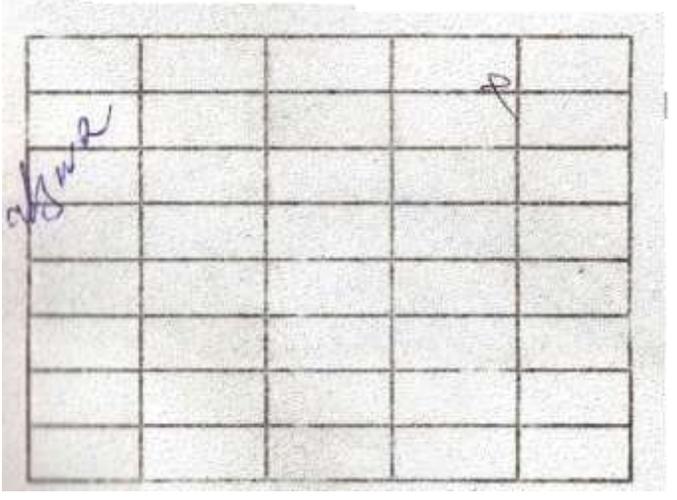

004
0-546
jB. Г.
Олифер Н. А.
Опийел
Безопасность компьютерных сетей
NTC
Network Training Center

В. Г.
Олифер
НИОлифер

Безопасность компьютерных сотой



Москва
Горячая линия - Телеком
2017

ББК 32.973.202 0-54

Олифер В. Г., Олифер Н.А.

0-54 Безопасность компьютерных сетей. - Телеком, М.: Горячая линия-2017. - 644 с.: ил.

ISBN 978-5-9912-0420-0.

Систематизированы обширные теоретические и практические сведения в области методов и способов обеспечения безопасности компьютерных сетей. Рассмотрены меры обеспечения безопасности компьютерных систем как органической части общей информационной системы предприятия; методы защиты программного обеспечения компьютеров и обрабатываемой ими информации; сетевые аспекты передачи информации между узлами компьютерной сети (вопросы безопасности сетевых протоколов и сервисов); базовые технологии, используемые для защиты информации в компьютерной сети, такие как шифрование, аутентификация, авторизация, организация защищенного канала и другие, которые в той или иной мере являются основой всех методов обеспечения безопасности компьютерных сетей.

Книга структурирована в виде учебного курса и отличается широким охватом затронутых тем, при этом авторы стремились сохранить достаточную глубину рассмотрения вопросов, позволяющую понять их суть. Все главы книги завершаются набором вопросов для проверки и самопроверки.

Для широкого круга читателей, которые хотят углубить и систематизировать свои знания в области безопасности компьютерных сетей. Будет особенно полезна для специалистов в области информационной безопасности, занимающихся практическими вопросами построения комплексных систем защиты информации, слушателей курсов переподготовки и повышения квалификации, студентов и аспирантов, обучающихся по направлению «Информационная безопасность».

32.973.202

Адрес издательства и Интернет www.techbook.ru

ISBN 978-5-9912-0420-0

ВВЕДЕНИЕ

Посвящается Ванечке

Безопасность компьютерных сетей обеспечивается разнообразными мерами и способами, которые в зависимости от их природы можно объединить в четыре большие группы.

В первую группу входят меры обеспечения безопасности компьютерных систем как органической части общей ***информационной системы предприятия.***

Вторая группа включает методы защиты ***программного обеспечения*** компьютеров и обрабатываемой ими информации.

Третья группа относится к ***сетевым аспектам*** передачи информации между узлами компьютерной сети и имеет дело с безопасностью сетевых протоколов и сервисов.

Четвертая группа включает ***базовые технологии,*** используемые для защиты информации в компьютерной сети, такие как шифрование, аутентификация, авторизация, организация защищенного канала и другие, которые в той или иной мере являются основой всех методов обеспечения безопасности компьютерных сетей.

Построение предлагаемой читателю книги отражает приведенный выше подход к структуризации методов обеспечения безопасности. Книга отличается широким охватом проблем каждой группы защитных мер, при этом авторы стремились сохранить достаточную глубину рассмотрения вопросов, позволяющую понять их суть.

Первая часть книги знакомит читателя с основными понятиями и принципами информационной безпасности, такими как идентификация, аутентификация, авторизация. Здесь рассматриваются основные типы угроз и атак, таких как отказ в обслуживании, внедрение вредоносных программ, кража личности, фишинг, сетевая разведка. Большое внимание уделено различным методикам оценки ущерба и управления рисками, связанными с атаками на информационную систему предприятия. Далее описываются законодательно-правовые ас-V пекты защиты информации: приводятся основные законы РФ, регулирующие деятельность в области информационной безопасности и определяющие ответственность за различные виды кибернетичких преступлений, рассматриваются вопросы стандартизации и сертификации средств защиты информации. Заключительные разделы первой части посвящены многоуровневому построения политики безопасности предприятия.

В качестве иллюстрации приводится документ, детально описывающий политику безопасности компьютерной инфраструктуры некоторого предприятия.

Вторую часть книги, посвященую базовым технологиям безопасности компьютерных сетей, открывает глава о криптографии — краеугольном камне всех служб информационной безопасности, будь то система аутентификации или авторизации, защищенный канал или средства безопасного хранения данных. Приводятся алгоритмы симметричного шифрования по методам DES и AES, а также шифрования с открытым ключом и односторонние функции. Далее рассматриваются различные способы аутентификации, основанные на одноразовых и многоразовых паролях, использовании цифровых сертификатов и цифровой подписи. К базовым технологиям отнесены также различные методы и подходы к авторизации/ управлению доступом, подробно рассматриваются мандатный, дескреционный и ролевой способы. При изучении конкретных алгоритмов и механизмов приводятся подробные примеры их работы, позволяющие проследить все основные этапы преобразования информации, обеспечивающие ее защиту, и избежать ошибок неверного понимания сути алгоритма. К числу важнейших технологий поддержания сетевой безопасности отнесены также технология защищенного канала, фильтрация анализ трафика и аудит состояний сети.

В третьей части книги рассматриваются проблемы безопасности транспортной инфраструктуры современной компьютерной сети. В транспортную инфраструктуру включаются все промежуточные узлы сети, представляющие собой маршрутизаторы, коммутаторы, а также транспортные средства операционных систем серверов и пользовательских компьютеров, установленных в конечных узлах сети. Для каждого из основных протоколов, обеспечивающих работу транспортной инфраструктуры сети — IP, TCP, UDP, ICMP, рассмотрены уязвимости и атаки, которые осуществлялись либо на данный протокол, препятствуя передаче информации между узлами сети, либо на хосты сети с использованием этого протокола, а также даются рекомендации по защите от данных атак. Особое внимание уделяется безопасности службы DNS и протоколу маршрутизации BGP — двум очень важным элементам архитектуры Интернет, которые обеспечивают связность составляющих сетей и узлов в глобальном масштабе. Приводится достаточно подробное описание устройства службы DNS, а также различных типов атак на нее — DNS-спуффинг, отравления DNS-кэша, DDoS-атаки на корневые серверы DNS, а также получившие в последнее время распространение DDoS-атаки на хосты с помощью трафика, отраженного от DNS-серверов. Аналогичный подход используется и при описании уязвимостей и методов защиты протокола BGP. Протокол маршрутизации BGP редко подвергается атакам типичных хакеров, так как он работает между маршрутизаторами, находящимися под управлением провайдеров Интернет. Однако ошибки провайдеров в конфигурировании маршрутизаторов BGP могут приводить к

катастрофическим для пользователей Интернет последствиям. В книге описываются несколько таких глобальных инцидентов, а также рассматриваются методы защиты протокола BGP, удостоверяющие подлинность маршрута на основе системы цифровых сертификатов.

В этой части книги также изучаются протокольно-независимые методы защиты транспортной инфраструктуры сети. Одним из таких методов является разбиение сети на логические зоны, включая демилитаризованную зону и несколько внутренних зон, и защиты их с помощью файерволов и систем обнаружения вторжения. Рассматриваются системы мониторинга трафика на основе сниферов и агентов протокола NetFlow, которые позволяют распознать атаку за счет выявления отклонений образцов трафика от стандартного поведения. Важным и популярным средством обеспечения безопасности сетевых коммуникаций являются виртуальные частные сети (VPN), работающие поверх стандартной IP-сети. В книге дается классификация этих сетей и рассматриваются преимущества и недостатки каждого типа. Завершается третья часть рассмотрением особенностей угроз и методов защиты для различных моделей облачных сервисов.

Четвертая часть книги посвящена безопасности системного и прикладного программного обеспечения. Она начинается с изучения архитектурной безопасности операционных систем, обеспечиваемой изоляцией адресных пространств процессов и концепцией ядра. Большое внимание уделяется системам аутентификации и авторизации операционных систем. Если во второй части книги изучались теоретические основы аутентификации и авторизации, то в четвертой части рассматриваются конкретные механизмы аутентификации и авторизации операционных систем семейства Unix/Linux и MS Windows. Изучаются как локальные системы аутентификации и авторизации, работающие в пределах ОС отдельного компьютера, так и доменные системы, обеспечивающие централизованное управление пользователями группы компьютеров. Подробно рассмотрены механизмы централизованной системы аутентификации Kerberos, работающей как в доменах MS Windows на основе справочной системы Active Directory, так и в доменах Unix/Linux. Приводятся описания средств управления паролями в средах MS Windows и Unix/Linux, а также даются рекомендации по их защите на основе рациональной политики безопасности. Системы аудита операционных систем изучаются на примере журналов сервиса syslog в среде Unix и журнала Security Log в среде Windows. Далее в этой части книги описываются типичные уязвимости программного кода и вредоносные программы, использующие эти уязвимости, — троянские кони, черви, вирусы и скрытые каналы. Отдельная глава посвящена стандартам сертификации программных систем на безопасность. Здесь приводятся требования двух систем стандартов — «Оранжевой книги» и «Общих критериев». Все главы книги завершаются набором вопросов для проверки и самопроверки.

Книга рассчитана на широкий круг читателей, которые хотят углубить и систематизировать свои знания в области безопасности компьютерных сетей. Для эффективного чтения книги желательно предварительное знакомство с принципами работы компьютеров и операционных систем, а также основами сетевых технологий.

Часть I

Основы информационной безопасности

Обеспечение безопасности является чрезвычайно широкой областью знаний, тематика которой простирается от инстинкта самосохранения элементарного живого организма до безопасности проведения космической военной операции. И хотя в основе всех этих разных «безопасностей» можно найти много общего (хотя бы важность маскировки или сохранения секретов), в этой книге мы ограничимся только специфическими вопросами ***безопасности компьютерных сетей.***

Вместе с тем, нельзя игнорировать тот факт, что каждая компьютерная сеть является частью некоторой ***информационной системы*** (ИС), которая в свою очередь тесно связана с бизнесом, для обслуживания которого она предназначена. В отличие от небольших предприятий, информационные системы которых практически совпадают с их компьютерной инфраструктурой (компьютерной сетью), ИС крупных предприятий и организаций существенно отличаются от компьютерных сетей, лежащих в их основе. Так, ИС может дополнительно включать такие компоненты, как средства связи (учрежденческие АТС, каналы связи и канальное оборудование, телефоны, факсимильные аппараты, мобильные средства связи), системы поддержки бумажного документооборота (копировально-множительную аппаратуру, библиотеки и хранилища), организационные компоненты, направленные на совершенствование структуры предприятия, штатного расписания, должностных инструкций персонала, и многие другие. Отсюда следует, во-первых, что понятие «безопасность компьютерной сети» значительно уже, чем понятие «безопасность информационной системы», а во-вторых, что они не могут рассматриваться изолированно, в отрыве одно от другого.

Именно эта задача — показать взаимозависимость проблем защиты ИС и компьютерных сетей — решается в этой части. С этой целью здесь с самых общих позиций рассматриваются базовые понятия информационной безопасности, методика управления рисками, многоуровневая структура средств обеспечения безопасности, принципы защиты информационной системы.

1 *Основные понятия и принципы безопасности*

Термины и определения

Существует много различных трактовок термина «информационная система». Так, международный стандарт ISO/IEC 2382-1 даёт широкое определение информационной системы: «Информационная система — система обработки информации, работающая совместно с организационными ресурсами, такими как люди, технические средства и финансовые ресурсы, которые обеспечивают и распределяют информацию».

Несколько более узкое определение даётся в Федеральном законе РФ: «информационная система — совокупность содержащейся в базах данных информации и обеспечивающих её обработку информационных технологий и технических средств»*.

В этой книге мы часто будем использовать термин ИС в ещё более узком смысле, понимая под ***информационной системой*** (ИС) компьютерную сеть: совокупность программных, аппаратных и информационных ресурсов (данных), а также персонал, поддерживающий их согласованное функционирование. Мы включаем в понятие «информационная система» компьютерные сети самых различных типов: локальные и глобальные, беспроводные и виртуальные, частные и публичные, корпоративные и домашние. Отдельные компьютеры мы рассматриваем как частный случай сети и также относим их к ИС. Под ***информационными технологиями*** (ИТ) в книге понимаются компьютеризованные технологии хранения, передачи и обработки информации. Мы не включаем в ИС системы хранения и обработки материалов на бумажном носителе, телефонные коммуникации и видеоконференции, основанные на телефонных сетях, которые в широком смысле также являются информационными системами. В то

Федеральный закон Российской Федерации № 149-ФЗ «Об информации, информационных технологиях и о защите информации» (см. Приложение 1. Обзор нормативно-правовых актов РФ в области информационной безопасности).

же время некоторые принципы, понятия и процедуры, рассмотренные в книге, могут применяться и к информационным системам, определяемым в более широком смысле, т. е. к информационным системам, включающим и бумажный документооборот, — но о такой расширенной применимости читатель должен судить самостоятельно.

В фокусе нашего внимания будут ***информационные системы предприятия*** (ИСП). Каждая компьютерная сеть создана для решения тех или иных прикладных задач. Компьютерные сети являются непременной частью любого, сколько-нибудь значимого бизнеса, а также важной частью повседневной жизни современного человека. Нас окружают ИС самого разного назначения и разного масштаба, мы сталкиваемся с ними, когда приходим в банк или покупаем билеты на самолет, когда говорим по телефону, смотрим цифровое телевидение, устанавливаем охранную сигнализацию. И критически важным условием успешного бизнеса любого предприятия является качественная работа его информационной системы.

Информационная безопасность (ИБ) — защищённость информации, хранящейся и обрабатываемой ИС, а также инфраструктуры самой ИС (программных и аппаратных средств, персонала, систем электроснабжения и пр.) от случайных или преднамеренных воздействий естественного или искусственного характера, которые могут нанести неприемлемый ущерб субъектам информационных отношений. (В разделе «Модели информационной безопасности» будут даны более формальные определения информационной безопасности.)

Достижение информационной безопасности какого-либо объекта/системы осуществляется средствами ***системы обеспечения информационной безопасности*** (СОИБ) данного объекта/системы. СОИБ — это совокупность органов и (или) исполнителей, используемой ими техники защиты информационной системы, а также объектов защиты информационной системы, организованная и функционирующая по правилам и нормам, установленным соответствующими документами в области защиты информации. В качестве синонима СОИБ часто используется термин ***«система защиты информации»*** (СЗИ).

Система обеспечения информационной безопасности обычно представляет собой сложный конгломерат специализированных программно-аппаратных средств, включающих системы антивирусной защиты, прокси-серверы, разные типы файерволов, системы обнаружения и предотвращения вторжений, средства мониторинга трафика, фильтрующие маршрутизаторы, шлюзы виртуальных частных сетей, а также набор организационных мер.

Если каждое из этих средств использовать взятым по отдельности, то неизбежно будут возникать несогласованность процедур, противоречивость конфигурационных параметров, дублирование и бреши

в защите. Поэтому необходимым элементом СОИБ является ***система управления (менеджмента) информационной безопасностью*** (СУПБ), которая координированно управляет всеми применяемыми на предприятии защитными и организационными мерами, объединяя их в единый комплекс. СУИБ позволяет на постоянной основе отслеживать и анализировать работу системы обеспечения ИБ и вносить коррективы в её работу.

ИС как система контролируемого доступа к ресурсам

Концепция совместного использования ресурсов

Для пояснения некоторых базовых понятий информационной безопасности представим ИС в виде упрощенной модели контролируемого доступа, когда несколько пользователей совместно используют ресурсы информационной системы. Концепция разделения ресурсов является фундаментальной в вычислительной технике. Родившаяся полвека назад и направленная на повышение эффективности использования компьютера, эта концепция была реализована сначала в виде мультипрограммирования, где память, процессорное время и внешние устройства компьютера разделялись между процессами одного и того же пользователя. Затем появился многопользовательский режим, при котором ресурсы одной вычислительной системы совместно использовались несколькими пользователями. Все это выдвинуло проблемы безопасности вычислительных систем на первый план. Так как в многопользовательском режиме с компьютером работают сразу несколько человек, к тому же часто географически удаленных от компьютера, то возникла *необходимость контролировать доступ пользователей к компьютеру, защищая системные и пользовательские данные от ошибочных или злонамеренных действий.* Для решения этой задачи были созданы системы аутентификации и авторизации, которые проверяли легальность пользователей, а также ограничивали их только теми ресурсами и только теми операциями, которые им по тем или иным соображениям были разрешены. Контролируемый доступ является важным направлением обеспечения безопасности и в современных информационных системах, однако теперь к нему добавились и другие средства безопасности, такие как криптографическая защита, аудит, сегментация сети и др.

