал утомительным и обескураживающим.

Хочу особо подчеркнуть, что безопасность вашей системы не ограничивается сведениями, содержащимися в данной книге. В Интернете постоянно появляются сообщения о новых способах взлома систем и нарушения систем защиты. Вы должны тщательно следить за новостями в этой области. Например, пока я писал эту книгу, я узнал о новом методе ping-сканирования узлов, игнорирующих по-, ступающие к ним ping-запросы. Представьте, что некоторый сканируемый узел блокирует эхо-запросы ICMP. Ранее я уже рассказывал вам о том, что для проверки существования этого узла в сети можно использовать сообщения SYN-ACK. Если в ответ на такое сообщение вы получаете пакет RST, значит, узел функционирует, однако соответствующий порт закрыт. Если же в ответ на сообщение SYN-ACK вы ничего не получаете, значит, либо узла с таким адресом не существует, либо порт, в который вы направляете сообщение, находится в открытом состоянии. Но что, если узел отбрасывает абсолютно все посылаемые ему пакеты, делая вид, будто он не существует в сети? Иными словами, представьте, что узел не предлагает внешним пользователям никаких служб, а также отбрасывает все эхо-запросы ICMP и пакеты TCP, которые не являются ответами на отправляемые этим узлом запросы на соединение. У внешних пользователей, пытающихся определить присутствие такого узла, может возникнуть впечатление, что узел отсутствует. Однако какой-то хитроумный компьютерный корифей придумал посылать такому узлу не эхо-запросы, а эхо-ответы ICMP. Конечно, сканируемый узел может блокировать (то есть отбрасывать) эхо-запросы ICMP, однако эхо-ответы, скорее всего, блокироваться не будут, так как в этом случае сама сканируемая система не сможет воспользоваться механизмом ping для тестирования связи с удаленными узлами. На самом деле возможность обнаружить существование некоторой удаленной системы — это не такая уж большая проблема. Для этой цели вы можете воспользоваться утилитой traceroute, которая основана на использовании UDP. Однако имейте в виду, что системы Windows для работы traceroute вместо UDP используют ICMP, поэтому если на отслеживаемом вами маршруте располагается система, блокирующая ICMP, вы не сможете воспользоваться Windows traceroute для слежения за этим маршрутом. Лично меня не очень беспокоит тот факт, что кто-то знает о существовании моей системы. Значение имеет лишь уязвимость служб, которые предлагаются системой для пользователей Интернета. Вряд ли вам удастся использовать ICMP для взлома системы.

Где искать информацию?

В наши дни информацию, связанную с защитой систем, можно получать из множества разнообразных источников. Выбор наиболее полезных из них подчас является непростой задачей. В данной

главе я упоминаю несколько списков рассылки, которые могут оказаться чрезвычайно полезными. Также я рассказываю о нескольких web-узлах, некоторые из которых поддерживают свои собственные списки рассылки.

Возможно, будет лучше, если вы будете постоянно получать информацию одновременно из нескольких источников. Например, я рекомендую вам подписаться на несколько списков рассылки и периодически обращаться к нескольким web-узлам. Все зависит от того, с какой частотой вы намерены получать информацию, в каком объеме и насколько сильно вас беспокоит тот факт, что некоторые сведения будут дублироваться в разных местах. Практика показывает, что ни один из посвященных безопасности списков рассылки нельзя считать исчерпывающим, а двукратное оповещение о некотором уязвимом месте системы — это лучше, чем полное отсутствие сведений об этом уязвимом месте.

Списки рассылки

В первую очередь я рекомендую вам обратить внимание на список рассылки, имеющий отношение к используемому вами комплекту Linux. Компания Caldera Systems поддерживает связанный с комплектом OpenLinux список рассылки под названием Caldera Security Advisory. Чтобы подписаться, необходимо обратиться по адресу www.caldera.com. Когда компании Caldera становится известно о некотором аспекте безопасности, имеющем отношение к официально распространяемому этой компанией программному обеспечению, она отправляет почтовое сообщение всем участником списка. Помимо описания проблемы сообщение содержит ссылку на исправленные пакеты RPM (если такие уже доступны) или совет относительно того, как обойти или решить проблему. Это не относится к программам, поставляемым независимыми разработчиками, например таким, которые содержатся в подкаталогах contrib. Вместо того чтобы подписываться на них, а также в случае, если вы хотите просмотреть предыдущие оповещения, вы можете обратиться к web-узлу Caldera Systems по адресу http://www.calderasystems.com/news/security/index.html.

Еще одним неплохим списком рассылки является список CERT Advisory. Организация CERT (Computer Emergency Response Team) занимается вопросами обеспечения компьютерной безопасности и отсылает подписчикам сообщения об обнаруженных уязвимых местах в системах компьютерной защиты. Оповещения отсылаются каждый раз, когда CERT обнаруживает уязвимое приложение, благодаря которому безопасность системы может понизиться. Оповещение CERT, как правило, детально описывает обнаруженную уязвимость, рассказывает о временных мерах, которые следует принять для обхода проблемы, а также содержит рекомендации по поводу надежного постоянного решения проблемы. Изучив сообщение CERT, вы можете понять, является ли ваша система уязвимой, насколько велик риск взлома вашей системы. Далеко не все уязвимые места повышают риск эксплуатации сетевого узла. Конечно, рекомендации CERT полны технического жаргона, однако они достаточно точно описывают проблему и позволяют найти решение. Если вы сомневаетесь, будет лучше последовать рекомендациям. Более подробную информацию можно получить по адресу http://www.cert.org/.