Итак, вернемся к модели контролируемого доступа. В этой модели (рис. 1.1) определены:

- объекты;
- субъекты;

Рис. 1.1. Модель контролируемого доступа

- операции, которые выполняются субъектами над объектами;
- система контроля доступа, которая решает, какие операции разрешены для данного субъекта по отношению к данному объекту.

Объекты представляют физические и логические информационные ресурсы ИС. К физическим ресурсам относятся как отдельные устройства целиком (процессор, внешние устройства, маршрутизаторы, коммутаторы, физические каналы связи и др.), так и физические разделяемые ресурсы устройств (разделы и сектора диска, процессорное время, физические соединения канала связи). Логическими ресурсами являются файлы, вычислительные процессы, сетевые сервисы, приложения, пропускная способности каналов связи и т. п.

Субъекты представляют сущности, между которыми разделяются информационные ресурсы. Это могут быть легальные пользователи ИС: персонал, поддерживающий работу ИС, внешние и внутренние клиенты; группы легальных пользователей, объединенные по различным признакам. Пользователь осуществляет доступ к объектам ИС не непосредственно, а с помощью прикладных процессов, которые запускаются от его имени. Поэтому в качестве субъектов выступают также прикладные вычислительные процессы. Иногда оказывается полезным представление в качестве субъектов и системных вычислительных процессов.

Для каждого типа объектов существует ***набор операций,*** которые (ними может выполнять субъект. Например, для файлов это операции чтения, записи, удаления, выполнения; для принтера — печать, перезапуск, очистка очереди документов, приостановка печати документа; для маршрутизатора — конфигурирование и т. д.

Система контроля доступа должна разрешать или запрещать субъектам выполнять ту или иную операцию по отношению к тому или иному объекту. Например, одному субъекту разрешена операция чтения и выполнения файла test.exe, а операция удаления запрещена. Другому субъекту разрешены все операции над файлом test.exe, а третьему — все операции запрещены.

Для автоматизированного контроля доступа необходимо, чтобы для каждой пары субъект-объект были однозначно определены ***правила доступа,*** на основании которых система могла бы принимать решение о разрешении или запрете выполнения каждой из предусмотренных для данного объекта операции. Такой селективный подход призван защитить ресурсы ИС от ошибочного или намеренно неправильного использования, другими словами, обеспечить безопасность ИС.

Важнейшими понятиями управляемого доступа являются идентификация, аутентификация и авторизация.

Идентификация

Все субъекты и объекты информационной системы должны иметь уникальные имена — ***идентификаторы,*** только при таком условии система получает возможность распознавать и оперировать субъектами и объектами. Одни идентификаторы автоматически генерируются ОС и приложениями (идентификаторы процессов, идентификаторы логических сетевых соединений), другие назначаются администратором компьютерной сети (идентификаторы пользователей, адреса компьютеров, доменные имена сетевых сервисов), третьи порождаются обычными сетевыми пользователями, обладающими таким правом (выбор собственного имени, назначение имен файлам).

Пользователь может быть представлен в системе в виде нескольких субъектов и соответственно иметь несколько пользовательских идентификаторов. Например, иметь один идентификатор, который он использует во время сетевой регистрации, другой идентификатор — для доступа к электронной почте, третий — для работы с корпоративной базой данных, четвертый — для просмотра веб-сайта. Такая практика ослабляет защищенность системы, более современной является концепция ***единого входа в систему,*** которая будет рассмотрена в главе 5, посвященном технологиям аутентификации.

Идентификация пользователей является одной из обязательных процедур, выполняемых при логическом входе в систему, когда пользователь в ответ на выведенное на экране приглашение печатает свой идентификатор-имя, а система, сверяясь со своими данными, определяет, входит ли данное имя в число имен зарегистрированных (легальных) пользователей.

Аутентификация

Термин «аутентификация» (authentication) происходит от латинского слова ***authenticus,*** которое означает подлинный, достоверный, соответствующий самому себе.

Рис. 1.2. «В Интернете никто не узнает, что ты собака, если успешно пройдешь аутентификацию» (рисунок Питера Штайнера)

Аутентификация — это процедура доказательства субъектом/объектом того, что он есть то, за что (кого) он себя выдает (рис. 1.2). Аутентификация, или, другими словами, процедура установления подлинности может быть применима как к пользователям, так и другим объектам и субъектам, в частности к данным, программам, приложениям, устройствам, документам.

В процедуре аутентификации участвуют две стороны:

- ***аутентифицируемый*** — доказывает свою аутентичность, предъявляя некоторое доказательство — ***аутентификатор,***
- ***аутентифицирующий*** — проверяет эти доказательства и принимает решение. Аутентификация бывает односторонней и двусторонней (взаимной).

Так, мы имеем дело с ***односторонней аутентификацией,*** в частности, при выполнении логического входа в защищенную систему. После того как пользователь сообщает системе свой идентификатор, он должен пройти процедуру аутентификации, т. е. доказать, что именно ему принадлежит введенный им идентификатор (имя пользователя). Аутентификация предотвращает доступ к сети нежелательных лиц и разрешает вход для легальных пользователей.

В качестве аутентификатора аутентифицируемый может продемонстрировать:

- знание некоего общего для обеих сторон секрета: слова (пароля) или факта (даты и места некоторого события, прозвища человека и т. п.);
- обладание неким уникальным предметом, в качестве которого может выступать, например, физический ключ или электронная магнитная карта;
- собственные биохарактеристики: рисунок радужной оболочки глаза или отпечатки пальцев. В некоторых случаях односторонняя аутентификация оказывается недостаточной и тогда используют

двустороннюю аутентификацию. Например, пользователь, обращающийся с запросом к корпоративному веб-серверу, должен доказать ему свою легальность, но он также должен убедиться сам, что ведёт диалог действительно с веб-сервером своего предприятия. Другими словами, сервер и клиент должны пройти процедуру взаимной аутентификации. Здесь мы имеем дело с двусторонней аутентификацией на уровне приложений. При установлении сеанса связи между двумя устройствами также часто предусматриваются процедуры взаимной аутентификации устройств на более низком, канальном, уровне.

Аутентификация данных означает доказательство целостности этих данных, а также то, что они поступили именно от того человека, который объявил об этом. Для этого используется механизм ***электронной подписи.***

А вторизация

Термин «авторизация» (authorization) происходит от латинского слова ***auctoritas,*** показывающее уровень престижа человека в Древнем Риме и соответствующие этому уровню привилегии.

Авторизация — это процедура контроля доступа субъектов (пользователей, вычислительных процессов, устройств) к объектам (например, файлам, приложениям, сервисам, устройствам) и предоставления каждому из них именно тех прав, которые им определены правилами доступа.

Рассмотрим процесс авторизации для случая, когда субъекты представляют пользователей системы. В отличие от аутентификации, которая позволяет распознать легальных и нелегальных пользователей, авторизация имеет дело только с ***легальными*** пользователями, успешно прошедшими процедуру аутентификации. Помимо предоставления пользователям прав доступа к каталогам, файлам и принтерам, средства авторизации могут контролировать возможность выполнения пользователями различных системных функций, таких как локальный доступ к серверу, установка системного времени, создание резервных копий данных, выключение сервера и т. п.

Доступ к объектам, полученный в обход разрешений системы контроля доступа, называется ***несанкционированным***

Модели информационной безопасности

Наряду со стандартным (и несколько тавтологичным) определением информационной безопасности как состояния защищенности, которое было дано в начале этого раздела, существуют более

формальные определения информационной безопасности, называемые ***моделями безопасности.*** Все эти модели построены по одному принципу, который заключается в следующем. Множество всех видов нарушений безопасности делится на несколько базовых групп таким образом, чтобы любое нарушение обязательно могло быть отнесено по крайней мере к одной из этих групп. Затем система объявляется безопасной, если она способна противостоять каждой из этих групп нарушений.

Триада «Конфиденциальность, доступность, целостность»

Одной из первых и наиболее популярных по сей день моделей безопасности является модель, предложенная Зальцером и Шредером*.

Авторы постулировали, что все возможные нарушения информационной безопасности всегда могут быть отнесены по меньшей мере к одной из трех групп: нарушения конфиденциальности, нарушения целостности или нарушения доступности (рис. 1.3).

Рис. 1.3. Триада «конфиденциальность, целостность, доступность»

Соответственно информационная система находится в *состоянии* ***безопасности,*** если она защищена от нарушений конфиденциальности, целостности и доступности, где:

конфиденциальность (confidentiality) — это состояние ИС, при котором информационные ресурсы доступны только тем пользователям, которым этот доступ разрешён;;

целостность (integrity) — это состояние системы, при котором информация, хранящаяся и обрабатываемая этой ИС, а также процедуры обработки информации не могут быть изменены, удалены или дополнены неавторизованным образом; ***доступность*** (availability) — это состояние системы, при котором услуги, оказываемые системой, могут гарантированно, с приемлемой задержкой быть предоставлены пользователям, имеющим на это право. Для ссылки на триаду — конфиденциальность, целостность, доступность — иногда используют аббревиатуру КЦД или, в англоязычной форме, CIA.

Jerry H. Saltzer, Mike D. Schroeder (September 1975), ***«The protection of information in computer systems».***

Требования к безопасности могут меняться в зависимости от назначения информационной системы, характера используемых данных и типа возможных угроз. Трудно представить систему, для которой нарушения целостности и доступности не представляли бы опасности, вместе с тем обеспечение конфиденциальности не всегда является обязательным.

Например, если вы публикуете информацию в Интернете на вебсервере и вашей целью является сделать ее доступной для самого широкого круга людей, то конфиденциальность не требуется. Однако требования целостности и доступности остаются актуальными. Действительно, если вы не предпримете специальных мер по обеспечению целостности системы, то злоумышленник может изменить данные на вашем сервере и нанести этим ущерб вашему предприятию. Преступник может, например, внести изменения в помещенный на веб-сервере прайс-лист, что негативно отразится на конкурентоспособности вашего предприятия, или испортить коды свободно распространяемого вашей фирмой программного продукта, что, безусловно, скажется на ее деловой репутации. Если бы модифицированные данные были к тому же секретными, то в таком случае мы бы имели не только нарушение целостности, но и конфиденциальности.

Не менее важным в данном примере является и обеспечение доступности данных. Затратив немалые средства на создание и поддержание сервера в Интернете, предприятие вправе рассчитывать на отдачу: увеличение числа клиентов, количества продаж и т.д. Однако существует вероятность того, что злоумышленник предпримет атаку, в результате которой помещённые на сервер данные станут недоступными для тех, кому они предназначались. Примером таких злонамеренных действий может служить «бомбардировка» сервера пакетами, каждый из которых в соответствии с логикой работы соответствующего протокола вызывает тайм-аут сервера, что в конечном счете делает его недоступным для всех остальных запросов.

Некоторые виды нарушений безопасности могут быть приведены к модели КДЦ только путем расширительного толкования основополагающих понятий конфиденциальности, доступности и целостности. Так, свойство конфиденциальности по отношению, например, к устройству печати можно интерпретировать так, что доступ к устройству имеют те и только те пользователи, которым этот доступ административно разрешен, причем они могут выполнять только те операции с устройством, которые для них определены. Свойство доступности устройства означает его готовность к работе всякий раз, когда в этом возникает необходимость. А свойство целостности может быть интерпретировано как свойство неизменности параметров данного устройства.

За 40 лет, прошедших с момента публикации статьи Зальцера и Шредера, информационные системы и среда, в которой они функционируют, претерпели революционные изменения, неудивительно, что появились новые типы нарушений, которые стало намного труднее (если вообще возможно) трактовать в терминах КДЦ. Рассмотрим, например, ситуацию, когда легальный клиент банка посылает по электронной почте запрос на снятие со счета крупной суммы, а затем заявляет, что этот запрос, который хотя и был послан от его имени, он не отправлял. Является ли это нарушением безопасности? Да. Были ли при этом нарушены конфиденциальность, доступность или целостность? Нет. Следовательно, список свойств безопасной системы следует расширить, добавив к КЦД еще одно свойство — « неотказуемость»:

Неотказуемость (non-repudiation) — состояние системы, при котором обеспечивается невозможность отрицания пользователем, выполнившим какие-либо действия, факта их выполнения, в частности отрицания отправителем информации факта ее отправления и/или отрицания получателем информации факта ее получения.

Гексада Паркера и модель STRIDE

Дискуссии о том, какой набор свойств И С исчерпывающе характеризует ее безопасность, продолжаются, в результате предлагаются все новые и новые модели безопасности.

Одной из наиболее популярных альтернатив триаде КДЦ является так называемая *гексада Паркера** (Parkerian Hexad), в которой определены шесть базовых видов нарушений, в число которых, помимо традиционных нарушений конфиденциальности, доступности и целостности, входят еще три вида нарушений (рис. 1.4).

- Нарушения аутентичности/подлинности/достоверности. ***Аутентичность*** (authenticity) — это состояние системы, при котором пользователь не сможет выдавать себя за другого, а документ всегда имеет достоверную информацию о его источнике (авторе). Из этого определения видно, что аутентичность является аналогом неотказуемости.

Дон Паркер предложил свою гексаду в работе ***Fighting Computer Crime***

МЩ0

Нарушения владения/контроля. ***Владение*** (possession) — это состояние системы, при котором физический контроль над устройством или другой средой, на которой хранится информация, предоставляется только тем, кто имеет на это право.

Нарушения полезности. ***Полезность*** (utility) — это такое состояние ИС, при котором обеспечивается удобство практического

("MUHAMMAD AL-XORAtMSIY NOMIDAGI j

(1998).

Рис. 1.4. Гексада Паркера

Spuffing	Подмена
Tampering	Изменение данных
Repudiation	Отказ от ответственности
Information disclosure	Разглашение сведений
Denial of service	Отказ в обслуживании
Elevation of privilege	Захват привилегий

Рис. 1.5. **Модель STRIDE**

использования как собственно информации, так и связанных с ее обработкой и поддержкой процедур. В безопасной системе меры, предпринимаемые для защиты системы, не должны неприемлемо усложнять работу сотрудников, иначе они будут воспринимать их как помеху и пытаться при всякой возможности их обойти.

Еще одним вариантом определения безопасности ИС является модель STRIDE1 (аббревиатура от англоязычных названий типов нарушений безопасности, перечисленных ниже). В соответствии с этой моделью (рис. 1.5) ИС находится в безопасности, если она ***защищена*** от следующих типов нарушений:

- ***подмена данных*** (Spoofing) —нарушение, при котором пользователь или другой субъект ИС путем подмены данных таких, например, как IP-адрес отправителя, успешно выдает себя за другого, получая таким путем возможность нанесения вреда системе;
- ***изменение*** (Tampering) — нарушение целостности;
- ***отказ от ответственности*** (Repudiation) —негативная форма уже рассмотренного нами свойства неотказуемости (non-repudiation);
- ***разглашение сведений*** (Information Disclosure) — нарушение конфиденциальности;
- ***отказ в обслуживании*** (Denial of Service) — нарушение доступности;
- ***захват привилегий*** (Elevation of Privilege) — нарушение, заключающееся в том, что пользователь или другой субъект ИС несанкционированным образом повышает свои полномочия в этой системе, в частности незаконное присвоение злоумышленником

1 Модель STRIDE используется компанией Microsoft в разработках безопасного программного обеспечения.

прав сетевого администратора, которое снимает практически все защитные барьеры на его пути.

В модели STRIDE все возможное разнообразие нарушений безопасности сводится также, как и в гексаде Паркера, к шести типам нарушений, три из которых повторяют КДЦ (с учетом того, что здесь эти три характеристики безопасности даны в негативном по отношению к КДЦ варианте), однако оставшиеся три — подмена данных, отказ от ответственности и захват привелегий — являются отличными от гексады Паркера.

Российский государственный стандарт* дает определение информационной безопасности на основе гексады Паркера:

«Информационная безопасность — [это] все аспекты, связанные с определением, достижением и поддержанием конфиденциальности, целостности, доступности, неотказуемости, подотчетности, аутентичности и достоверности информации или средств ее обработки».

Уязвимость, угроза, атака, ущерб

Уязвимость (vulnerability) — это слабое звено информационной системы, которое, став известным злоумышленнику, может позволить ему нарушить ее безопасность.

Уязвимостями являются, например, ошибка в программе, примитивный пароль, неправильное назначение прав доступа к файлу с важными данными и множество других дефектов в разработке, эксплуатации или настройке системы.

Уязвимости системы могут быть скрытыми, т. е. еще не обнаруженными, они могут быть известными, но только теоретически, или же общеизвестными и активно используемыми злоумышленниками. Для общеизвестных уязвимостей в программных продуктах производители регулярно выпускают исправления, называемые ***патчами*** (patch — заплатка). Так, компания Microsoft даже назначила специальный день — каждый второй вторник каждого месяца, когда она объявляет о новых исправлениях в семействе ОС Windows. Многие из этих исправлений направлены на устранение уязвимостей. Однако к этой рутинной процедуре — регулярному внесению исправлений не все и не всегда относятся с должным вниманием, из-за этого общеизвестные, но неисправленные ошибки в программном обеспечении являются одним из самых распространённых типов уязвимостей.

ГОСТ 13335-1-2006 «Информационная технология. Методы и средства обеспечения безопасности. Часть 1. Концепция и модели менеджмента безопасности информационных и телекоммуникационных технологий».

Другой тип уязвимостей, которыми часто пользуются злоумышленники, это ошибки в конфигурировании программных и аппаратных средств. Например, имена «администратор» и «гость», установленные по умолчанию во многих ОС, могут облегчить злоумышленникам доступ к системе, поэтому они должны быть сразу, при начальном конфигурировании ОС, заменены на другие, менее очевидные имена. С этой же целью администратор должен настроить подсистему интерактивного входа на то, чтобы она не отображала последнего имени пользователя, систему аудита — чтобы фиксировала все успешные и неуспешные попытки входа пользователей, а также выполнить другие столь же простые, но необходимые настройки.

Поиск уязвимостей — важная часть задачи обеспечения безопасности. Эта работа включает в себя регулярное тестирование системы с привлечением программных инструментов. Существующие многочисленные программные средства обнаружения уязвимостей не требуют от пользователя особой квалификации, как правило, они выдают на выходе довольно длинный перечень потенциальных брешей в защите системы. Однако без включения в этот процесс человека, профессионала, обладающего знаниями обо всех аспектах функционирования системы, трудно рассчитывать на успех.

Другими базовыми понятиями информационной безопасности являются угроза, атака и ущерб.

Угроза (***threat***) — набор обстоятельств и действий, которые потенциально могут привести к нарушению безопасности системы (т. е. к нарушению ее конфиденциальности, целостности и доступности, если пользоваться моделью КДЦ).

Атака (***attack***) — реализованная угроза.

Ущерб (loss, impact) — негативное влияние на систему, оказываемое проведённой атакой.

Мы в основном ограничимся рассмотрением только *технических* угроз, т. е. угроз, исходящих из искусственно созданного человеком мира техники и технологий (в частности, из Интернета), не принимая во внимание угрозы, возникающие от природных катаклизмов, военных действий, террористических атак или экономических потрясений. К числу технических угроз относятся, например, неправильное использование и утрата данных, ошибки в работе программ, неисправность оборудования.

Атака может произойти только тогда, когда одновременно существуют уязвимость и направленная на использование этой уязвимости угроза (рис. 1.6). То есть вполне возможна ситуация, когда система имеет некую уязвимость, но эта уязвимость еще не стала известной злоумышленникам, т. е. в данном случае соответствующая угроза отсутствует, а значит, и атака не может быть проведена. Аналогично,

существование общеизвестной угрозы не влечет никакой опасности для системы, в которой нет соответствующей уязвимости. Например, существование общеизвестной угрозы протоколу TCP в реализации Microsoft не представляет опасности для ИС, использующих ОС Unix.

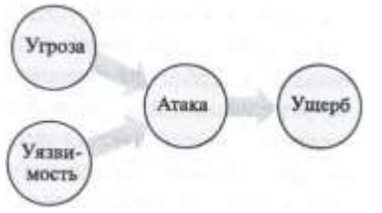


Рис. 1.6. Логическая связь между понятиями «уязвимость», «угроза», «атака», «ущерб»

Таким образом, любая угроза направлена на поиск и/или использование уязвимостей системы. В некоторых случаях злоумышленник работает наощупь, пытаясь обнаружить тот или иной дефект системы. Система реагирует на такого рода угрозы выдачей сообщений о мелких, но странных неполадках, а также флуктуациями в статистических характеристиках работы системы, на основании которых администратор сети или специалист по безопасности может заподозрить подготовку атаки.

Другие угрозы выражаются в чёткой последовательности действий и имеют формализованное воплощение в виде ***эксплойта*** (***exploit***) — программы или просто последовательности командных строк, некоторой порции данных и/или пошаговом описании действий, которые, будучи выполненными, позволяют злоумышленнику воспользоваться некоторой конкретной уязвимостью информационной системы в своих интересах. Особая опасность эксплойта состоит в том, что, имея его в своем распоряжении, даже малоподготовленный хакер способен провести успешную атаку. Для этого ему достаточно зайти на один из многочисленных сайтов, снабжающих всех желающих своей «продукцией». Более того, в придачу к инструкциям и программам в Интернете можно найти даже предложения о сдаче в аренду целых ***бот-сетей2,*** готовых к реализации мощных кибер-атак. С другой стороны наличие у эксплойтов фиксированных признаков, таких, например, как специфические кодовые последовательности, облегчают распознавание и отражение соответствующих атак.

Угрозы могут исходить как от легальных пользователей сети, так и от внешних злоумышленников. В последние два года в статистике нарушений безопасности зафиксирован резкий сдвиг от внешних к внутренним угрозам. Примерно 2/3 от общего числа всех наиболее серьёзных инцидентов, связанных с безопасностью, составляют нарушения или ошибки со стороны легальных пользователей сетей:

2 Бот-сеть, или ботнет (botnet) — организованная совокупность компьютеров, связанных через Интернет, способных согласованно выполнять задачи, поставленные перед ними злоумышленником.
http://www.merit.edu/mail.archives/nanog/1997-04/msg00444.html

сотрудников и клиентов предприятий, студентов, имеющих доступ к сети учебного заведения, и др. Внутренние атаки обычно наносят меньший ущерб, чем внешние.

Угрозы внешних злоумышленников, называемых ***также хакерами,*** по определению являются умышленными и обычно квалифицируются как преступления. Среди внешних нарушителей безопасности встречаются люди, занимающиеся этой деятельностью профессионально или просто из хулиганских побуждений. Целью, которой руководствуются внешние злоумышленники, всегда является нанесение ущерба предприятию. Это может быть, например, получение конфиденциальных данных, которые могут быть использованы для снятия денег с банковских счетов, или установление контроля над программноаппаратными средствами сети для последующего их использования в атаках на сети других предприятий. Ущерб может быть и сугубо моральный — хакер может просто захотеть показать свое профессиональное превосходство над администратором якобы хорошо защищенной сети.

Угрозы со стороны легальных пользователей могут быть как умышленными, так и неумышленными.

К ***умышленным*** угрозам относится, например, мониторинг системы с целью получения персональных данных других сотрудников (идентификаторов, паролей) или конфигурационных параметров оборудования. Это может быть также злонамеренное получение доступа к конфиденциальным данным, хранящимся на серверах и рабочих станциях сети «родного» предприятия с целью их похищения, искажения или уничтожения; прямое «вредительство» — вывод из строя сетевого программного обеспечения и оборудования. Кроме того, к умышленным угрозам относится нарушение персоналом правил, регламентирующих работу пользователей в сети предприятия: посещение запрещенных веб-сайтов, вынос за пределы предприятия съемных носителей, небрежное хранение паролей, и другие подобные нарушения режима.

Однако не меньший материальный ущерб предприятию может быть нанесен в результате ***непреднамеренных*** нарушений пользователей и обслуживающего персонала — ошибок, приводящих к повреждению сетевых устройств, данных, программного обеспечения, ОС и приложений, беспечность в обеспечении секретности паролей и др.

Известно, что правильное конфигурирование устройств является одним из мощных инструментов обеспечения безопасности. Но будучи выполненной с ошибками, эта операция может обернуться своей противоположностью — угрозой. Как выяснилось, некоторые «атаки» на ИС были на самом деле не атаками, а ошибками администраторов сетей при выполнении конфигурирования элементов системы.

Например, широко известен случай неверного конфигурирования протокола маршрутизации BGP в сети клиента провайдера AS7007, который привел к отказам работы большой части Интернета в 1997 году*. Наверно, каждый пользователь корпоративной системы сможет припомнить случай, когда после конфигурирования маршрутизаторов или программного обеспечения серверов в сети что-то стало работать «не так» или вообще не работать — принтер

перестал принимать задания на печать или почта от определенных адресатов перестала доставляться. Поэтому очень важно, чтобы интерфейс, которым пользуется администратор или пользователь при конфигурировании устройства или программы, был понятным, исключал неоднозначность трактовок вводимых параматеров, направлял работу пользователя в нужное русло и предотвращал наиболее типичные и тяжелые по последствиям ошибки.

Типы и примеры атак

Пассивные и активные атаки

Атаки разделяют на активные и пассивные.

Активные атаки включают явные воздействия на систему, изменяющие ее состояние. Это могут быть зловредный программный код- вирус, внедрённый в исполняемую системой программу, или искажения данных на страницах взломанного веб-сайта, блокировка сетевого сервиса путем бомбардировки его ложными запросами или внедренное в коммуникационный протокол ложное сообщение. Главной отличительной чертой активных атак является то, что после своего завершения они, как правило, оставляют следы.

Многие активные кибер-атаки относят к типу ***«взламывание»*** (***breaking-in***), проводя аналогию с бытовыми ограблениями со взломом, когда хозяин заходит в свой дом и сразу обнаруживает повреждённые замки, опустошённые ящики и разбросанные на полу вещи. В компьютерной системе после активного проникновения злоумышленника тоже остаются следы «взлома», например изменяется содержимое памяти, поступают странные диагностические сообщения, приложения начинают выполняться неправильно, замедленно или вообще зависают, в характеристиках сетевого трафика и в других статистических данных о работе системы появляются необъяснимые всплески активности.

Заметим, что иногда грабитель так хорошо «заметает следы», что пострадавший может сразу и не заметить преступления, особенно если он не обладает наблюдательностью Шерлока Холмса или Эр-

кюля Пуаро. Так и в информационной системе тщательно подготовленная активная атака может пройти незамеченной, если специалисты, отвечающие за ее безопасность, плохо осведомлены о возможных последствиях такого рода атак.

Пассивные атаки не нарушают нормальной работы ИС, они связаны со сбором информации о системе, например прослушиванием внутрисетевого трафика или перехватом сообщений, передаваемыхм по линиям связи. Во многих случаях пассивные атаки не оставляют следов, поэтому их очень сложно выявить, часто они так и проходят незамеченными. Если использовать военную аналогию, то это разведка (но не боем).

Противопоставление активной и пассивной формы атак является некоторой идеализацией. На практике мы редко имеем дело с активной или пассивной атакой «в чистом виде». Чаще всего атака включает подготовительный этап сбора информации об атакуемой системе, а затем на основе собранных данных осуществляется активное вмешательство в ее работу. Сбор информации о системе помогает не только эффективно спланировать атаку, но и скрыть все следы проникновения в систему. К полезной для хакера информации относятся типы операционных систем и приложений, развёрнутых в сети, IP-адреса, номера портов клиентских частей приложений, имена и пароли пользователей. Часть информации такого рода может быть получена при анализе открытой информации или простом общением с персоналом (это называют ***социальным инжинирингом***), а часть — с помощью тех или иных программ. В последнем случае мы сталкиваемся с другой последовательностью этапов: сначала выполняется активная фаза внедрения на атакуемый компьютер подслушивающей программы, затем период пассивного сбора информации (например, паролей пользователей), а затем снова активная фаза проникновения в компьютер.

Сейчас мы коротко, ограничиваясь обсуждением общей идеи, рассмотрим несколько типов популярных атак: отказ в обслуживании, спуфинг, внедрение кода, кража личности, фишинг, сетевая разведка. Более подробно эти, а также иные типы атак описаны в других разделах.

Отказ в обслуживании

К числу активных атак относятся две весьма распространенные атаки: «Отказ в обслуживании (Denial of Service, DoS) и «Распределенная атака отказа в обслуживании» (Distributed Denial of Service, DDoS).

Смысл атаки « ***Отказ в обслуживании»*** прямо следует из ее названия. Система, предназначенная для выполнения запросов легальных

пользователей, вдруг перестает это делать или делает это с большими задержками, что эквивалентно отказу. Очевидный пример такой системы — веб-сайт. Наверняка, 17 млн британских болельщиков Энди Марри «обрушили» бы сайт BBC, если бы трансляция финального теннисного матча Уимблдона в 2013 году шла только в Интернете (к счастью, параллельно шла телевизионная передача). Такие всплески запросов являются экстраординарными, и правильно спроектированные серверы справляются с нагрузкой, на которую они рассчитаны. Однако отказ в обслуживании может наступить не только в результате редкой флюктуации интенсивности запросов, но и в результате злонамеренных действий, когда перегрузка создается искусственно, а именно, на атакуемый компьютер посылается интенсивный поток запросов, сгенерированных средствами атакующего компьютера (рис. 1.7).

Рис. 1.7. Схема DDoS-атаки

Этот поток «затопляет» атакуемый компьютер, вызывая его перегрузку, и в конечном счете делает его недоступным. Блокировка происходит в результате исчерпания ресурсов либо процессора, либо операционной системы, либо канала связи (полосы пропускания). DoS-атака использует тот простой факт, что компьютер подключен к сети — именно это является в данном случае уязвимостью. К сожалению для большинства современных пользователей устранить эту уязвимость просто отключением компьютера от Интернета нельзя, хотя в некоторых случаях, требующих особо высокого уровня безопасности, так и поступают.

Злоумышленник может многократно усилить эффект от проведения атаки «Отказ в обслуживании» путем кражи чужой вычислительной мощности. Для этого он получает контроль над атакуемым компьютером, загружает в него вредительское программное обеспечение и активирует его. Таким образом злоумышленник незаметно от владельца «ответвляет» часть вычислительной мощности, заставляя компьютер работать на себя. При этом владельцу компьютера не наносится никакого другого вреда, кроме снижения производительности его компьютера. Для проведения мощной атаки злоумышленник захватывает контроль над некоторым множеством компьютеров, организует их согласованную работу и направляет суммарный многократно усилившийся поток запросов с множества копьютеров-«зомби» на компьютер-жертву. Говорят, что в таких случаях имеет место ***распределенная атака отказа в обслуживании,*** или DDoS-атака.

При проведении атак злоумышленнику важно не только добиться своей цели, заключающейся в причинении ущерба атакуемому объекту, но и уничтожить все следы своего участия в этом. Одним из основных приёмов, используемых злоумышленниками для «заметания следов», является ***подмена содержимого пакетов*** (***спуфинг***; ***spoofing).*** В частности, для сокрытия места нахождения источника вредительских пакетов (например, при атаке отказа в обслуживании) злоумышленник изменяет значение поля адреса отправителя в заголовках пакетов. Поскольку адрес отправителя генерируется автоматически системным программным обеспечением, злоумышленник вносит изменения в соответствующие программные модули так, чтобы они давали ему возможность отправлять со своего компьютера пакеты с любыми IP-адресами. Еще труднее определить адрес источника распределенной атаки, так как непосредственными исполнителями выступают «зомбированные» компьютеры и именно их адреса содержатся в поле «адрес отправителя» пакетов, бомбардирующих компьютер- жертву. И хотя ничего не подозревающие владельцы компьютеров- непосредственных исполнителей стали участниками распределенной атаки помимо своей воли, большая часть ответственности ложится и на них. Ведь именно их недоработки в деле обеспечения безопасности собственных систем сделали возможной эту атаку.

Внедрение вредоносных программ

Многочисленная группа активных атак связана с внедрением в компьютеры ***вредоносных программ (malware***, сокращение от ***malicious software).*** К этому типу атак относятся (рис. 1.8) троянские и шпионские программы, руткит, черви, вирусы, спам, логические бомбы и др.

Эти программы могут проникать на атакуемые компьютеры разными путями. Самый простой из них — «самодос- тавка», когда пользователь загружает файлы из непроверенных источников (съемных носителей или веб-сайтов) либо беспечно открывает подозрительный файл, пришедший к нему как приложение по электронной почте. Существуют и более сложные представители вредоносных программ, обладающие собственными механизмами «размножения», копии таких программ распространяются по компьютерам сети без участия пользователей.

Рис. 1.8. Вредоносные программы

Одним из примеров вредоносных программ являются шпионские программы (spyware), которые тайно (как правило, удаленно) устанавливаются злоумышленниками на компьютеры ничего не подозревающих пользователей, чтобы отслеживать и фиксировать все их действия. В число таких действий может входить введение имени и пароля во время логического входа в систему, посещение тех или иных вебсайтов, обмен информацией с внешними и внутренними пользователями сети и прочее, и прочее. Собранная информация пересылается злоумышленнику, который применяет ее в преступных целях.