Существует также множество более общих списков рассылки и групп новостей, посвященных вопросам безопасности. Один из таких списков называется SANS NewsBites. Второе название списка — Weekly Security News Overview — еженедельный обзор новостей компьютерной безопасности. Еженедельно рассылаемое сообщение содержит смесь заметок из разных концов земного шара, посвященных компьютерной безопасности. Чтобы подписаться, достаточно отправить электронное письмо по адресу autosans@sans.org, в графе Subject (Тема) которого следует указать Subscribe NewsBites.

.

Web-узлы, посвященные компьютерной безопасности

В наши дни в Интернете функционирует огромное количество web-узлов, посвященных вопросам компьютерной безопасности. Многие из них являются всего лишь удобным местом публикации самых свежих сведений по этой тематике, а также содержат ссылки на другие ресурсы. Какие-то содержат достаточно оригинальные сведения. Существуют также web-узлы, не связанные напрямую с безопасностью, однако содержащие в себе сведения, относящиеся к защите данных, которые стали доступными для владельцев узла.

Например, такие web-узлы, как http://slashdot.org и http://freshmeat.net (этот узел содержит информацию о свежевыпущенных программных пакетах и обновлениях программного обеспечения), публикуют сведения о уязвимых местах программ, однако ни один из этих узлов не публикует никаких подробных данных. Вместо этого публикуется ссылка, которая, как правило, указывает на одно из оповещений CERT. Текст такого оповещения CERT (в переводе на русский язык) приведен в листинге

26.1. Те из вас, кто следовал инструкциям главы 21 и включил в состав компилируемых программных пакетов библиотеку RSAref, будут заинтересованы в этом сообщении.

Данное оповещение CERT сообщает нам о следующем: библиотека rsaref2 содержит в себе ошибку, благодаря которой злоумышленник может запустить в вашей системе некоторый произвольный принадлежащий ему исполняемый код. Запуск кода выполняется изнутри службы, использующей данную библиотеку от лица пользователя, на уровне привилегий которого работает данная служба (как правило, root). Чтобы исправить ошибку, необходимо применить в отношении библиотеки rsaref исправление (patch), о котором упоминается в тексте оповещения, затем следует перекомпилировать библиотеку rsaref2, а затем — перекомпилировать все приложения, которые используют эту библиотеку. Файл исправления можно получить по адресу http://www.cert.org/advisories/CA-99-15/rsa-patch.txt.

Листинг 26.1. Пример оповещения CERT

Subject: CERT Advisory CA-99.15 - Buffer Overflows in SSH Daemon and RSAREF2 Library Date: Mon, 13 Dec 1999 18:49:47 -0500 From: CERT Advisory <cert-advisoryiacert.org> Reply-To: cert-advi sory-request@cert.org Organization: CERT(sm) Coordination Center - +1 412-268-7090 To: cert-advisory@coal.cert.org

..... BEGIN PGP SIGNED MESSAGE

Hash: SHA1

CERT Advisory CA-99-15 Buffer Overflows in SSH Daemon and RSAREF2 Library Оповещение CERT CA-99-15 Переполнение буфера в демоне SSH и библиотеке RSAREF2 Дата публикации: December 13, 1999 Дата последней ревизии: --Источник: CERT/CC

Полная история всех ревизий приведена в конце данного файла. Системы, к которым относится данное оповещение:

Системы, на которых работают некоторые версии sshd

Системы, на которых используются программные продукты, использующие библиотеку RSAREF2

(например, некоторые web-серверы с поддержкой SSL) I. Описание

Некоторые версии sshd содержат в себе ошибку переполнения буфера, благодаря которой злоумышленник может влиять на значения некоторых внутренних переменных программы. Сама по себе эта ошибка не позволяет злоумышленнику исполнять какой-либо код.

Однако ошибка в библиотеке RSAREF2, обнаруженная и исследованная компанией Core SDI, может использоваться в комбинации с ошибкой в демоне sshd, благодаря чему злоумышленник получает возможность через сеть запускать в системе произвольный исполняемый код.

Дополнительная информация об ошибке в библиотеке RSAREF2 находится по адресу http//www.core-sdi.com/avisories/buffer*20overflowK201ng.htm

Библиотека RSAREF2 была разработана с использованием кода, отличающегося от кода, который использовался в других реализациях

алгоритма RSA, включая реализации, разработанные компанией RSA Security Inc. Уязвимое место (ошибка), описываемое в данном оповещении, имеет место только в библиотеке RSAREF2 и никак не влияет на безопасность других реализаций алгоритма RSA и самого алгоритма RSA.

Уязвимыми следует считать только версии демона SSH, которые были скомпилированы с поддержкой RSAREF, то есть с использованием параметра --with-rsaref.

При использовании библиотеки RSAREF2 в других программных продуктах возможно проявление других связанных с этим уязвимых мест и ошибок. Библиотека RSAREF2 может использоваться в таких программных продуктах, как web-серверы с поддержкой SSL, клиенты SSH или другие продукты, связанные с криптографией. В приложение А данного оповещения будут добавляться дополнительные сведения о проблемах,

связанных с использованием RSAREF2 в других продуктах, о которых, возможно, станет известно в будущем.

II. Эффект

Благодаря комбинации двух ошибок в двух разных программных продуктах злоумышленник получает возможность выполнить в системе произвольный исполняемый код на уровне привилегий процесса, в ранках которого функционирует демон sshd. Как правило, демон sshd работает на уровне привилегий root.