Заметим, что в качестве шпионских программ могут использоваться не только созданные специально для этих целей вредоносные программы, но и программы легального назначения. Так, опасным средством шпионажа могут стать легальные системы мониторинга сети, такие, например, как популярные сетевые мониторы Wireshark или Microsoft Network Monitor. Исходное назначение этих программ состоит в том, чтобы дать администратору сети возможность следить за сетевым трафиком, в частности, захватывать пакеты, используя механизм фильтрации, просматривать их содержимое, собирать статистику по загрузке устройств. В руках злоумышленника такая программа превращается в мощный инструмент взлома сети, который позволяет перехватывать пакеты с паролями и другой секретной информацией. Они также позволяют путем сканирования TCP- и UDP-портов определять типы приложений, работающих в сети, что является очень важной информацией для подготовки многих типов атак.

Потери, вызванные вредоносными программами, могут заключаться не только в уничтожении, искажении или похищении информации, приведении в нерабочее состояние программного обеспечения, а значит, и компьютера в целом, но и в значительных затратах времени и сил администраторов на обнаружение и распознавание атак, фильтрацию внешних сообщений, тестирование и перезагрузку систем. Вредоносные программы в начале этого десятилетия были одной из основных причин нарушения безопасности компьютерных сетей. Однако, как показала статистика, в последние два года суммарный ущерб, нанесенный вредоносными программами предприятиям, резко снизился. Это связывают в том числе с улучшением качества антивирусных средств и ужесточением наказаний за такого рода преступления.

Красна личности, фишинг

Пассивные атаки по своей природе не выглядят настолько опасными как активные, а на самом деле они могут быть более опасными. Практически для проведения любой активной атаки требуются данные о системе, на которую готовится нападение. Если такой системой является отдельный человек, то сбор данных называют ***кражей личности (identity theft).***

По мере развития услуг, оказываемых через Интернет, все более популярными становятся аферы, когда один человек выдает себя за другого. Действительно, ведь в этом случае не требуется личное присутствие в офисе и индивидуум доказывает свою идентичность, передавая обслуживающему центру свои персональные данные по телефону или используя интерактивную систему веб-сайта. Злоумышленник решает выдать себя за другого для того, например, чтобы взять кредит на чужое имя, получить доступ к чужому счету, рассчитаться за покупку чужой карточкой, получить именное приглашение на закрытое мероприятие.

Итак, пусть, например, преступник планирует воспользоваться чужим банковским счетом. Для того чтобы он мог выдать себя за другого, ему предстоит узнать достаточно много о потенциальной жертве: как минимум, ему понадобятся все паспортные данные и банковские реквизиты. Существует множество различных способов, с помощью которых злоумышленник добывает персональные данные. Например, очень многое он может узнать, просто просматривая мусорную урну, куда беспечные люди часто выбрасывают уже ненужные им, но чрезвычайно «полезные» для преступника бумаги. Казалось бы, что секретного можно найти в конверте с адресами и именами отправителя и получателя, квитанции, рецепте, использованном билете или распечатке банковского отчёта, однако в совокупности эти данные могут позволить злоумышленнику подменить вашу личность.

Фишинг (phishing — искаженное ***fishing***) — это ещё один приём, используемый мошенниками для «выуживания» персональных данных. Вы можете ответить на телефонный звонок, и человек, представившийся сотрудником банка или государственной налоговой службы, работником ЖКХ или представителем провайдера мобильной связи, начинает выспрашивать критичные данные о вас. Угроза может прийти и по электронной почте. Будущая жертва получает сообщение, в котором, к примеру, сообщается о якобы произведенной ею покупке, как правило, достаточно дорогой. Далее говорится, что при снятии средств за эту покупку у банковской системы возникли некие проблемы. Для разрешения ситуации клиенту предлагается срочно пройти по ссылке на сайт банка. Жертва, взволнованная тем, что никакой такой покупки она не совершала, торопится прояснить ситуацию, кликает по предложенной ссылке и видит на экране знакомый логотип своего банка и интерактивную форму, запрашивающую персональные данные клиента, ИНН, номер счета, девичью фамилию матери и другие данные, которые нужны злоумышленнику.

В других случаях в сообщении электронной почты клиенту предлагается позвонить по указанному в сообщении телефонному номеру, который якобы является телефоном клиентской службы его банка.

Набрав номер (и не убедившись, что это действительно телефон банка), клиент слышит вопросы автоматизированной голосовой системы, которая выполняет обычную и знакомую ему по предыдущим обращениям в банк процедуру аутентификации клиента, в ходе которой клиент сообщает номер счета, карты, ПИН-кода, пароля доступа к счету. Увы, но клиент имеет дело с поддельной «системой», и все данные, которые он сообщает, попадают к злоумышленнику.

Чем больше людей узнает о приемах выуживания информации, тем более изощренные методы обмана используют преступники. Они создают поддельные сайты, выглядящие совсем как настоящие, они используют доменные имена, очень похожие на настоящие. Когда вам предлагают пойти на сайт международной платежной системы PayPal, а в адресной строке браузера появляется адрес www.peypal.com, вы можете и не заметить подмены. Когда же наученные горьким опытом пользователи Интернета стали более внимательны, мошенники научились, используя несовершенства браузеров, помещать в поле адресной строки браузера имя настоящего сайта, в нашем случае — paypal.com. В такой ситуации даже самый внимательный пользователь мог потерять бдительность и перейти на подставной сайт. И хотя эта уязвимость браузера была вскоре устранена, расслабляться нельзя — преступники продолжают совершенствовать свои приемы фишинга.

Следующим изобретением стали всплывающие окна. Пусть клиент получает доступ к сайту своего банка (действительному, не поддельному) в результате прохождения стандартной процедуры идентификации и аутентификации. Он просматривает страницы сайта, нет никаких сомнений, что это реальный сайт. В какой-то момент на экране появляется всплывающее окно, которое стилистически выглядит как неотъемлемая часть сайта. В этом окне размещена интерактивная форма, запрашивающая персональные данные. Клиент чувствует себя в полной безопасности и вводит все запрашиваемые критичные данные. Однако настоящий сайт банка является только фоном, на котором располагаются окна-ловушки злоумышленника.

Сетевая разведка

К этому типу относятся атаки, направленные на сбор информации о компьютерной системе, которая позже, на основании собранных данных, может быть подвергнута другой атаке — взлому. Понятно, что не каждая конкретная атака пригодна для нападения на любую конкретную систему, и ноутбук с установленными на нем Windows 8 и браузером Internet Explorer подвержен совсем другой группе рисков, нежели Ipad с ОС iOS6 и браузером Safari. Поэтому так важно для преступника провести разведку и узнать «имена, пароли, явки», или

в данном случае — «адреса компьютеров, версии ОС, порты приложений».

Для этих целей злоумышленники выполняют ***сканирование*** системы, т. е. с помощью специальных программ пытаются направить в исследуемую систему запросы, формат которых может оказаться совместимым с форматом сообщений, используемым тестируемой системой. Получив запрос, атакуемая система может принять его за обычное протокольное сообщение и отправить ответ, из которого злоумышленник извлечёт нужную ему информацию. Например, атака Christmas Tree (рождественская елка), предназначенная для определения типа операционной системы, состоит в том, что на тестируемую систему посылается TCP-пакет со всеми установленными признаками в заголовке пакета (заголовок, «обвешенный гирляндами»). Такой формат TCP-пакета вызывает различную реакцию у различных ОС, и простейший анализ ответного пакета приводит преступника к решению поставленной задачи. Другой популярной формой разведки является атака «Сканирование портов», когда на компьютер направляется последовательность TCP- и UDP-запросов для того, чтобы по ответам системы выяснить, какие порты операционной системы открыты, и тем самым узнать, какие приложения в системы установлены и активны. Так как многие вирусы, подобно обычным приложениям, используют для взаимодействия с внешним миром вполне определенные порты, то сканирование портов позволяет злоумышленнику добыть информацию о заражении компьютера вирусами.

Для сканирования сетей хакеры используют как собственные утилиты, так и легальные стандартные программные средства, предназначенные для повседневной работы администраторов сетей. Такие средства могут быть оснащены удобным графическим интерфейсом, который упрощает анализ пакетов и в некоторых случаях позволяет преступнику воссоздать полную конфигурацию связей, адреса подсетей и узлов исследуемой сети, т. е. выполнить ***network-mapping.***

Сетевая разведка может проводиться злоумышленником, который уже выбрал конкретный тип атаки, знает, как ее осуществить, имеет все необходимые для этого средства, и ему осталось только найти такую систему-жертву, характеристики которой подходят для проведения этой конкретной атаки. Преступник «забрасывает невод», сканирует тысячи сетей, пока не найдет компьютер, ОС, приложение, обладающие теми уязвимостями, на которые рассчитана его атака. Сетевая разведка может включать также поиск компьютеров с конкретными вредительскими программами, которые были ранее внедрены другими кибер-преступниками и которые злоумышленник хочет использовать для организации собственной атаки. Благодаря высокой степени стандартизации программного и аппаратного обеспечения и наличию большого числа однотипных вредительских программ, вероятность успеха такого поиска очень высока.

Несколько другая ситуация, когда сетевая разведка выполняется злоумышленником, который уже определился с системой-жертвой, но пока не решил, какую он будет проводить атаку. Он сканирует эту конкретную систему,

чтобы найти ее уязвимости. Чем больше известно о системе: модели компьютеров и устройств сетевой инфраструктуры, типы и версии ОС и приложений, параметры конфигурации устройств и данные об активированных сервисах и протоколах — тем проще преступнику обнаружить слабое место системы.

Вопросы к главе 1

1. Как соотносятся понятия «безопасность компьютерных сетей» и «безопасность информационных систем»:

а) они совпадают;

б) понятие «безопасность компьютерных сетей» уже понятия «безопасность информационных систем»;

в) понятие «безопасность компьютерных сетей» шире понятия «безопасность информационных систем».

2. Что из перечисленного может быть отнесено к субъектам системы контроля доступа к ресурсам ИС:

а) пользователи;

б) устройства;

в) прикладные процессы;

г) файлы.

3. Что из перечисленного может быть отнесено к объектам системы контроля доступа к ресурсам ИС:

а) пользователи;

б) устройства;

в) пропускная способности каналов связи;

г) прикладные процессы;

д) файлы;

е) сетевые сервисы.

4. Используя приведенный ниже список терминов, вставьте пропущенные слова в следующее предложение: «Пользователи (...) получают ... к ресурсам (...) ИС в результате ..., однако прежде им необходимо успешно пройти ... и ...». В ответе перечислите буквенные обозначения терминов в той последовательности, в которой они идут в предложении.

а) аутентификация;

б) объекты;

в) авторизация;

г) идентификация;

д) субъекты;

е) права доступа.

5. Какие из перечисленных ниже свойств образуют триаду безопасности CIA:

а) захват привилегий;

б) доступность;

в) конфиденциальность;

г) неотказуемость;
д) аутентичность;
е) владение;
ж) целостность;
з) полезность.

6. Какие из перечисленных ниже свойств образуют гексаду Паркера:
а) захват привилегий;
б) доступность;
в) конфиденциальность;
г) неотказуемость;
д) аутентичность;
е) владение;
ж) целостность;
з) полезность.

7. Вставьте пропущенные слова из списка. «Есть информация о том, что в Интернете уже появились .... направленные на использование ... новой версии браузера. Реализация данной ... может привести к ..., которая нанесет ... нашему предприятию». В ответе перечислите буквенные обозначения терминов в той последовательности, в которой они идут в предложении:
а) атака;
б) уязвимость;
в) эксплойт;
г) ущерб;
д) угроза.

8. Какие из перечисленных атак являются активными:
а) прослушивание сетевого трафика;
б) спуфинг;
в) социальный инжиниринг;
г) атака «Отказ в обслуживании»;
д) внедрение вируса.

9. Установите соответствие следующих терминов на английском и русском языках:

ущерб;	а) Denial of Service;
уязвимость;	б) Availability;
вредоносное ПО;	в) Integrity;
доступность;	г) Vulnerability;
целостность;	д) Malware;
подмена содержимого пакета; распределенная атака «отказ в обслуживании»; атака «отказ в обслуживании»	е) Impact; ж) Spoofing; з) DDoS.

В ответе перечислите буквенные обозначения англоязычных терминов в той последовательности, в которой идут термины на русском языке.

10. Если система контроля доступа не разрешила пользователю распечатать документ на принтере, то такую ситуацию можно назвать:
а) отказ в обслуживании;
б) пользователь не авторизован выполнять данную операцию;
в) пользователь сделал попытку несанкционированного доступа к устройству;
г) операция «печать» не входит в набор допустимых операций принтера.

11. Для каких из следующих ИС доступность может быть важнее конфиденциальности:
а) ИС завода по производству вооружений;

б) ИС университета;
в) ИС пенсионного фонда РФ;
г) ИС интернет-магазина.

12. Приведите примеры ситуаций, при которых обеспечивается конфиденциальность, но не гарантируется целостность данных.

13. Приведите примеры действий воображаемого злоумышленника, направленных на нарушение доступности данных.

2 Управление рисками

Каждое предприятие (организация, компания) создается, чтобы решать некую задачу: издавать книги, обучать студентов, продавать продукты, разрабатывать законы и т. д. Такого роды задачи называют бизнес-задачами. В наше «компьютерное» время практически все предприятия для решения своих бизнес-задач привлекают информационные системы. К сожалению, использование информационных технологий приносит с собой не только повышение производительности, новое качество услуг и другие преимущества, оно является также источником новых угроз и рисков для бизнеса.

Для защиты бизнеса предприятия от информационных угроз разрабатываются и используются самые разнообразные методы и средства обеспечения информационной безопасности, направленные на достижение защищенности того или иного компонента ИС или системы в целом. Разные типы предприятий требуют различных подходов — принципы организации информационной защиты информационной системы крупного промышленного предприятия, частного интернет-магазина или государственного учреждения отличаются коренным образом. В некоторых случаях может быть использовано типовое, стандартизованное решение, не требующее больших затрат времени и сил, в других же —• для стратегического планирования системы информационной безопасности большого предприятия — нельзя обойтись без сложного многофакторного анализа и многоступенчатой процедуры принятия решений.

Именно такой фундаментальный подход используется при решении задачи ***управления информационными рисками,*** т. е. рисками, связанными с атаками на информационную систему предприятия. Основная цель управления рисками — защитить *предприятие и его бизнес,* а не только информационную систему, которая поддерживает этот бизнес. Поэтому решение этой задачи возможно только совместными усилиями представителей бизнеса и специалистов по информационным технологиям.

Суть управления рисками — это системный анализ угроз, прогнозирование и оценка их последствий для предприятия, ранжирование угроз по степени их вероятного осуществления и опасности последствий и, наконец, выбор на приоритетной основе контрмер, направленных на смягчение или исключение возможного негативного воздействия этих нарушений на деятельность предприятия. Заметим, что источником рисков для предприятия могут стать не только (а зачастую и не столько) уязвимости и угрозы,

относящиеся к информационной инфраструктуре, но и ошибочные действия в финансовой и производственной сфере, однако мы выносим эти риски за рамки нашего рассмотрения.

Управление рисками включает следующие три укрупненных этапа (рис. 2.1):

- анализ уязвимостей и угроз;
- оценка рисков;
- управление рисками.

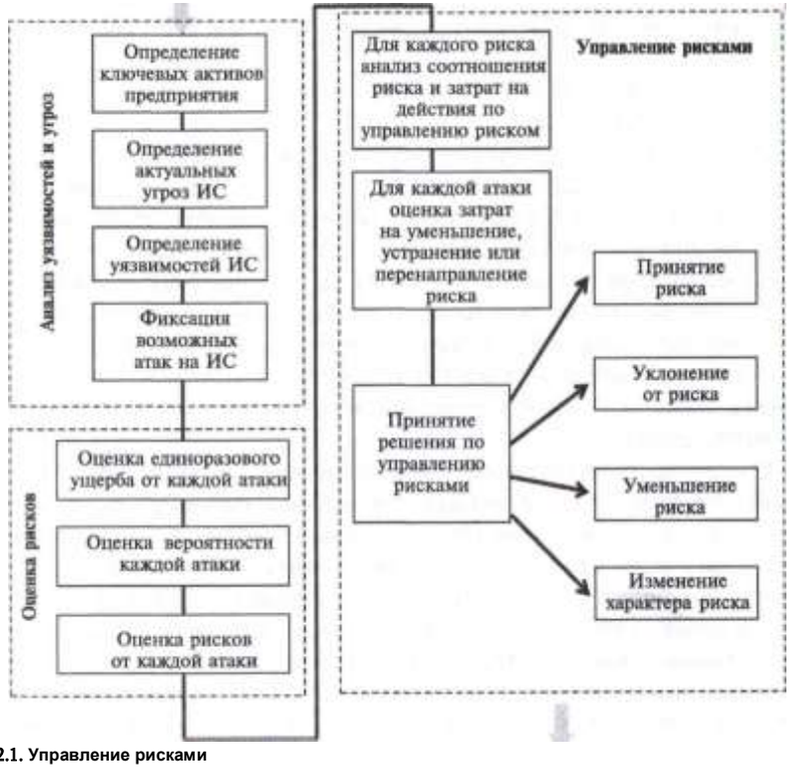


Рис. 2.1. Управление рисками

Анализ уязвимостей и угроз

Важным понятием управления рисками является понятие ***«актив».*** В данном контексте под этим термином понимается ресурс предприятия, представляющий для него ценность и являющийся объектом защиты. К активам могут быть отнесены:

- физические ресурсы, такие как оборудование, недвижимость, транспортные средства, вычислительные устройства и др.;
- информационные ресурсы — различные базы данных, программное обеспечение, документация;
- клиентская база, сотрудники предприятия, партнеры, портфели заказов и др.

Именно по отношению к активам предприятия определяются угрозы. Для того чтобы идентифицировать существующие угрозы, разработчики системы безопасности должны выяснить следующее:

Какие активы/информационные ресурсы предприятия нуждаются в защите прежде всего?

Кто является наиболее вероятным нарушителем безопасности? Какие наиболее опасные угрозы могут исходить от вероятного противника?

Какие уязвимости ИС могут позволить нарушителю воплотить его угрозы в действительность?

Ответы на эти вопросы отражают специфику предприятия, для которого разрабатывается система безопасности. Для предприятия, бизнесом которого является разработка программного обеспечения, наиболее защищаемым ресурсом скорее всего являются файлы с программными кодами, а наиболее вероятным противником — внешние или внутренние технические шпионы от компании-конкурента. В качестве наиболее опасной угрозы в данном случае можно рассматривать несанкционированный доступ к файлам, а уязвимостью может служить, например, недостаточно надежная система аутентификации или авторизации.

Совсем другие ответы на те же самые вопросы может дать специалист по безопасности автоматизированной системы управления технологическим процессом (АСУТП) энергетического предприятия. Здесь критически важными активами являются управляющая программа, а также все аппаратные средства (процессоры, датчики и исполнительные устройства), которые реализуют процесс управления. Потенциальным нарушителем может быть персонал АСУТП, а угрозой — неумышленные ошибки (вспомним причину крупнейшей катастрофы в Чернобыле). Уязвимостями в таких случаях могут, например, оказаться непродуманная система внесения изменений в ПО, отсутствие или неполнота процедур проверки на допустимость команд, поступающих на исполнительные устройства. Для некоторых систем управления процессами реального времени могут быть рассмотрены и террористические угрозы. Портрет потенциального нарушителя и характер угроз зависит от того, подключена ли система управления к Интернету или реализована на базе изолированной от Интернета локальной сети.

Для решении задачи идентификации угроз широко привлекаются экспертные оценки, а также анализ статистических данных об инцидентах и тенденциях нарушений безопасности данного предприятия либо предприятия, подобного данному. Однако сбор релевантной статистики возможен только при соблюдении нескольких условий: предприятие должно быть крупным, территориально распределённым, иметь продолжительную историю и не содержать новейших технических компонентов. Если же предприятие относительно небольшое и построено на основе новейших технологий, то для него невозможно найти достоверные статистические данные. В таких случаях используют ***метод анализа факторов риска,*** т. е. обстоятельств, способствующих возникновению того или иного нарушения безопасности. Так, например, угроза несанкционированного доступа к некоторым данным более вероятна, если среди персонала ИС имеются недостаточно проверенные люди, или если эти данные представляют существенную ценность для потенциальных злоумышленников, или если из-за проходящей на предприятии реорганизации некоторые защитные механизмы временно ослаблены.

В результате проведённых опросов, изучения статистики и учета факторов, влияющих на возникновение угроз, формируется список возможных угроз.

Заметим, что все вопросы и ответы относительно потенциального противника и угроз с его стороны формулируются в *предположительной* манере, с использованием вероятностных категорий. Совсем другое дело, когда речь идет об идентификация уязвимостей — этот процесс включает в себя объективное обследование *реально существующей* компьютерной сети, реально существующих административных процедур и реально существующего персонала.

В любой момент времени для любой системы можно указать множество различных видов уязвимостей, например для операционных систем и приложений новые уязвимости появляются чуть ли не каждый день. Поэтому выявление их вручную является очень трудоемкой задачей. В некоторых случаях уязвимость легче устранить, чем обнаружить. Поэтому для автоматизации поиска уязвимостей используют различные программные инструменты — ***средства сканирования*** уязвимостей, такие, например, как McAfee, Nessus, Qualys, Rapid 7,

Tenable Network Security, каждое из которых обладает большой базой данных о существующих уязвимостях. Сканирование заключается в последовательном (адрес за адресом узла, или номер за номером порта, или идентификатор за идентификатором сетевого соединения) направлении запросов целевой системе. Затем на основании полученных ответов генерируется «информационный отпечаток» и, наконец, сравнением «отпечатка» с записями в базе данных выполняется идентификация уязвимости. Прежде чем приступить к использованию системы сканирования, необходимо ее настроить, чтобы исключить частые ложные срабатывания.

Чтобы сделать поиск более обозримым, исследуемую ИС рекомендуется разбить на несколько частей и для каждой части проводить отдельную процедуру сканирования, настроенную на те типы уязвимостей, которые характерны именно для данной части сети. Например, при анализе внешних частей системы, непосредственно выходящих в Интернет, основное внимание уделяется поиску уязвимостей в работающих в зоне публичного доступа веб-серверах, а также в устройствах контроля входного трафика, таких как файерволы и фильтрующие маршрутизаторы. А во внутренних частях сети преимущественно исследуются бреши защиты хостов, в частности уязвимости операционных систем и приложений.

При использовании сканеров надо проявлять осторожность, так как они могут быть использованы злоумышленниками для изучения слабых мест системы на этапе подготовки атак.

Помимо сканера для поиска уязвимостей может быть использован другой, более активный метод исследования системы — ***тестирование на проникновение (penetration testing).*** В этом случае специалисты по безопасности «входят в роль» злоумышленника и пытаются найти и использовать бреши в защите системы. Им разрешается проводить беседы с сотрудниками тестируемой системы, в которых они методами социальной инженерии пытаются «вытянуть» из них информацию, полезную для организации атаки. Преимуществом тестирования на проникновение перед системами сканирования является то, что первые включают обнаружение и попытки использования не только известных типов уязвимостей, но и тех, о существовании которых администратор сети даже не подозревал.

В результате сканирования или тестирования на проникновение специалисты по безопасности получают список уязвимостей, который может оказаться весьма обширным. Как правило, этот список может быть существенно сокращён исключением из него так называемых стандартных (или типовых) уязвимостей, которые эксплуатируются наиболее распространёнными атаками — вирусами, несанкционированным доступом и др. Перечень такого рода угроз дают некоторые регламентирующие документы, стандарты и спецификации. Стандартами предписывается все типовые уязвимости устранять безусловно, не проводя для них анализа рисков, например, установкой необходимых патчей или конфигурационных настроек для ликвидации уязвимостей ОС.

Далее список уязвимостей сопоставляется со списком угроз с тем, чтобы определить перечень возможных атак. Напомним, что успешная атака возможна лишь при совпадении двух условий:

- наличие уязвимости;
- существование угрозы, направленной на использование данной уязвимости.

Исключив из списка все уязвимости, для которых отсутствуют соответствующие угрозы, можно зафиксировать перечень возможных атак.

Ущерб как мера риска

На этом этапе специалисты, отвечающие за безопасность ИС, совместно с руководителями бизнеса должны относительно каждой атаки ответить на следующие вопросы:

Какие потери несёт предприятие в результате данной атаки?

Насколько вероятно проведение данной атаки?

Какие суммарные потери может понести предприятие из-за данной атаки в течение некоторого периода?

Известно, что абсолютная безопасность информационной системы не может быть обеспечена никакими средствами: всегда есть вероятность появления ошибок в программах и аппаратуре, в работе персонала и административных процедурах. Кроме того, регулярно появляются новые угрозы со стороны злоумышленников. Поэтому целью обеспечения информационной безопасности является не исключение, а ***минимизация возможного негативного влияния,*** вызванного нарушениями конфиденциальности, целостности или доступности информационной системы.

Из этого также следует, что надо каким-то образом ранжировать угрозы, чтобы решить, какими угрозами можно пренебречь, а на какие обратить основное внимание, направить материальные средства и усилия по обеспечению защиты. Но как определить, какие угрозы являются более, а какие менее значимыми, что, например, опаснее: раскрытие внутренней документации компании, которое произошло в результате взлома файловой системы, или недоступность в течение часа корпоративной базы данных, вандализм, приведший к невозможности использования веб-сайта компании, или компрометация паролей сотрудников финансового отдела?

Естественной мерой опасности атак и угроз является возможный ущерб, связанный с каждым из этих нарушений. Причём в качестве ущерба рассматриваются не только и не столько потери, связанные с восстановлением работы ИС, в частности серверов, файловой системы или системы аутентификации, — главное внимание должно быть уделено потерям, которые в результате этих нарушений понесло *предприятие,* которое строит свой бизнес на базе этой ИС. Еще раз подчеркнем, что в конечном счете ущерб от успешно проведенной атаки наносится не ИС, не компьютерной сети, а предприятию и, более конкретно, обладателю информации* или собственнику предприятия, которое использует данную ИС. Например, когда злоумышленник взламывает компьютерную сеть банка и получает доступ к персональным данным клиента, а затем, воспользовавшись ими, снимает средства с его счета, то ущерб несет банк, информационная система которого оказалась скомпрометированной. Наряду с банком, обладателем данных, позволяющих получить доступ к счету, является клиент. Если эти данные стали известны злоумышленнику по вине клиента, например он позволил их подсмотреть, то материальный ущерб может быть возложен на него3 4.

Разные атаки на ИС вызывают разные последствия для бизнеса. После некоторых атак предприятие может понести тяжелейший урон, но все-таки сохранить «живучесть» и продолжать работать, хотя и в деградированном состоянии. Поражение же критически важных функций предприятия приводит его к краху. Например, недоступность базы данных с персональными данными всех клиентов в течение нескольких часов может нанести банку значительный ущерб, в то время как раскрытие, утечка этих же данных может вызвать его банкротство. Ущерб может быть:

- *материальным* — прямые денежные потери, упущенная выгода, снижение прибыли, затраты на восстановление информационной системы;
- *репутационным,* который обычно имеет свой материальный эквивалент;
- *невосполнимым,* который не может быть выражен в деньгах. Например, ущерб, возникший в результате атаки на информационную систему, управляющую ядерным реактором, может привести к невосполнимому ущербу: человеческим жертвам и катастрофическим изменениям окружающей среды.

В этом разделе мы ограничимся рассмотрением только тех случаев, когда ущерб может быть выражен денежным эквивалентом.

Будучи сложной сама по себе, процедура количественной оценки ущерба осложняется еще тем, что для ее проведения требуется привлечение руководителей предприятия, так как во многих случаях только они могут судить о масштабах материальных потерь бизнеса.

Однако знание возможного ***единоразового ущерба*** (***L***) от той или иной

' ***Обладатель информации*** — лицо, самостоятельно создавшее информацию либо получившее на основании закона или договора право разрешать или ограничивать доступ к информации (Федеральный закон № 149-ФЗ).

атаки еще недостаточно, чтобы ранжировать нарушения по степени их опасности. Для оценки суммарных потерь, вызванных нарушениями безопасности, например, за год, необходимо для каждой рассматриваемой атаки оценивать не только возможный ущерб, но и ***вероятность*** (***P***) ее проведения.

Для ранжирования атак по степени опасности используется пара *{ущерб от атаки, вероятность атаки},* которую называют риском (***R***). ***Риск —*** это вероятностная оценка ущерба, который может быть нанесен предприятию данной атакой в течение некоторого периода времени. Риск тем выше, чем больше ущерб от атаки и чем выше ее вероятность. Риск вычисляется как произведение разового ущерба на вероятность атаки: ***R = LP.***

Некоторые методики (например, RiskWatch) подходят более формально к вычислению суммарного ущерба — в качестве вероятностной оценки они предлагают использовать не вероятность осуществления отдельной атаки, а *математическое ожидание количества атак данного вида* (*N*), приходящихся на заданный период. Ведь одна атака может случиться только один раз за все время существования системы, а другая — несколько раз в месяц, и тем самым ущерб от последней будет многократно увеличен. Произведение *LN* дает вероятностную оценку суммарного ущерба от каждого вида атак, на основании этой величины выполняется ранжирование соответствующих рисков.

Вычислить риск значительно сложнее, чем обнаружить уязвимость, так как для этого необходимо решить две трудные задачи: оценить вероятность возникновения угрозы и вычислить связанный с этой угрозой ущерб. Действительно, одно дело — просто обнаружить, что некто в системе использует легко разгадываемый пароль, совсем другое дело — оценить как быстро этим может воспользоваться злоумышленник и какие возможные потери в результате этого может понести предприятие.

Рассмотрим схематично оценку риска, связанного с использованием сервиса удаленного доступа. Пусть, например, удаленный доступ использует компания, разрабатывающая программное обеспечение. Основным режимом работы сотрудников компании является работа из дома. В тех случаях, когда сотруднику требуется взаимодействие с коллегами (отладка нескольких программных единиц, написанных

4 Банки часто берут на себя ущерб, вызванный неумышленными ошибками клиентов.

разными людьми), он использует удаленный доступ. На предыдущих этапах поиска уязвимостей и угроз было определено, что существующее программное обеспечение удаленного доступа недостаточно надежное и что существуют угрозы использования этих уязвимостей программного кода для несанкционированного доступа к устройствам локальной сети. Вероятность (P) атаки на подсистему удаленного доступа оценивается как весьма высокая — в соответствии с имеющимися статистическими данными более половины всех атак используют небезопасный удаленный доступ.

Возможный ущерб предприятия (***L***), связанный с несанкционированным доступом конкурентов к разрабатываемым программным продуктам, может быть оценен недополученной прибылью с продаж этих продуктов. Риск может быть оценен как произведение ***PL.*** Заметим, что оценить вероятность атаки более сложно, чем оценить ущерб, так как очень трудно найти хоть какие-то объективные данные, на которые можно было бы положиться и, кроме того, вероятность значительно меняется со временем. Косвенную информацию, характеризующую вероятность атаки, можно извлечь, анализируя возможные мотивы злоумышленника, то, при каком стечении обстоятельств она скорее всего произойдет, а также в какие моменты защита наиболее ослаблена (особенно это касается нарушений физической защиты).

Управление рисками

Управление рисками предполагает, что по отношению к каждому риску могут быть предприняты следующие меры:

Принятие риска (***assumption***). Такая мера избирается для неизбежных атак, наносящих приемлемый ущерб. Какой уровень ущерба считать приемлемым, зависит от специфики предприятия.

Устранение риска (***avoidance***). Из того, что атака, а значит, и риск, существует только при наличии уязвимости и использующей ее угрозы, следует, что существующий риск можно свести на нет устранением либо уязвимости, либо угрозы. Обратимся снова к примеру компании, разрабатывающей программное обеспечение. Если на этапе оценки ущерба было определено, что существует значительный риск нарушения конфиденциальности из-за уязвимостей сервиса удаленного доступа, то самый простой и радикальный способ избавиться от этого риска — устранить уязвимость удаленного доступа путем полного отказа от него и перевода всех сотрудников на работу в офисах компании.

Мы можем также устранить риск несанкционированного доступа к некоторым конфиденциальным данным, сделав эти данные несекретными, например перейдя от коммерческого к открытому программному обеспечению. Таким же образом, переместив веб-сервер из локальной сети в зону публичного доступа, мы устраняем риск несанкционированного доступа к ресурсам локальной сети, исходящий от клиентов веб-сервиса. Однако для того чтобы решиться на подобные изменения, нужно прежде всего переосмыслить и пересмотреть соответствующие ***бизнес-решения,*** а это не всегда оказывается возможным.

Другим способом исключения риска, как уже было сказано, является

устранение угрозы. Например, риск, связанный с внедрением некоторого вируса, исчезнет, если установить надежную противовирусную систему, рассчитанную на обнаружение и блокировку данного типа вируса.

Снижение риска (***mitigation***). В тех случаях, когда риск невозможно ни принять, ни устранить, предпринимаются действия по его снижению. Например, всегда существует некоторая вероятность проникновения злоумышленников в систему, скажем, путем подбора паролей. А значит, всегда существует риск несанкционированного доступа к данным. Однако этот риск можно снизить, уменьшив уязвимость системы аутентификации пользователей, например установив более строгие требования к длине и сменяемости паролей.

Перенаправление риска (***transference***). Если риск невозможно ни принять, ни устранить, ни даже существенно снизить, то риск может быть передан страховой компании или компании-партнеру. Например, предприятие может застраховать свою ключевую базу данных от простоев, вызванных атаками злоумышленников, определив размер страховой премии равным соответствующему риску. Такую операцию называют также ***изменением характера риска.***

Для принятия решения о том, какая мера должна быть применена к каждому из установленных рисков, рассматриваются все имеющиеся в распоряжении предприятия способы устранения, смягчения и перенаправления соответствующего риска, а также вариант принятия этого риска.