В настоящее время мы пытаемся определить, существуют ли

в других программных продуктах помимо sshd уязвимые места и ошибки,

которые позволили бы злоумышленнику воспользоваться ошибкой

библиотеки RSAREF2. В случае обнаружения таких программных продуктов

данное оповещение будет обновляться.

Более подробно о возможном эффекте данной ошибки сообщается

в приложениях А и Б данного оповещения.

III. Решение

Следует использовать исправления, предоставляемые поставщиками программного обеспечения.

Следует применить исправления библиотеки RSAREF2. Компания RSA Security Inc. обладает патентом на алгоритм RSA и правами на копирование реализации библиотеки RSAREF2. Прежде чем применять какие-либо исправления в отношении вашего программного обеспечения, мы рекомендуем вам проконсультироваться у патентных юристов относительно внесения этих изменений. Пожалуйста, изучите также заявление компании RSA в приложении А. Чтобы злоумышленник смог воспользоваться уязвимым местом библиотеки RSAREF2, прикладная программа должна обратиться к библиотеке RSAREF2, передавая ей при этом специально подготовленные некорректные данные. Если используемый вами программный продукт позволяет контролировать данные, передаваемые библиотеке RSAREF2, возможно, вы сможете блокировать атаку, если организуете проверку данных, передаваемых RSAREF2 на корректность.

Приложение А содержит сведения, связанные с этим оповещением и полученные нами от различных поставщиков программного обеспечения. В приложении Б содержится информация о тестировании, выполненном координационным центром CERT (CERT Coordination Center, CC) и другими людьми, а также советы, основанные на этих тестах. По мере получения новых сведений и выполнения новых исследований мы будем обновлять содержимое приложений. Если имя вашего поставщика программного обеспечения не упоминается в приложении А, это означает, что CERT/CC не

получали от этого поставщика никаких сведений. В этом случае рекомендуется связаться с поставщиком напрямую.
Используйте реализацию алгоритма RSA, в котором отсутствует описанная ошибка
Узлы, не ограниченные патентным законодательством, могут воспользоваться реализацией алгоритма RSA, в которой описанная ошибка отсутствует. Однако, возможно этот вариант для вас неприемлем по юридическим соображениям, так как патент на алгоритм RSA принадлежит компании RSA Security Inc. Для прояснения этого вопроса рекомендуется получить консультацию у юриста.
Приложение A. Информация поставщиков программного обеспечения Compaq Computer Corporation

ИСТОЧНИК:
Compaq Computer Corporation
Compaq Services
Software Security Response Team USA
Система Tru64 UNIX компании Compaq не содержит рассматриваемой ошибки, так как компания Compaq не поставляет поддержку ssl
Covalent Technologies
Covalent Raven SSL module for Apache Covalent Raven SSL модуль для Apache
Модуль Raven SSL не содержит в себе описанной ошибки, так как используемая этим модулем библиотека SSL не использует библиотеку RSAREF.
Data Fellows Inc.
Продукт F-Secure SSH версий ниже 1.3.7 подвержен описанной ошибке и уязвим для подобных атак, однако продукт F-Secure SSH версий 2.x не содержит ошибки и неуязвим перед атаками такого характера.
FreeBSD
Версии FreeBSD 3.3R и более ранние содержат в себе пакеты, в которых присутствует описанная проблема. Ситуация исправлена 2 декабря 1999 года в дереве адаптации. Пакеты, компонованные после указанной даты с обновленной версией rsaref, не содержат в себе описанной ошибки. Некоторые или все из перечисленных далее адаптации могут быть подвержены описанной ошибке и должны быть перекомпилированы:
p5-Penguin, p5-Penguin-Easy, jp-pgp, ja-w3m-ssl, ko-pgp, pgpsendmail, pine4-ssl, premail, ParMetis, SSLtelnet, mpich, pipsecd, tund, nntpcache, p5-Gateway, p5-News-Article, ru-pgp, bjorb, keynote, OpenSSH, openssl, p5-PGP, p5-PGP-Sign, pgp, slush, ssh, sslproxy, stunnel, apache+mod_ssl, apache+ssl, lynx-ssl, w3m-ssl, zope
Подробная информация о том, как получить свежую копию дерева адаптации и как перекомпилировать указанные адаптации, зависящие от rsaref, содержится в документации FreeBSD Handbook.
Hewlett-Packard Company
HP не поставляет SSH. Компания HP не выполняла тестирование на совместимость с версией 1.2.27 программы SSH с использованием при компиляции параметра -with-rsaref. Более того, на текущий момент библиотека RSAREF2 не тестировалась. На текущий момент складывается впечатление, что HP не подвержена этой уязвимости.
IBM Corporation
IBM AIX в настоящее время не включает в себя Secure Shell (SSH), кроме того, базовые компоненты AIX не содержат и не компонуются с использованием библиотеки RSAREF2.
IBM и AIX являются зарегистрированными торговыми марками компании International Business Machines Corporation.
Microsoft
Команда компьютерной безопасности Microsoft Security Response Team изучила данную проблему и пришла к выводу, что ни один из продуктов Microsoft не подвержен данной ошибке.
NetBSD
В настоящее время NetBSD не включает в себя SSH как в американских, так и в международных вариантах этой системы, таким образом, установка NetBSD по умолчанию не подвержена данной ошибке.
Однако ssh устанавливается и широко используется во многих установленных системах NetBSD, кроме того, ssh присутствует в дереве программных пакетов в форме исходных файлов. Пакет ssh, входящий в состав NetBSD, может быть компилирован как с использованием RSAREF2, так и без использования этой библиотеки. Это определяется администратором во время компиляции в соответствии с действующими патентными и лицензионными ограничениями.
Системы, в которых при компиляции ssh используется RSAREF2, следует считать уязвимыми. Мы рекомендуем перекомпилировать их без использования RSAREF2, если, конечно, при этом вы не нарушите действующего патентного законодательства.
Следует отметить, что в составе пакетов NetBSD присутствуют следующие пакеты, которые зависят от библиотеки RSAREF2:
archivers/hpack
security/openssl
security/pgp2
security/pgp5
www/ap-ssl
Пакет security/openssl сам по себе является библиотекой, и от этой библиотеки зависят следующие пакеты:
net/ppp-mppe
net/speakfreely-crypto
www/ap-ssl
Мы рекомендуем заново откомпилировать и установить эти пакеты без использования RSAREF2, если, конечно, это допустимо с точки зрения действующего патентного законодательства.
Network Associates, Inc.