Далее для каждого из предложенных способов проводится анализ затрат. Стоимость устранения или снижения риска зависит от избранного способа. Так, например, решение уже упоминавшейся софтверной компании об отказе от удаленного доступа с целью избежания риска потребует значительных затрат на оборудование рабочих мест для программистов в офисе компании, включая, например, арендную плату за помещения, затраты на энергообеспечение, стоимость содержания вспомогательного персонала и т. п.

Окончательно принять решение о том, какие меры следует применить к риску, можно только сопоставив риск с затратами на каждый из предложенных вариантов его устранения, снижения, перенаправления или принятия. Понятно, что для всех рисков, имеющих денежный эквивалент, имеют смысл только такие варианты, затраты на которые меньше самого риска.

Следует заметить, что задача управления рисками не может быть решена строго математически. Это связано со сложностью реальной информационной системы и базирующегося на ней бизнеса. Многообразие угроз и уязвимостей, специфические особенности реализации атак в каждом конкретном случае, взаимозависимость нарушений безопасности и взаимосвязь средств защиты — это и многое другое делает практически невозможным разработать адекватную математическую модель информационной системы предприятия, а также определить значения исходных данных, необходимых для математического решения задачи оценки рисков. Как, например, определить один из важнейших параметров — вероятность проведения конкретной атаки? В лучшем случае может быть получена некоторая приблизительная оценка этой вероятности

после анализа статистических данных об атаках, нанесенных в прошлом данной системе или подобным ИС в аналогичных условиях, а также после сбора и обобщения мнений экспертов.

В условиях, когда принципиально невозможно получить точные количественные оценки, во многих методиках оценки рисков используют для оценки вероятностей *качественные* категории в диапазоне от «маловероятно» до «почти достоверно» с несколькими промежуточными градациями. Каждой качественной категории ставится в соответствие число в диапазоне от 0 до 1, которое интерпретируется как вероятность ***P*** проведения атаки. Подобным же качественным образом может оцениваться ущерб от атак: «незначительный», «средний», «большой», «очень большой». Точность полученных оценок в этом случае зависит от квалификации экспертов, от того, насколько богатая статистика накоплена за время существования данного предприятия. Сочетания качественных оценок вероятности и ущерба даёт в результате качественную оценку риска, например, если возникновение соответствующей угрозы «маловероятно», а ущерб «незначительный», то риску присваивается категория «незначительный», что означает его приемлемость и отсутствие необходимости в принятии каких-либо мер.

Пример качественного подхода к оценке рисков содержится в ***«Методикеуправления рисками FRAP***»5, в которой предлагается оценивать как вероятность атаки, так и ущерб от атаки по шкале качественных характеристик: «высокий», «средний», «низкий» (табл. 2.1).

Таблица 2.1

Матрица рисков FRAP

Вероятность	Высокий ущерб	Средний ущерб	Низкий ущерб
Высокая	Оценка риска A: корректирующие действия обязательно должны быть выполнены и немедленно	Оценка риска B: действия, связанные с данным риском, следует выполнить	Оценка риска C: в данный момент требуется только мониторинг ситуации
Средняя	Оценка риска B: действия, связанные с данным риском, следует выполнить	Оценка риска B: действия, связанные с данным риском, следует выполнить	Оценка риска C: в данный момент требуется только мониторинг ситуации
Низкая	Оценка риска B: действия, связанные с данным риском, следует выполнить	Оценка риска C: в данный момент требуется только мониторинг ситуации	Оценка риска D: в данный момент не следует предпринимать никаких действий

Риск также оценивается качественно, в соответствии с правилом, задаваемым таблицей 2.1 (где A, B, C и D — градации риска, от высокого к низкому).

Такой подход широко используется и в других методиках оценки рисков, при этом количество уровней в шкалах и правило, задаваемое матрицей, могут быть самыми различными.

---

5 Томас Пелтиер (Thomas R. Peltier) «Групповой процесс анализа рисков» (Facilitated Risk Analysis Process, FRAP).

Помимо сложности получения исходных данных для оценки рисков, проблемой также является отсутствие формального алгоритма управления рисками. Как, например, поступить с риском ***R\,*** если для его устранения требуются затраты $А_ь$ а для снижения этого риска до уровня $Д_2$ ¯ затраты $А_2$? Если меньше $А_2$, то первый вариант предпочтительнее, ну а если Ai больше $А_2$, то ответ уже не столь очевиден.

А как быть, если в результате применения мер, направленных на снижение риска, нарушаются другие требования безопасности ИС, например, удобство работы пользователей?

Как поступить, если устранение одного риска увеличивает другие риски?

Эти и другие подобные вопросы делают процесс управления рисками неоднозначным, требующим от специалистов не только глубоких знаний и эрудиции, но и умения принимать решения в условиях неопределённости. И хотя почти все количественные параметры, фигурирующие в методиках управления рисками, пока плохо поддаются точному количественному анализу, даже качественные оценки этих параметров, базирующиеся на мнениях экспертов, позволяют выработать план построения системы безопасности, представив его в виде логической последовательности шагов.

Рис. 2.2. Циклический характер процесса управления рисками

Важной особенностью процесса управления рисками является его цикличность (рис. 2.2). Предприятие, как живой организм, постоянно меняется во времени, в каждый момент возникают одни и устраняются другие уязвимости и угрозы, время от времени в систему защиты вносятся исправления, снижающие размеры рисков, или, напротив, реализуются новые бизнес-решения, увеличивающие риски. Все это требует регулярной переоценки рисков и принятия заново решений об управлении ими.

Стандартные методики оценки рисков

Как видно из предыдущего, процесс анализа рисков и принятия решений относительно того, как с ними поступать, является далеко не тривиальным. И в такой ситуации чрезвычайно востребованными оказываются стандартные методики, целью которых является оказание помощи руководителям и сотрудникам, отвечающим за безопасность, в том, как успешно справиться с задачей защиты своих предприятий от информационных рисков.

Все представленные ниже методики сходятся в том, что представляют процесс управления рисками в виде уже знакомой нам последовательности этапов: идентификация угроз, оценка рисков, собственно управление рисками. Однако они имеют и существенные отличия друг от друга, заключающиеся в специфике предприятий, для которых они предназначены (для крупных или небольших предприятий, для государственных организаций или частных компаний и т. п.), природе рисков (информационные, финансовые, производственные), характере критериев оценки (качественные или количественные), уровне решаемых задач, степенью формализации, качеством предоставляемого сервиса и др. Следовательно, решение задачи оценки рисков стоит начинать с выбора подходящей для ваших целей методики.

Ниже мы коротко рассмотрим несколько популярных методик, ориентированных на анализ информационных рисков.

Рекомендации NIST

Под эгидой Национального института стандартов и технологий (NIST) выпущено ряд общепризнанных документов, относящихся к

так называемой 800-й серии, которые содержат рекомендации в области управления рисками.

Одним из самых популярных документов NIST является «Руководство по оценке рисков для компьютерных систем», ***NIST SP 800- 30**** оно представляет собой 40-страничный документ, адресованный опытным специалистам и новичкам, техническому персоналу и административным работникам — всем тем, кто имеет дело с управлением рисками.

Особенности данного метода:

- данный метод имеет дело с рисками, связанными с использованием компьютерных систем;
- для анализа рисков используется системный подход, выражающийся в том, что фазы решения задачи анализа рисков встраиваются в жизненный цикл исследуемой системы, включая пред- проектную стадию, стадии проектирования, создания и функционирования предприятия:
- метод рассчитан на использование качественных оценок вероятностей, ущерба и риска, хотя методика способна обрабатывать и количественные оценки риска.

Основу данного руководства составляют три главы:

- «Обзор основных понятий» — в этой главе описывается место управления рисками в жизненном цикле развития системы, а также определяются ключевые функциональные роли специалистов, которые должны быть привлечены к решению задачи оценки рисков. В команду должны входить собственники, бизнес- менеджеры, специалисты по безопасности информационных систем:
- «Оценивание рисков», в ней описывается последовательность из 9 шагов, которые нужно предпринять команде аналитиков, чтобы получить практически значимые результаты по оценке рисков;
- глава «Уменьшение рисков» содержит рекомендации по приоритезации, анализу, выбору и реализации соответствующих мер, направленных на уменьшение рисков. Другими словами, этот раздел посвящён управлению рисками.

При рассмотрении этой методики мы ограничимся обзором тех 9 шагов, которые описываются в разделе «Оценивание рисков».

1. ***Сбор информации о системе —*** это начальный этап работы, идачей которого является очерчивание границ системы, которая бу- Д|' | подвергнута анализу. В пределах этих границ необходимо провести

http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf

анализ бизнес-процессов, а также инвентаризацию программных и аппаратных средств, информационных ресурсов, персонала и пользователей. Кроме того, должны быть отмечены все внешние и внутренние связи, сетевая топология. На этом шаге должны быть собраны характеристики всех используемых в настоящий момент средств безопасности, описаны политики безопасности и текущие требования к безопасности. Для систем, которые только разрабатываются, характеристики системы должны быть извлечены из проектной документации. Для сбора информации могут использоваться опросные листы, интервьюирование персонала, анализ технической документации, средства автоматизированного сбора информации (например, CA Asset Manager) о программных и аппаратных ресурсах системы.

2. *Идентификация уязвимостей* — сравнительно хорошо проработанная процедура, включающая анализ баз данных существующих уязвимостей, таких, например, как БД NIST I-CAT*, и тестирование с использованием сканеров. Уязвимости ищут как в технических средствах системы, так и в «идеологических» — политиках безопасности, планах развития предприятия, в организационных процедурах, затрагивающих информационную безопасность. Результатом этого шага является список обнаруженных уязвимостей.

3. *Идентификация угроз.* На этом шаге рекомендуется создать модель потенциального нарушителя, которая включает тип нарушителя (сотрудник предприятия, уволенный сотрудник, хакер, технический шпион и др.), его мотивацию (неумышленная ошибка, месть, хулиганское самоутверждение, желание получить конкурентное преимущество), возможный сценарий атаки (модификация файла, неавторизованный доступ к финансовым данным, приведение в неисправность почтового сервера, удаление файлов с финансовой информацией). Суммируя все возможные сценарии, команда аналитиков формирует список возможных угроз.

4. *Анализ мер безопасности.* Для этого рекомендуется заполнить таблицу, в которой перечислены практически все виды, семейства и категории средств и процедур безопасности. Для каждого вида и категории должно быть отмечено, насколько данное средство соответствует текущим требованиям к безопасности. Одним из методов, применимом на данном шаге, является использование опросных листов, например, таких, которые рекомендуются для этих целей в другом документе NIST SP 800-26.

5. *Определение вероятностей.* В данной методике вероятность угроз оценивается в качественных категориях «высокая», «средняя», «низкая», для которых в руководстве даны описания.

http://icat.nist.gov

6. *Анализ ущерба* является одной из самых сложных задач управления рисками, оставаясь такой даже в упрощенной постановке, когда необходимо провести только ранжирование ущерба для разных типов атак. В руководстве приводится детальная классификация видов ущерба: ущерб информационным активам от нарушений конфиденциальности, целостности, доступности, ущерб

репутации, падение конкурентоспособности, упущенная выгода, кража, мошенничество, судебные преследования и др. Для разных видов ущерба определяется смысл категорий «высокий ущерб», «средний ущерб», «низкий ущерб». Присвоение ущербу той или иной категории может быть выполнено на основании результатов предыдущих исследований системы, если таковые были; косвенно об ущербе можно судить по требованиям защиты того или иного ресурса; часто очень полезным окапывается интервьюирование собственников предприятия или информационной инфраструктуры.

7. *Определение риска.* В данном руководстве риск определяется для конкретной пары {уязвимость, угроза} и является функцией следующих переменных:

- вероятности использования данной уязвимости данным источником угрозы;
- ущерба, который будет нанесен системе в случае успешной реализации угрозы;
- адекватности планируемых или существующих средств защиты, направленных на предотвращение или уменьшения данного риска. Для оценки риска в методике NIST используется тот же подход,

что и в упомянутой ранее методике FRAP, с той разницей, что качественно заданным уровням ущерба и вероятности ставятся в соответствие условные численные значения, например шкале вероятности «высокая-средняя-низкая» соответствует численная шкала «1,0 — 0,5 — 0,1», а для соответствующей качественной шкалы ущерба — условная численная шкала «100 — 50 — 10». На пересечении строк и столбцов вычисляется произведение, которое соответствует численной оценке риска (табл. 1.2). Наличие численных значений для рисков позволяет снизить степень произвольности правила, задаваемого матрицей, которое определяет уровень риска для каждого сочетания уровней вероятности и ущерба. Заметим, что использование условной численной шкалы не означает, что для оценки рисков применяется количественный подход, так как при количественном подходе оценивается не относительные, а абсолютные величины ущерба, его вероятности и риска.

Далее на этом шаге определяется словесная интерпретация каждого из полученных уровней риска, включающая определение степени

Таблица 2.2

Матрица рисков, используемая в методике оценки рисков NIST 800-30

Вероятность	Высокий ущерб (от 100 до 50)	Средний ущерб (от 10 до 50)	Низкий ущерб (от 1 до 10)
Высокая (1,0)	Высокий риск: 100-1,0 = 100	Средний риск: 50 ■ 1,0 = 50	Низкий риск: 10- 1,0 = 10
Средняя (0,5)	Средний риск: 100 • 0,5 = 50	Средний риск: 50 • 0,5 = 25	Низкий риск: 10-0,5= 5
Низкая (0,1)	Низкий риск: 100-0,1 = 10	Низкий риск: 50 • 0,1 = 5	Низкий риск: 10 • 0,1 = 1

серьезности понесенного ущерба и рекомендованной реакции на возникшую ситуацию.

8. ***Выработка рекомендаций.*** На этом шаге должны быть определены те меры безопасности, которые необходимо предпринять, чтобы устранить или уменьшить риски. При этом должны учитываться эффективность и надежность принимаемых решений, их соответствие общему законодательству и политике безопасности предприятия, влияние этих мер на производительность и другие функциональные характеристики предприятия. При выборе мер защиты необходимо использовать системный подход, принимать во внимание возможности разных уровней системы безопасности. Контрмерами могут стать, например, изменения в политике безопасности или в должностных инструкциях, установка дополнительного программно-аппаратного комплекса.

9. ***Составление отчета по оценке рисков.*** Данное руководство содержит подробное описание формы отчета и рекомендации по генерации его содержимого. Кроме того, в приложениях даны примеры других документов, используемых в процедурах исследования системы: формы для проведения опросов, фиксирования результатов анализа, таблиц, для выработки планов мероприятий, относящихся к процедуре управления рисками.

Приведем еще несколько нормативных документов NIST в области анализа рисков.

NIST SP 800-39 «Управление рисками информационной безопасности: Взгляд с точки зрения предприятия, его бизнес-задач и информационных систем», 2011 г. Авторы определяют документ как руководство по созданию интегральных, охватывающих все предприятие, программ по управлению рисками, которым подвергается предприятие из-за нарушений информационной безопасности. Особенностью документа является то, что, не претендуя на замену других, относящихся к анализу рисков стандартов и рекомендаций NIST, он сводит их воедино и тем самым становится выразителем единой непротиворечивой концепции информационной безопасности организации.

NIST 800-66 «Информационная безопасность», 2008 г. Этот документ первоначально предназначался для поддержки работ в области ыщиты персональных данных в здравоохранении, однако через некоторое время он нашел широкое применение и в других отраслях. Особый интерес в данном контексте представляет глава «Основы управления рисками».

NIST SP 800-53 «Средства управления безопасностью и конфиденциальностью государственных информационных систем и органи- мций», 2013 г. Документ предлагает расширенный подход к информационной безопасности предприятия и управлению рисками. Ключевым элементом предлагаемой стратегии является идея сочетания развития системы с ее непрерывным мониторингом. Цель мониторинга — снабжать менеджеров старшего звена, которые принимают нжущие решения на основе анализа рисков, актуальной информацией, генерируемой практически в режиме реального времени.

NIST CVSS «Общая система оценки уязвимостей», 2007 г. предлагает методику подсчёта баллов для каждого вида уязвимостей, основанную на учёте, во-первых, сущностных, неотъемлемых особеннос- гей уязвимости данного вида, во-вторых, динамических, меняющихся но времени характеристик уязвимости, в-третьих, свойств уязвимос- 1 и, специфических для данной конкретной среды, в которой функционирует тестируемая система. Эта универсальная методика подсчё- ia баллов даёт возможность разным специалистам по информационной безопасности, исследователям и разработчикам систем и приложений, говорить об уязвимостях на едином, общем для всех языке. Рассчитанные балы могут быть использованы в процедурах анализа рисков для сравнения уязвимостей и для количественного ранжиро- пания уязвимостей по степени возможного влияния на систему, в которой они обнаружены. Методика CVSS используется совместно с Национальной базой данных уязвимостей (The National Vulnerability Database, NVD), предоставляющей оценки для почти всех известных уязвимостей. В 2014 году ожидается выход следующей версии данного стандарта.

Методика оценки рисков RiskWatch

RiskWatch — это методика и набор поддерживающих ее программных инструментов для анализа рисков. RiskWatch была разработана одноименной компанией на основе методики NIST 800-30. К особенностям RiskWatch можно отнести:

- привлечение для оценки рисков количественных вероятностных характеристик, в том числе математического ожидания числа реализованных угроз;

- использование количественных критериев для анализа рисков, в частности в методике предлагается использовать количественную оценку ожидаемого годового ущерба;
- оценивание эффективности защитных мероприятий сравнением ожидаемого ущерба в случаях ***использования*** и ***неиспользования*** средств защиты. Количественная оценка соотношения потерь от угроз безопасности и затрат на создание системы защиты позволяет рассчитать уровень возврата инвестиций.

Комплекс включает четыре интерактивные программы следующего предназначения:

1) для аудита физических средств безопасности;
2) для оценки информационных рисков;
3) для оценки соответствия требованиям стандарта HIPAA*;
4) для оценки соответствия требованиям стандарта ISO 17799.

Интерактивный характер программ и удобный пользовательский интерфейс, построенный на основе вложенных меню, предлагающих пользователю выбор из множества вариантов, существенно упрощают работу аналитиков. Например, пользователю, выбравшему определенный тип организации, система предлагает специфические для данного типа организации перечни типов критически важных ресурсов, уязвимостей, угроз и средств безопасности.

Кроме многовариантного выбора, предлагаемого экранными формами, пользователь программного комплекса может вводить данные об исследуемой системе вручную, а также выполнять импорт информации из отчетов, сгенерированных программными сканерами уязвимостей или другими инструментальными средствами аудита компьютерных сетей.

Некоторые численные значения параметров исследуемой системы извлекаются из внутренних баз данных программного комплекса RiskWatch. Например, для многих классов угроз система располагает следующими полезными данными:

- среднее число реализаций угрозы данного типа на данной ограниченной территории — это локальная оценка LAFE (Local Annual Frequency Estimate);
- среднее число реализаций угрозы данного типа на существенно более общирной территории — это глобальная оценка SAFE (Standard Annual Frequency Estimate).

Эти данные используются при расчетах ожидаемых годовых потерь/рисков.

Стандарт HIPAA (US Healthcare Insurance Portability and Accountability Act) разработан для регулирования вопросов информационной безопасности в сфере здравоохранения США.

Программный комплекс имеет также базу данных типовых решений для защиты от типовых угроз. Многие из стандартных решений дают возможность вычислить важную экономическую оценку —

■ чдачу от инвестиций. Пусть, например, для уменьшения риска от некоторой

угрозы планируется применить некоторое средство защи- ıЫ. Отдача от инвестиций в данное средство защиты вычисляется мк разность между ожидаемым снижением потерь от данной угрозы при использовании данного защитного средства и затратами на его пнедрение и поддержание:

отдача от инвестиций = снижение потерь - затраты на защиту.

Суммируя полученные значения отдачи по всем угрозам, получаем общую отдачу от инвестиций в обеспечение безопасности системы.

Простота данной процедуры сделала возможным проектирование | истемы безопасности предприятия методом виртуальных проб и виртуальных ошибок. При таком подходе, называемом «что если?» (what it?), выполняется проверка эффективности и потенциальных последствий самых разных вариантов защиты.

Программный комплекс RiskWatch имеет развитые средства генерации отчетов разной степени подробности и разной тематики: краткие итоговые и полные отчеты по результатам аудита исследуемой | истемы, отчеты о стоимости критически важных активов и ожидаемых потерях, об угрозах и средствах защиты от них, об эффективности инвестиций.

Методика CRAMM

Метод анализа и управления рисками ***CRAMM*** (CCTA Risk Analysis and Management Method) был разработан в 1987 году Центральным агентством по компьютерам и телекоммуникациям (CCTA) Великобритании, сейчас переименованным в Министерство правительственной связи (Office of Government Commerce, OGC). CRAMM стал одним из первых общепризнанных практически используемых наборов инструментальных средств для анализа рисков. Последней версией агого продукта является CRAMM v.5.

Главной особенностью данного метода является его гибкость и универсальность:

- Он применим к анализу как крупных, так и малых предприятий, как правительственного, так и коммерческого сектора. Универсальность достигается тем, что в состав комплекса CRAMM входят две различные базы знаний — Правительственный профиль (Government profile) и Коммерческий профиль (Commercial profile), каждая из которых используется в соответствующих случаях.
- CRAMM применим к анализу как технических (нарушения безопасности аппаратных и программных средств ИС), так и нетехнических (нарушения физической защиты, административных процедур) аспектов безопасности.
- В основе метода CRAMM лежит комплексный подход к оценке рисков, сочетающий количественные и качественные методы анализа.
- Методика включает обширную библиотеку защитных решений, ориентированную на разные типы предприятий и угроз.

Метод реализуется в виде трех стадий, соответствующих типовому процессу управления рисками.

1- я стадия — идентификация активов и определение их ценности. Эта типичная задача для всех методик анализа рисков, отличия состоят лишь в том, какие типы ресурсов, принимаются во внимание той или иной методикой. В данном случае в число ресурсов включаются вычислительная аппаратура, программные системные средства и прикладные пакеты, хранящиеся в системе данные, — т. е. все, что составляет информационную вычислительную систему.

На данном этапе должна быть определена ценность каждого из ресурсов, которая косвенно измеряется ущербом, который наносится предприятию в результате атаки на данный ресурс. В методике определено 8 видов ущерба, которые укрупненно можно представить двумя видами — финансовым и репутационным. Каждый вид ущерба оценивается по 10-бальной шкале. Так, например, ценность актива, на восстановление которого потребуется меньше $1000, устанавливается в размере 2 баллов, а 10 баллов дается при затратах более $100 000. В числе других критериев ценности ресурса могут использоваться репутационные или иммиджевые потери, для оценки которых может быть использована, например, следующая шкала: если в результате атаки наблюдается негативная реакция группы частных лиц то ущерб оценивается в 2 балла, критика в средствах массовой информации и общественных низовых организациях — б баллов, критика в национальных СМИ, имеющая широкий общественный резонанс 8 баллов, международная негативная реакция — 10 баллов. Если все активы системы получают средние оценки по всем критериям ниже 3 баллов, то выдается заключение, что исследуемой системе достаточно базового уровня защиты, никаких дополнительных средств безопасности не требуется, и вторая стадия процесса анализа рисков пропускается.

2- я стадия — оценка угроз и уязвимостей. CRAMM способна анализировать широкий спектр умышленных и неумышленных угроз, включая угрозы от хакеров, вирусы, ошибки аппаратуры и программ, намеренные повреждения системы или терроризм, непреднамеренные человеческие ошибки. Для каждой угрозы определяется вероятность

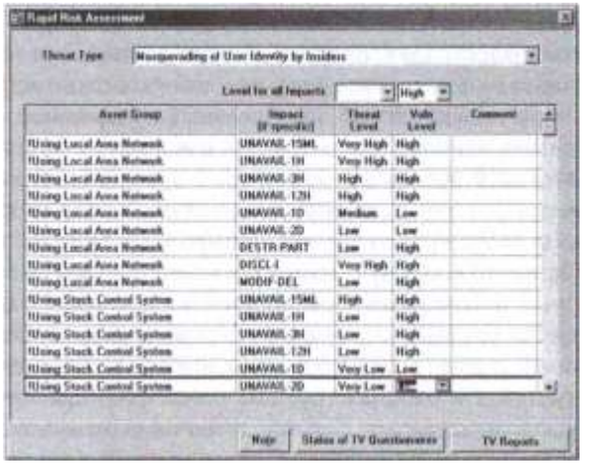


Рис. 2.3. Пример экранной формы программной компоненты метода CRAMM, предназначенной для получения оценки вероятности угрозы использования чужого идентификатора сотрудниками организации («маскарад»)

ее возникновения. Как и в других методиках, для этих целей рекомендуется использовать статистические данные, мнения экспертов, а также метод оценки факторов риска. CRAMM предлагает бальную процедуру оценки угроз и уязвимостей. В соответствии с этой процедурой для каждого вида угрозы система предлагает два перечня факторов:

- факторы, косвенно свидетельствующие о возможности возникновения данной угрозы;
- факторы, косвенно свидетельствующие о наличии уязвимостей, которые могут позволить реализовать данную угрозу.

Например, для угрозы «маскарад» (заключающейся в том, что один сотрудник организации использует идентификатор другого) в числе факторов угрозы система по умолчанию рассматривает текущее количество зарегистрированных инцидентов такого рода, тенденцию в статистике, наличие в системе информации, представляющей интерес для потенциальных нарушителей, наличие альтернатив данному виду угрозы и др. Насколько каждый фактор риска соответствует реальному состоянию исследуемого предприятия, выясняется с помощью вопросов и ответов в режиме многовариантного выбора (рис. 2.3). Каждому ответу пользователя система CRAMM ставит в соответствие некоторое число. Например, если на вопрос «Какова тенденция в количестве зарегистрированных инцидентов такого рода на вашем предприятии?» пользователь системы выбирает ответ «повышается», то данный фактор получает оценку +10, а если ответом

было «снижается», то —10. Или если из ответа на вопрос о наличии альтернативы «маскараду», следует, что для достижения той же цели (несанкционированный доступ) злоумышленник может использовать другой, альтернативный способ, не связанный с подменной идентификатора, то оценка данной угрозы также снижается на 10 баллов. Для получения итоговой оценки степени угрозы определенного класса все баллы суммируются и по заданной условной шкале определяется уровень угрозы в диапазоне от «очень низкая» (9 баллов) до «очень высокая» (более 40 баллов) с тремя промежуточными значениями.

Аналогично, с помощью опрашивания и назначения баллов ответам оценивается уровень уязвимости. Уровень уязвимости системы по отношению к угрозе «маскарад» повышается с увеличением количества сотрудников в рабочих группах, когда факт данного нарушения может быть легко скрыт, а руководство не вполне осведомлено о действиях сотрудников, и т. д. Поэтому за каждый подтверждающий эти факты ответ начисляются положительные баллы. Однако уровень данной уязвимости снижается (и начисляются отрицательные баллы), если, например, в системе большая часть информация является открытой. Итоговый уровень уязвимости устанавливается как «низкий» (9 баллов), «средний» (от 10 до 19 баллов) или «высокий» (20 и более баллов).

На основе оценок уровней угроз и уязвимостей определяется оценка уровня риска по семибалльной шкале.

3- я стадия — выбор средств противодействия и рекомендаций. CRAMM включает в себя очень большую библиотеку защитных мер, в которой представлены более 3000 детализированных решений, разбитых на 70 логических групп. Программное обеспечение метода CRAMM использует оценки риска, полученные на предыдущем этапе, и сравнивает их с уровнем безопасности (пороговый уровень, ассоциируемый с каждой контрмерой) для того, чтобы определить, является ли риск существенно большим, чтобы оправдать установку данной меры. На этом этапе могут быть использованы различные вспомогательные инструменты, входящие в состав CRAMM, в частности сервис «Что если?» и средства генерации отчета.

Методика OCTAVE

Методика ***OCTAVE*** «Оценка критичных угроз, активов и уязвимостей» (Operationally Critical Threat, Asset, and Vulnerability Evaluation) разработана в 2001 году институтом Software Engineering Institute (SEI) при университете Карнеги Меллон (Carnegie Mellon University). Методика предлагает формализованную процедуру оценки рисков, представленную в виде четкой последовательности шагов с ясно определенными целями. Для каждого шага определена структура исходных сведений, необходимых на данном этапе, а также структура данных, которые должны быть получены в результате. Особенность этой методики состоит в том, что для решения задачи оценки рисков она предлагает использовать только сотрудников тестируемого предприятия, без привлечения сторонних консультантов. Утверждается, что

они более информированы о том, как функционирует их предприятие, тают его слабости и внутренние резервы. Привлекательность методики OCTAVE заключается также в том, что вся поддерживающая ее документация: комплект руководств, методические рекомендации, специально разработанные формы для заполнения в ходе исследования системы и протоколирования результатов обсуждений — все это общедоступно и бесплатно.

В соответствии с методикой OCTAVE для оценки рисков на предприятии создаются небольшие, по 3-5 человек, группы сотрудников предприятия, имеющих различную специализацию и относящихся к разным уровням служебной иерархии. Эти команды проводят серию рабочих совещаний, на которых фокусируются на трех (в общем-то достаточно типичных для задачи оценки рисков) проблемах:

- определение профилей угрозы для критически важных активов предприятия;
- идентификация уязвимостей инфраструктуры, как организационных, так и технологических;
- разработка стратегии безопасности и планов снижения рисков, направленных на поддержку основных целей бизнеса.

Определение профилей угрозы для ключевых активов

Работа на данном этапе начинается с определения списка наиболее важных активов, оценки их стоимости, и определения требований безопасности для них.

Для того чтобы сформулировать требования безопасности к каждому ключевому активу, команда аналитиков должна ответить на следующие вопросы:

Является ли информация, составляющая этот актив, чувствительной? Содержит ли она персональные данные? Должен ли доступ к ней быть ограничен для кого-либо?

Насколько необходимо гарантировать для данного информационного ресурса аутентичность, полноту и точность?

Является ли доступность обязательным требованием?

Могут ли быть предъявлены к данному активу какие-либо другие требования, касающиеся безопасности?

В результате этих исследований для каждого актива формулируются требования безопасности в терминах конфиденциальности, целостности, доступности.

Таблица 2.3

Параметры типового профиля угрозы, порождаемой человеком

Параметр профиля	Английский термин	Возможные значения параметра
Ссылка на актив	Asset	Любой важный актив Доступ через сеть Физический доступ Сотрудник предприятия Внешний пользователь
Тип доступа (опционально)	Access	Неумышленное, случайное нарушение Умышленное нарушение Раскрытие информации Модификация информации
Источник угрозы, субъект, нарушающий безопасность	Actor	Разрушение информации, потеря информации
Тип нарушения (опционально)	Motive	Отказ в обслуживании
Результат нарушения требований безопасности	Outcome	

Далее для каждого из ключевых активов создается так называемый ***профиль угрозы, т.*** е. набор характеристик возможного нежелательного события, связанного с данным активом. Методика предусматривает выполнение ряда шагов, прежде чем с каждым активом матривает будет связан набор угроз.

В методике используется понятие типового (generic) профиля угрозы

Существует три типовых профиля, отличающихся типом источника угрозы и типовым набором последствий, к которым может привести реализация данной угрозы. Профиль угрозы, исходящей от человека-нарушителя, получающего доступ к активу либо сеть (умышленный или неумышленный, с использованием физического доступа (умышленный), либо с использованием технических проблем информационной системы (ошибки аппаратуры, ошибки программного обеспечения, вирусы, вредные программы и т.п.);

профиль угрозы, вызванной прочими угрозами (природные катаклизмы, террористические атаки, перерывы в электроснабжении и

Типовые профили угрозы характеризуются частично перекрывающимися наборами параметров. Рассмотрим вначале типовой профиль угрозы, порождаемой неловеком-нарушителем (табл. 2.3). В первом столбце таблицы перечислены параметры, характерные именно для этого типового профиля. Каждый параметр имеет несколько возможных значений. Все возможные комбинации значений параметров типового профиля могут быть графически представлены в виде дерева (рис. 2.4). На рисунке изображена часть дерева, которая соответствует физическому способу проникновения, она симметрично повторяет профиль угрозы, возникающей из-за доступа нарушителя через сеть.

На рис. 2,5 показан еще один типовой профиль — профиль угроз, вызванных системными техническими проблемами.