После технического анализа ошибки переполнения буфера в библиотеке RSAREF мы пришли к выводу, что система PGP не подвержена данной ошибке благодаря тому, что PGP очень осторожно использует RSAREF.
Это относится ко всем версиям PGP, когда-либо выпущенным MIT. Именно версии, выпущенные MIT, основаны на использовании RSAREF. Все остальные версии PGP, в частности коммерческие версии и международные версии, вообще не используют библиотеку RSAREF.
Philip Zimmermann 10 December 1999
[Замечание CERT/CC: Подписанная с использованием PGP копия данной информации и дополнительные технические детали также доступны для ознакомления.]
OpenSSL
Пакет OpenSSL с использованием RSAREF не подвержен данной ошибке. OpenBSD / OpenSSH
Более подробную информацию можно получить по адресу:
http://www.openbsd.org/errata.html#sslUSA RSA Security Inc.
RSA Security Inc. рекомендует разработчикам внести предлагаемое или аналогичное исправление в библиотеку RSAREF версии 2.0 или в противном случае обеспечить выполнение следующего условия: длина в битах модуля, передаваемого RSAREF, должна быть меньше или равна константе MAX_RSA_MODULUS_BITS.
Компания RSA Security Inc. больше не распространяет комплект разработки RSAREF, который в середине 1990-х годов распространялся RSA Laboratories как бесплатные исходные коды современных криптографических алгоритмов. В соответствии с условиями лицензирования RSAREF изменения в эту библиотеку можно вносить только с целью адаптации или улучшения производительности. Для внесения изменений другого характера требуется письменное разрешение компании RSA Security. Данным заявлением компания RSA Security предоставляет разрешение на внесение изменений в RSAREF для того, чтобы исправить описанную ошибку.
Данное оповещение имеет отношение только к библиотеке RSAREF и никак не связано с другими продуктами и программными пакетами компании RSA
Security, которые разрабатывались независимо от RSAREF.
Несмотря на то, что RSA Security больше не распространяет RSAREF, этот пакет все еще распространяется в составе некоторых свободно распространяемых (freeware) продуктов, таких как SSH, на условиях изначального лицензионного соглашения RSA Security RSAREF v2.0 (файл "license.txt" датированный 25 марта 1994 года), текст которого входит в состав данных продуктов. Напоминаем, что данное лицензионное соглашение разрешает использование RSAREF только в некоммерческих целях. Библиотека RSAREF, приложения RSAREF, а также службы, основанные на RSAREF, не могут быть проданы, лицензированы или иным образом использованы для получения прибыли. Существует незначительное исключение для небольших условно-бесплатных (shareware) программ, о чем дополнительно сказано в файле "info.txt", датированном 25 марта 1994 года.
SSH Communications
Ошибка проявляется в SSH только в случае, если эта программа компилируется с использованием RSAREF (то есть только в случае, если в командной строке явно указан параметр -with-rsaref). Любая версия, откомпилированная без использования параметра -with-rsaref, не содержит в себе описываемой ошибки. Проблема не имеет отношения к коммерческим версиям (использующим лицензированный встроенный код RSA), а также к пользователям SSH, находящимся вне Соединенных Штатов (предполагается, что такие пользователи не используют RSAREF и могут использовать встроенный код RSA, не обладая соответствующей лицензией). Иными словами, данная ошибка угрожает только некоммерческим пользователям, которые осуществляют компиляцию с использованием библиотеки RSAREF, полученной независимо от SSH.
Ошибка присутствует во всех версиях SSH1, вплоть до и включая 1.2.27. В версии ssh-1-2.28 (которая должна появиться в течение ближайших дней) эта ошибка будет исправлена. В SSH2 эта ошибка отсутствует. (Обратите внимание, что продукт sshl теперь поддерживается только для осуществления исправлений ошибок - дальнейшее развитие sshl прекращено из-за фундаментальных проблем, которые исправлены в ssh2.)
Любая реализация, откомпилированная без явного указания параметра --with-rsaref, не подвержена данной ошибке.
Исправление, предоставленное компанией SSH Communications,
можно получить на web-узле CERT/CC. Эта версия исправления подписана
CERT/CC.
Stronghold
Продукт Stronghold не использует RSAREF и поэтому не подвержен данной ошибке.
Приложение Б. Тесты, выполненные CERT/CC и другими людьми и организациями Исправление RSAREF от Core SDI и CERT/CC
При поддержке компании Core SDI Координационный центр CERT выполнил тестирование демона sshd версии 1.2.27, функционирующего в операционной среде Intel RedHat Linux, и обнаружил, что его конфигурация уязвима для описанной атаки. Тестирование, выполненное компанией Core SDI, указывает на то. что демон sshd версии 1.2.27, работающий в среде OpenBSD и FreeBSD на платформе Intel, также подвершен этой ошибке. Вероятнее всего, что другие конфигурации демона также подвержены ошибке.
На основе исследований Core SDI центр CERT/CC разработал исправление библиотеки RSAREF2.
Это исправление можно получить по адресу
ftp://ftp.core-sdi.com/pub/patches/rsaref2.patch
http://www.cert.org/advisories/CA-99-15/rsa-patch.txt
Получив его от CERT/CC, вы можете проверить это исправление без подписи PGP.
Мы полагаем, что исправление, изначально предлагаемое Core SDI в их оповещении, может оказаться неполным. То есть в результате его использования проблема может оказаться решенной не полностью. Мы приложили усилия для того, чтобы разработать более свежее исправление на основе их работы, о чем и указывается в данном оповещении. Ни CERT/CC, ни Software Engineering Institute, ни Carnegie Mellon University не предоставляют каких-либо гарантий, связанных с данным исправлением. Пожалуйста, ознакомьтесь с нашим заявлением об отказе от гарантий в конце данного оповещения.
Возможная уязвимость клиентов ssh
В настоящее время у нас вызывает озабоченность возможная уязвимость

клиентов ssh. Если мы сможем получить дополнительную информацию, связанную с уязвимостью клиентов ssh, мы обязательно обновим данное оповещение. Один из возможных способов атаки клиента ssh предусматривает создание злонамеренного сервера ssh. Злоумышленник может заманить или каким-либо иным обманным образом заставить клиента подключиться к этому серверу, в результате чего в клиентской системе может быть запущен произвольный код. Клиент ssh предупреждает пользователей о том. что происходит подключение к узлу, который предоставил ключ, не соответствующий ключу, ранее ассоциированному с сервером. Соответствующее предупреждающее сообщение может выглядеть приблизительно следующим образом:

```
% ssh badhost
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@     WARNING: HOST IDENTIFICATION HAS CHANGED!      @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the host key has just been changed.
Please contact your system administrator.
Add correct host key in /etc/.ssh/known_hosts to get rid of this message.
Are you sure you want to continue connecting (yes/no)? no
%
```

Если вы увидите данное предупреждение, вы должны ответить "по" в ответ на приглашение и выяснить, почему принятый вами от сервера ключ не соответствует ключу, который вы ожидаете.

Координационный центр CERT выражает благодарность Alberto Solino <Alberto_Solino@core-sdi.com> и Gerardo Richarte <Gerardo_Richarte@core-sdi.com> из компании Core SDI S.A. Seguridad de la informacion, Buenos Aires, Argentina (http://www.core-sdi.com), которые обнаружили проблему в RSAREF2 и обеспечили ценную техническую поддержку.
Мы также хотели бы поблагодарить Andrew Cormack из JANET CERT за то, что он обеспечил техническую поддержку; Theo de Raadt из проекта OpenBSD за ценную информацию, использованную при составлении данного оповещения; Burt Kaliski из RSA Security Inc.; и Tatu Ylonen из SSH Communications Security.

Данный документ располагается по адресу: http://www.cert.org/advisories/CA-99-15-RSAREF2.html

---

CERT/CC Контактная информация
Emai1: certPcert.org
Телефон: +1 412-268-7090 (24-hour hotline)
Факс: +1 412-268-6989
Почтовый адрес:
CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
U.S.A.
Персонал CERT отвечает по телефону горячей линии во время 08:00-20:00 EST(GMT-S) / EDT(GMT-4) с понедельника по пятницу.
Использование шифрования
Мы настоятельно рекомендуем шифровать важную информацию, передаваемую через электронную почту. Наш открытый (публичный) ключ PGP можно получить по адресу
http://www.cert.org/CERT_PGP.key
Если вы предпочитаете использовать DES, пожалуйста, свяжитесь с горячей линией CERT для получения дополнительной информации.
Получение информации о компьютерной безопасности
Публикации CERT и другие сведения, связанные с компьютерной безопасностью, можно получить на нашем Web-узле по адресу:
http://www.cert.org/
Чтобы стать членом нашего списка рассылки и получать оповещения и бюллетени, следует послать электронное письмо по адресу cert-advisory-request@cert.org, в поле Subject (Тема) этого письма следует включить текст SUBSCRIBE <ваш-электронный-адрес>.

марки.

_____

История ревизий Декабрь 13, 1999: Изначальный выпуск

.....BEGIN PGP SIGNATURE

Version: PGP for Personal Privacy 5.0 Charset: noconv

iQA/AwUBOFV9alr9kb5qlZHQEQI7bACglxlZVHntIvhRHjUlf8BaNVGJlbkAnA6Y

kOuU3ddT09uguGEvOEuR9Rw3

=IqXt

.....END PGP SIGNATURE

Прочитав данное оповещение, вы можете обратить внимание на уровень его детализации. Оповещение может показаться излишне подробным. Однако, к счастью, подобные оповещения рассылаются не так уж часто, поэтому если вы войдете в состав этого списка рассылки, вы не очень сильно загрузите ваш почтовый сервер.

Web-узел под названием SecurityFocus также является неплохим источником информации о компьютерной безопасности. Его адрес http://www.securityfocus.com. Раньше этот узел назывался BugTraq, и сейчас он стал в большей степени коммерческим. Узел по-прежнему предлагает значительный объем полезных сведений, однако он в большей степени ориентирован на компьютерных профессионалов, а не на простых смертных. На узле поддерживается несколько списков рассылки для людей с самыми разными интересами.

Еще один узел поддерживается Лэнцем Спицем (Lance Spitz) и располагается по адресу http://www.enteract.com/~lspit2/pubs.html, однако этот узел является чисто информационным. Здесь вы найдете богатый материал о том, как работают взломщики и что им можно противопоставить. Информация на данном узле в большей степени предназначена для опытных пользователей, однако наверняка любой посетитель узла узнает для себя что-нибудь ценное. Материалы, публикуемые на узле, можно либо читать прямо из Интернета, либо предварительно загрузить их на локальный диск.

Создатель сканирующей программы nmap, человек по прозвищу Fyodor, поддерживает свой собственный web-узел, посвященный вопросам безопасности. Этот узел располагается по адресу http://www.insecure.org/. Здесь вы найдете сведения о разнообразных уязвимых местах программного обеспечения, а также самую свежую информацию о сетевом сканировании. Здесь же описываются самые свежие технологии, позволяющие избежать обнаружения современными детекторами сканирования.

Узлы сомнительной направленности

Некоторые из web-узлов публикуют не только оповещения о возможных уязвимых местах программ: эти узлы делают доступными для широкой интернет-общественности исполняемый код, который можно использовать для того, чтобы взламывать системы, обладающие уязвимыми местами. Многие посетители подобных узлов подключаются к ним с одной лишь целью: загрузить код, позволяющий взломать защиту чужой системы. Чаще всего эти люди не имеют представления о том, как именно работает данный код, они не знают о характере уязвимости и о том, как именно происходит нарушение защиты, однако они получают возможность просто запустить машинный код или сценарий и таким образом проникнуть туда, где их не ждут. Чаще всего подобные злоумышленники оказываются подростками, жаждущими продемонстрировать свою антисоциальную направленность.

Старейшим и наиболее хорошо известным узлом подобной направленности является узел http://www.rootshell.com/. На этом узле можно найти новости о различных способах взлома, архивы, описания взломов систем, которые имели место в прошлом, а также код, позволяющий воспроизвести эти атаки.

Еще одним узлом этой категории является узел Cult of the Dead Cow (культ дохлой коровы), расположенный по адресу http://www.cultdeadcow.com/. Этот узел является не просто излюбленным местом малолетних компьютерных хулиганов. Члены команды cDc являются авторами программы Back Orifice — знаменитого средства удаленного управления операционной системой Windows.

Еще одним узлом, достойным внимания, является домашняя страничка журнала 2600, который называет себя «хакерский ежеквартальник». В журнале и на web-узле термин «хакер» используется не совсем корректно. Под этим термином авторы журнала понимают взломщиков и компьютерных хулиганов. Узел журнала располагается по адресу http://www.2600.com/.

Все упомянутые узлы громогласно объявляют о том, что не одобряют незаконного использования предоставляемых ими сведений и программного обеспечения. Они как бы говорят посетителям: «вот программа, с помощью которой можно взломать чужую систему; мы сами ею не пользуемся и вам не советуем». При желании вы можете посетить эти узлы, однако имейте в виду, что по большей части этот материал ориентирован на 12-летних подростков.

Последнее замечание

Если вы просматриваете программные пакеты, которые вы намерены загрузить из Интернета, вы часто можете видеть файлы, содержащие в себе сигнатуры PGP (Pretty Good Privacy). Сигнатуру (подпись) PGP можно видеть в конце оповещения CERT в листинге 26.1, при помощи такой сигнатуры вы можете проверять программное обеспечение, размещенное на web-узле. Сигнатуры PGP часто используются в отношении программ, обеспечивающих работу служб с ограниченным доступом, а также для защиты важных сообщений, таких как оповещения CERT. Конечно, вы можете просто поверить в тот факт, что никто не вносил никаких изменений в расположенное на web-узле программное обеспечение, однако надежнее будет выполнить проверку программного пакета при помощи сигнатуры PGP. Сигнатура сообщает вам имя автора пакета, а также находится ли пакет в том же состоянии, в котором сформировал его автор. Вы можете использовать сигнатуру или нет, однако, как правило, она прилагается к программному пакету. Чтобы использовать сигнатуру, вы должны установить в системе PGP.

Заключение

Безопасность — это не то, что вы изучаете только один раз и продолжаете использовать в течение всего времени. Безопасность требует, чтобы вы постоянно продолжали процесс обучения. Каждый день появляются все новые программы, осуществляющие сканирование и взлом систем. Со временем методы взлома становятся все более изощренными. Вы сможете успешно противостоять атакам только в случае, если постоянно будете в курсе самых свежих событий в области компьютерной безопасности. Помните, что если ваша система будет чуточку более защищенной, в нее будет чуточку сложнее пролезть, чуточку сложнее, чем в другие системы, работающие в Интернете. И в этом случае, если вы, конечно, не один из часто посещаемых web-узлов, у вас не будет слишком больших проблем с компьютерными хулиганами. Дело в том, что эти самые хулиганы, как правило, предпочитают иметь дело с легкой добычей, не обнаружив в вашей системе хорошо известных лазеек, они теряют к ней интерес. Если в любой из поисковых систем Интернета вы введете Linux security, вы сможете получить множество дополнительных ссылок и дополнительной информации о безопасности в среде Linux.

A Обзор средств сетевого сканирования и утилит безопасности

В данном приложении приведен перечень ссылок на сетевые сканеры и утилиты безопасности. Этот список не следует считать исчерпывающим — постоянно появляются новые программы, а старое программное обеспечение выходит из употребления, и его дальнейшее развитие прекращается. Изучив данное приложение, вы получите представление о том, что в настоящий момент доступно для использования. Наиболее хорошими источниками подобных средств являются узлы http://www.linuxberg.com/ и http://freshmeat.net/. Оба этих узла поддерживают поиск и предлагают разнообразную информацию о безопасности, а также связанное с этим программное обеспечение.

Далее следует алфавитный список некоторых средств, связанных с безопасностью, которые вы можете загрузить и установить в своей системе. Указывается домашняя страница приложения или, если домашней страницы нет, место, откуда приложение можно загрузить.

AIDE: свободно распространяемая замена trip wire (http://www.cs.tut.fi/~rammer/ aide.html).

BASS: Bulk Auditing Security Scanners (загружается по адресу http://www. securityfocus.com/data/tools/network/bass-1.0.7.tar.gz).

Bastille Linux: программа, усиливающая защиту Red Hat Linux 6.0 (http:// bastille-linux.sourceforge.net/).

Check.pl: средство аудита разрешений на доступ к файловой системе (http:// checkps.alcom.co.uk/).

firesoft: инструменты для просмотра журналов ipchains и snort (http://www. unix.gr/).

Firewall Manager: графический интерфейс для брандмауэров (http://www. tectrip.net/arg/).

FreeS/WAN: безопасная глобальная сеть WAN для ядер Linux 2.0 и 2.2 (http:// www.xs4all.nl/~freeswan/).

Fwctl: высокоуровневое средство конфигурирования для пакетных фильтров Linux 2.2 (http://indev.insu.com/Fwctl/).

gfcc: брандмауэр GTK+ (ipchains) (http://icarus.autostock.co.kr/).

gSentiel: основанный на GTK графический интерфейс для Sentiel (http:// zurk.netpedia.net/zfile.html).

gSheild: модульный брандмауэр Godot (http://muse.linuxgeek.org/).

HostSentry: средство обнаружения аномальных подключений к системе и реагирования на них (http://www.psionic.com/abacus/hostsentry/).

hping2: инструмент аудита и тестирования сети (http://www.kyuzz.org/antirez/ hping2.html).

ipchains: утилита управления пакетным фильтром Linux (для ядер 2.2.#) (http:// www.rustcorp.com/linux/ipchains/).

ipchains-firewall: набор сценариев для формирования правил для ipchains и маскировки IP (http://ipchains.nerdherd.org/).

ipfa: средство учета и контроля для брандмауэра IP (http://www.soaring-bird.com. cn/oss_proj/ipfa/).

ISIC: программа, которая посылает контролируемые частично случайные пакеты для тестирования стеков IP и брандмауэров (http://expert.cc.purdue.edu/ -frantzen/).

John the Ripper: взломщик паролей для обнаружения слабых паролей Unix (http://www.openwall.com/john/).

Linux Intrusion Detection System: программное средство, обнаруживающее проникновение

в систему злоумышленников; работающее на уровне ядра (http://www.soaring-bird.com.cn/oss_proj/lids/).

Logcheck: помогает обнаруживать проблемы и нарушения защиты системы на основании сведений, содержащихся в файлах журналов (http://www.psionic.com/).

maillog: передает системные журналы по почте на указанный почтовый адрес, для этого используется задание cron (http://old.dhs.org/).

Mason: автоматически формирует брандмауэр на базе продуктов ipfwadm или ipchains (http://www.pobox.com/~wstearns/mason/).

Nessus: простое в использовании средство аудита безопасности (http://www. nessus.org/).

netfilter: программный пакетный фильтр IP для ядер Linux *2A.x* (http://www. samba.org/~netfilter/).

nmap: средство сканирования сети (http://www.insecure.org/nmap/).

nstreams: анализатор сетевых потоков (http://www.hsc.fr/cabinet/produits/index. html.en).

OpenSSH: версия защищенной оболочки Secure Shell с открытым кодом (http:// www.openssh.com/).

Ping Sting: идентификатор трафика ICMP (http://www.ksrt.org/psting/).

PMFirewall: утилита конфигурирования ipchains в режиме брандмауэра и маскировки IP (http://www.pointman.org/).

PortSentry: в реальном времени обнаруживает попытки сканирования портов и реагирует на них (http://www.psionic.com/abacus/portsentry/).

PSPG: Pretty Simple Password Generator — программа генерации паролей (http:// members.xoom.com/miscreants/).

QIPchains: сценарий командной оболочки, который позволяет быстро добавлять/удалять правила брандмауэра Linux (http://www.vano.odessa.net/software/).

redir: перенаправляет порт TCP на другой IP-адрес и порт (http://sammy.net/ ~sammy/hacks/).

S/key: система одноразовых паролей (можно загрузить по адресу ftp://thumper. bellcore.com/pub/nmh/).

SAINT: Security Administrator's Integrated Network Tool — интегрированный сетевой инструмент администратора безопасности (http://www.wwdsi.com/saint/).

samhain: средство проверки целостности файлов (http://samhain.netpedia.net/).

SARA: средство аудита безопасности, подобное SATAN/SAINT, если в системе установлена программа nmap, использует nmap (http://home.arc.com/sara/ index.html).

secure delete: безопасное удаление файлов, безопасная перезапись виртуальной памяти и незанятого дискового пространства (только для загрузки http:// thc.pimmel.com/files/thc/secure_delete-2.1.tar.gz).

Secure Remote Password Protocol: протокол аутентификации и обмена ключами, основанный на использовании паролей (http://srp.stanford.edu/srp/).

Secure Shell (ssh): программное средство для защищенных подключений с использованием шифрования и взаимной авторизации (загрузка: ftp://ftp.cs.hut.fi/ pub/ssh/; домашняя страница: http://www.ssh.fi/).

Secure-Linux Patch: исправление ядра Linux, позволяющее блокировать большую часть атак, основанных на переполнении стека (http://www.openwall.com/ linux/).

Sentiel: средство быстрого сканирования системных файлов (http://zurk. netpedia.net/zfile.html).

sifi: пакетный фильтр TCP/IP для Linux, основанный на состояниях (http:// www.ifi.unizh.ch/ikm/SINUS/firewall/).

Slinux Kernel: улучшенное в плане безопасности ядро Linux (http://www. slinux.cx/).

snort: система обнаружения несанкционированного проникновения в сеть (http://www.clark.net/~roesch/security.html).

sslwrap: служба, выполняющая функцию SSL-оболочки для других серверных

приложений, таких как демоны POP3/IMAP (http://www.rickk.com/sslwrap/).

Sportal: средство слежения за файлами с графическим интерфейсом GTK (http://sportal.sourceforge.net/).

sXid: средство слежения за suid/sgid, написанное на C (загрузка: ftp://marcus. seva.net/pub/sxid/).

TARA (Tiger Analytical Research Assistant): набор сценариев для проверки локальной безопасности (http://home.arc.com/tara/index.html).

The Phreak Firewall: средство установки настройки вашего брандмауэра с использованием маскировки IP (http://bewoner.dma.be/Phreak/).

TheBox: набор сценариев для установки, настройки и управления механизмами маскировки IP и прозрачного кэширования (http://yak.airwire.net/).

Triplight: средство обнаружения несанкционированного проникновения в систему и целостности файлов, использует md5surn для проверки перечня файлов (http://linux.rice.edu/magic/triplight/).

Tripwire: система обнаружения несанкционированного доступа для Linux (http://www.tripwiresecurity.com/).

Wipe: безопасное удаление файлов с магнитных носителей (http://gsu.Hnux. org.tr/wipe/).

ya-wipe: безопасное удаление файлов (http://www.erols.com/thomassr/zero/ download/wipe/).

Б Что на компакт-диске?

На прилагающемся к данной книге компакт-диске содержится следующее:

Полный комплект Caldera System OpenLinux 2.3 (только бинарные файлы). Чтобы установить эту операционную систему, необходимо, находясь в операционной среде Windows, вставить данный компакт-диск в устройство чтения CD-ROM. Произойдет автоматический запуск, и на экране появится меню. Следуйте отображаемым на экране инструкциям. Если позволяет ваша система BIOS, для начала установки OpenLinux вы можете также просто загрузить компьютер с компакт-диска. Для получения дополнительной информации посетите web-узел компании Caldera Systems (www.calderasystems.com)

Также на прилагаемом компакт-диске можно обнаружить несколько каталогов, представляющих интерес:

каталог ../col/contrib/RPMS содержит неофициальные, не поддерживаемые компанией Caldera пакеты RPMS;

каталог ../col/security/RPMS содержит в себе некоторые представляющие интерес пакеты:

- courtney-1.3 — сценарий Perl, позволяющий обнаруживать попытки скани рования системы;
- dailyscript-3.9.2 — набор сценариев, позволяющих автоматизировать еже дневные рутинные служебные процедуры в отношении системных журна лов — в частности, поиск аномалий и подозрительных записей в журналах (сценарии модифицированы специально для Caldera OpenLinux);
- ipmasqadm-0.4.1 — программа, обеспечивающая перенаправление портов с использованием ipchains;
- libpcap-0.4 — библиотека, необходимая для работы разнообразных средств прослушивания сетевого трафика;
- makepasswd-1.7 — программа для создания надежно защищенных паролей;
- tnason-0.13.0.92 — средство, помогающее формировать наборы правил для брандмауэров ipchains и netfilter;
- mucreate-0.1 — сценарий создания нескольких пользователей с защищенны ми паролями;
- nmap-2.BETA10 — средство сканирования сетей (в настоящее время одно из наиболее популярных);
- rfc-0.2 — сценарий Perl для просмотра документов RFC;
- squid-2.2Stable5 — программное обеспечение кэширующего proxy (если в ка талоге contrib/ содержится более ранняя версия, не используйте ее);
- sudo-1.6.1 — программа, позволяющая пользователям запускать программы от имени root, не выступая при этом от лица пользователя root и даже не зная пароля root;
- tknotepad-7.1 — сценарий tk, по своим функциональным возможностям напоминающий программу Windows Notepad (блокнот) и позволяющий в удобной форме редактировать конфигурационные файлы.

В подкаталоге tarballs можно обнаружить еще две программы, которые поставляются не в виде пакетов RMP, а в виде компрессированных пакетов tar:

- nessus-0.99.1 — программное средство сканирования сетей;

- SAINT — программное средство сканирования сетей, дальнейшее развитие программы SATAN;

Две последние программы содержатся в компрессированных пакетах tar, в комплект входят также подкаталоги с компилированными бинарными файлами. Чтобы установить и использовать nessus, необходимо в заданном порядке войти в каждый из перечисленных далее подкаталогов и выполнить команду make install:

nessus-libraries
libnasl
nessus-core
nessus-plugins

После установки libasl, но перед самым первым запуском программы следует запустить ldconfig.

Чтобы использовать SAINT, необходимо скопировать весь каталог saint с компакт-диска в каталог по вашему выбору (например, в каталог /root) и запустить эту программу оттуда.

Алфавитный указатель

A—B

C-D

E—F

G-H

I—K

L—M

N-O

P—O

R-S

Т—И

V—Z

А-Б

Г-Ж

З-Й

К—М

О—П

P-C

Т—Ш

ЧЬгЯ_