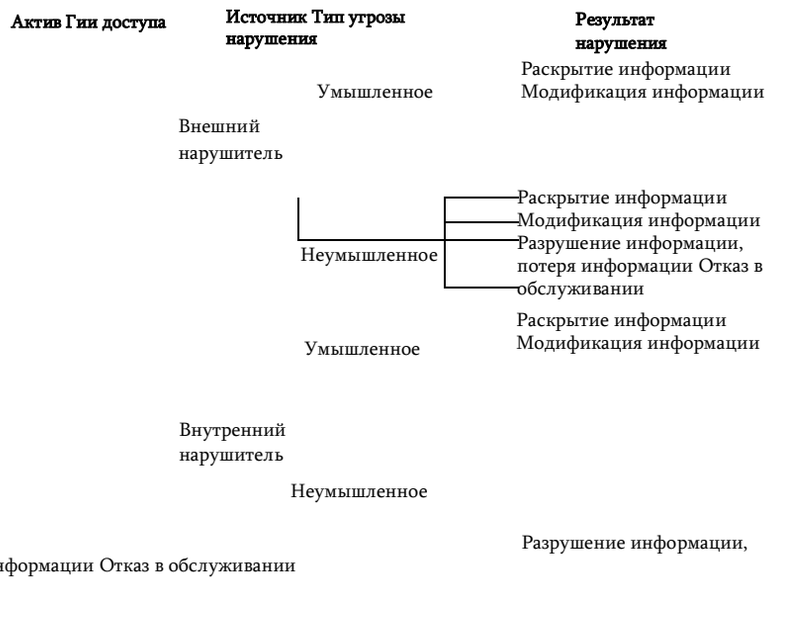
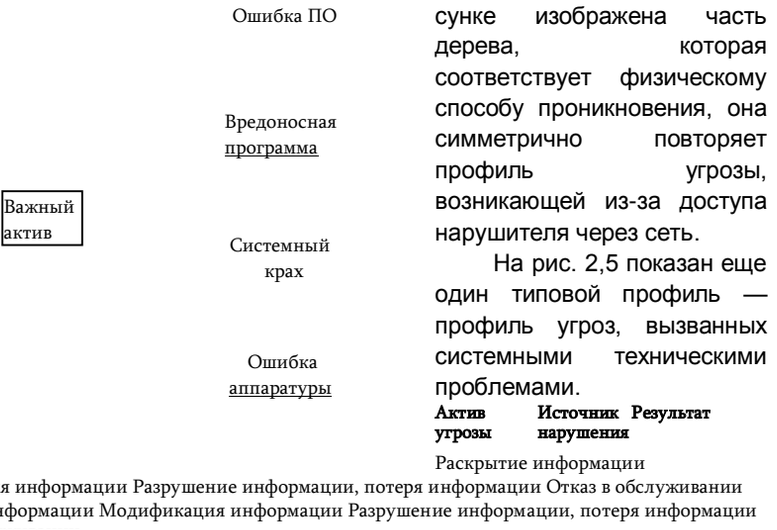


Рис. 2.4. **Типовой профиль угрозы, порождаемой человеком при физическом доступе**



Раскрытие информации Модификация информации Разрушение информации, потеря информации Отказ в обслуживании Раскрытие информации Модификация информации Разрушение информации, потеря информации Отказ в обслуживании

Рис. **2.5. Профиль угроз, вызванных техническими проблемами**

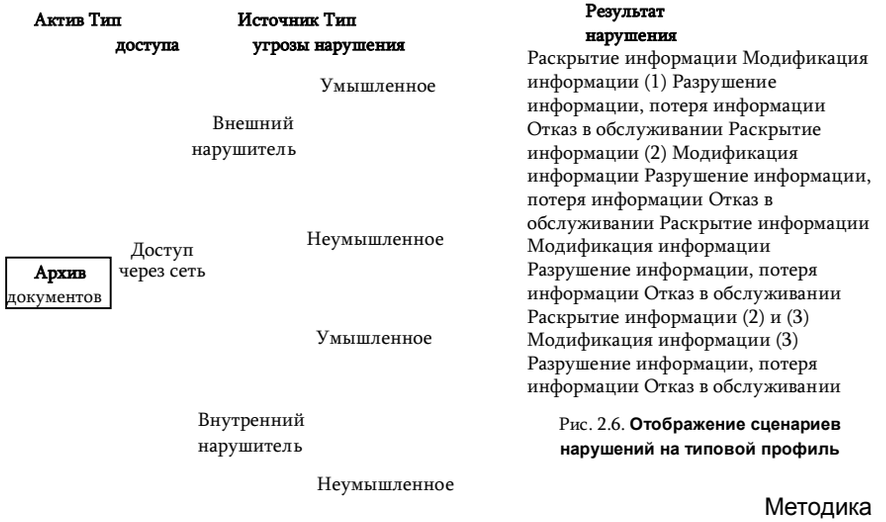


Рис. 2.6. **Отображение сценариев нарушений на типовой профиль**

Методика предусматривает возможность «подгонки» типовых профилей под особенности предприятия. Этот процесс может включать добавление новых и исключение не применимых в данных условиях типов угроз. Например, профиль угрозы, вызванной человеком-нарушителем, может быть расширен за счет более тонкой дифференциации источника угрозы: вместо простой классификации на инсайдера и внешнего нарушителя, могут быть дополнительные виды лиц- нарушителей — шпионы, конкуренты, вандалы и другие категории.

Руководствуясь типовыми профилями, команда специалистов, проводящих оценку рисков, должна составить реальные профили угроз для каждого важного актива. Рассмотрим пример, в котором в качестве актива выступает архив проектной документации на разрабатываемое компанией программное обеспечение (спецификации, исходные и объектные коды, инструкции по инсталляции и т. п.). Пусть этот архив хранится на сервере в виде простой файловой структуры. В результате обсуждений на рабочих совещаниях были сформулированы три сценария возможных нарушений по отношению к этому активу (табл. 2.4).

Затем было сделано отображение всех трех сценариев нарушений на типовой профиль угроз с участием человека-нарушителя (рис. 2.6). Толстые линии показывают существующие угрозы активу (всего было установлено 4 угрозы), а тонкие — угрозы, которые не были установлены.

Заметим, что в нашем примере для простоты все угрозы связаны с человеком-нарушителем, получающим доступ через сеть. В реальности вполне возможна ситуация, когда по отношению к данному активу — архиву документации — действуют и другие типы угроз, например сбои дискового накопителя, перебои источника питания или внедрение вируса. Команда аналитиков должна отобразить все возможные сценарии нарушений на соответствующие типовые профили.

Таблица 2.4

Сценарии потенциальных нарушений

	Сценарий потенциального нарушения
	Значения параметров угрозы
Люди, работая с файловой системой, иногда случайно вводят ошибочные данные. Результат таких ошибок — некорректные изменения содержимого или ошибочное удаление файлов	**Актив — архив документации Тип доступа — через сеть Источник угрозы — внутренние нарушители Тип нарушения — неумышленное Результат — модификация, потеря информации**
Некоторые лица могут попытаться использовать информацию из архива в своих личных интересах	**Актив — архив документации Тип доступа — через сеть Источник угрозы — внутренние и внешние нарушители Тип нарушения — умышленное Результат — раскрытие информации (несанкционированный доступ)**
Присущие данному программному архиву дефекты и уязвимости могут быть использованы внешними атакующими для просмотра и/или изменения хранимой в нем информации	**Актив — архив документации Тип доступа — через сеть Источник угрозы — внешний нарушитель Тип нарушения — умышленное Результат — модификация, несанкционированный доступ**

Выполнив подобные действия по отношению ко всем важным активам и всем возможным сценариям нарушений, специалисты по оценке рисков должны установить, все ли угрозы, существующие в системе в данный момент, учтены, например, не может ли внешний злоумышленник в третьем сценарии не только раскрыть и модифицировать информацию из архива, но и удалить/разрушить данные или даже вызвать отказ в обслуживании. В результате такого анализа на дереве профиля угроз могут появится новые толстые линии.

Таким образом, результатами первого этапа анализа являются связывание каждого актива с относящимися к нему угрозами, а также собранная и структурированная информация об этих угрозах (значения параметров профилей безопасности).

Идентификация уязвимостей инфраструктуры

На втором этапе основное внимание уделяется определению ***инфраструктурных уязвимостей, т.е.*** уязвимостей программных и аппа-

.отельной системы, поддерживающих су- ратных компонентов вычислите ,, у каждого из выдели „итрл^нных активов. Например, для архива ществование документации, в число таких компонентов входят файловый сервер (компьютер и ОС), рабочие станции сотрудников, обращающихся к архиву документации, сетевое оборудование (коммутаторы, маршрутизаторы) того сегмента сети, * к0т°Р°м Работают указанные сервер и рабочие станции, сервер удаленного доступа, который используют сотрудники-телекомьютеры ДЛЯ Работы из дома, их домашние компьютеры и др.

Для всех этих компонентов [1] ный анализ уязвимостей с инфраструктуры проводится тщатель- применением стандартных методик, специальных сканеров, а также каталога уязвимостей для разных программных и аппаратных компонентов инфраструктуры, который предусматривается методикой в числе ДРУ™х вспомогательных материалов.

Идентифицированные уязвимости подразделяются на три класса- уязвимости, подлежащие немедленному устранению, уязвимости, которые надлежит устранить в ближайшее время и уязвимости, не требующие особого внимания. Результаты этой работы должны быть „паяпительного отчета об уязвимостях клю- представлены в виде предвари ,,,кт\/ры. Этот отчет бӱдет̈использован на чевых элементов инфраструктур̈ ... поофилеи рисков, третьем этапе при создании пр ^

Разработка стратпеии безопасности и планов снижений рисков

На третьем этапе вырабатывается стратегия обеспечения безопасности и планы снижения Р^ков, для чего решаются следующие задачи:

- идентифицируются риски Д[ля актив]ов,
- разрабатываются критерии оценки рисков,
- проводится оценка рисков;
- разрабатывается стратегия защиты,
- . разрабатываются планы уменьшения рисков;
- . выполняется корректировка стратегии и планов защиты руководством компании;
- определяются следующие щзги.

Идентификация риска в Данной методике сводится к идентификации ущерба. Последний прежде всего зависит от того, к каким результатам привело нарушение: к Раскрытию, модификации, потере, разрушению или недоступно[сти] информации. Для идентификации „ лгновываясь на значениях параметра «рерисков аналитики должны, осн^ ^ пей угроз каждого актива, дать ответы зультат нарушения» из профил *i* на следующие вопросы:

Какой ущерб репутации компании может быть нанесён данным нарушением?

Насколько может снизится доверие клиентов?

Возможен ли ущерб здоровью клиентов?

Скажется ли это нарушение на производительности предприятия?

Может ли это нарушение привести к штрафным санкциям?

К какому финансовому ущербу может привести данное нарушение?

В результате анализа ответов на вышеперечисленные вопросы для каждого актива создаются словесные описания ущерба, называемые ***дескрипторами ущерба*** (*impact description).*

Методика OCTAVE не требует обязательной оценки вероятности осуществления угрозы, самым главным в оценке рисков здесь считается определение возможного ущерба. Если все же было решено привлечь вероятности для оценки рисков, то команда аналитиков прежде всего должна прийти к согласию, какую вероятность в числовом выражении они считают высокой, средней и низкой. Затем на основе этой универсальной шкалы качественных оценок должны быть определены вероятности атак. Полученные данные о вероятностях могут быть привлечены позже (на 6 шаге — корректировке стратегии) во время пересмотра приоритетов нарушений.

Следующей задачей является *определение критериев оценки риска/ущерба.* выраженных в категориях здравого смысла. Для этого команда аналитиков должна прийти к согласию о том, что в их понимании означает «большой», «средний» или «незначительный» ущерб для их предприятия. Так, например, для одной организации простой корпоративного почтового сервера в течение часа является незначительным ущербом, а для другой — большим. В результате обсуждений создается универсальная шкала качественных оценок, применимая ко всем видам ущерба: репутационному, физической безопасности, финансовому, правовому, ущербу производительности.

Собственно *оценка рисков/ущерба* сводится к анализу дескрипторов ущерба, сгенерированных ранее, и приписыванию им соответствующей качественной оценки по шкале «большой - средний - незначительный». В результате для каждого актива создается ***профиль риска,*** включающий следующие данные:

- профиль угрозы;
- требования безопасности;
- дескриптор ущерба;
- качественная оценка ущерба;
- перечень инфраструктурных компонентов, относящиеся к данному активу, и отчет об уязвимостях;
- характеристики вероятности угрозы (опционально).

Следующей задачей является создание *стратегии защиты.* Для этого команде аналитиков потребуется ответить на ряд вопросов общего характера:

Какие меры, направленные на обучение и тренинг персонала, могли бы усовершенствовать безопасность предприятия?

Что могло бы способствовать интеграции вопросов безопасности в бизнес-стратегию предприятия?

Что может гарантировать понимание всеми членами коллектива их роли и ответственности в деле поддержания безопасности?

Какой уровень расходов на обеспечение безопасности предприятия можно считать достаточным?

Являются ли принципы и процедуры обеспечения безопасности, принятые на предприятии (в том числе в сфере работы с внешними организациями), удовлетворительными? Если нет, то как их можно улучшить?

Что нужно сделать, чтобы гарантировать непрерывность функционирования предприятия и готовность персонала к работе в режиме восстановления после природных и техногенных катастроф?

На этом этапе аналитики ревизируют принятый порядок и текущие способы решения проблем безопасности, фиксируют слабости организационных процедур, тщательно изучают профили риска для всех ключевых активов предприятия. Команда начинает работу по созданию стратегии с изучения каталога образцов стратегических решений в области безопасности, предусматриваемого в числе вспомогательных материалов методики OCTAVE. Из каталога выбираются решения, в наибольшей мере соответствующие нуждам предприятия, например те из них, которые используют средства безопасности, уже используемые на предприятии, или направлены на устранение тех угроз, которые в условиях данного предприятия отнесены к высокоприоритетным, или касаются именно тех уязвимостей, которые были обнаружены в компонентах инфраструктуры предприятия.

Полученная в результате стратегия безопасности представляет собой набор предложений, адресуемых командой аналитиков руководству предприятия. Предложения в основном имеют характер долговременных инициатив, действующих в масштабе всего предприятия.

Следующим шагом является разработка *планов по уменьшению рисков.* Эта работа выполняется на основе всей собранной ранее информации. На этом шаге для каждого ключевого актива необходимо выяснить следующее:

Какие типы угроз могут вызвать наибольший ущерб для бизнес- целей предприятия?

Какие действия могут быть предприняты для распознавания этого типа угроз, когда они возникают?

Какие действия могут быть предприняты для предотвращения или повышения устойчивости к данному типу угроз ?

Какие действия могут способствовать восстановлению после реализации угроз этого типа?

Какие риски следует всеми силами уменьшать, а какие можно принять, не предпринимая никаких действий по их смягчению?

Какие меры могут быть использованы для подтверждения эффективности

предлагаемых планов по уменьшению рисков?

Вопрос о принятии рисков решается на основе оценок ущерба. В качестве дополнительного фактора на этом этапе могут быть использованы оценки вероятности (если они делались ранее). Оценки ущерба используются также для ранжирования рисков, с тем чтобы гарантировать наибольшее внимание рискам с наивысшим приоритетом. Предложения по уменьшению рисков подготавливаются на основе критического анализа текущего состояния используемых на предприятии средств обеспечения безопасности, а также профилей риска для всех ключевых активов.

Результатом данного этапа должны стать конкретные планы действий по смягчению последствий атак на критически важные ресурсы предприятия. Эти планы адресуются административному персоналу предприятия.

Далее руководство компании совместно с командой аналитиков изучает представленную стратегию безопасности и планы по уменьшению рисков и при необходимости вносит в них свои *коррективы.* Основаниями для внесения исправлений в представленные документы могут оказаться финансовые и организационные ограничения, а также перспективы развития предприятия. Данный этап важен не только тем, что в разработку стратегии и планов свой вклад внесли бизнес- руководители, имеющие более точное представление ресурсах, которыми располагает предприятие, но и тем, что команда специалистов- «безопасников» получила поддержку бизнеса, столь необходимую для проведения в жизнь важного проекта.

Последнее, что предусмотрено в методике OCTAVE, — это определение руководством предприятия своих *следующих шагов* в направлении реализации стратегии безопасности и планов по снижению рисков.

В результате анализа рисков совместными усилиями специалистов по безопасности и руководителей предприятия формируется ***политика безопасности*** (*security policy)* — совокупность документированных управленческих решений, руководящих принципов, правил, процедур и практических приёмов, направленных на защиту информации и поддерживающей ее инфраструктуры. Политика безопасности определяет *стратегические* направления информационной защиты предп

риятия, а именно, очерчивает круг критически важных информационных ресурсов предприятия, защита которых представляет наивысший приоритет, предлагает меры, которые могут быть предприняты для устранения или уменьшения связанных с этими ресурсами рисков. На основе найденной стратегии разрабатывается программа обеспечения безопасности **ИС,** планируется совокупный бюджет, необходимый для выполнения программы, назначаются руководители и очерчивается зона их ответственности.

Вопросы к главе 2

1. Конечной целью управления рисками является защита:

а) компьютерной инфраструктуры ИС;

б) бизнеса предприятия:

в) собственника предприятия;

г) владельца информационных ресурсов;

д) информационной системы предприятия;

е) ничего из перечисленного.

2. Возможные источники рисков предприятия:

а) ошибки в финансовой сфере;

б) производственные ошибки;

в) неправильная работа программных средств;

г) внешние вредительские действия;

д) неумышленные ошибки персонала;

е) ничего из перечисленного.

3. Что из перечисленного можно отнести к активам учебного предприятия:

а) здание;

б) автомобили;

в) вычислительная техника;

г) программное обеспечение;

д) базы данных, документация;

е) сотрудники предприятия;

ж) ничего из перечисленного.

4. Выберите из списка возможные факторы риска, свзязанного с несанкционированным доступом к кодам разрабатываемого на софтверном предприятии программного комплекса:

а) на предприятии проходит реструктуризация подразделений;

б) разрабатываемая программа составит острую конкуренцию существующему аналогу;

в) большое количество зарегистрированных инцидентов такого рода;

г) в последнее время участились DoS-атаки на сервер, хранящий исходные коды программных продуктов, разработанных на предприятии.

5. Какой процедурой пользовался администратор при исследовании системы, если известно, что ему удалось обнаружить совершенно новый тип уязвимости:

а) тестирование на проникновение;

б) сканирование уязвимостей.

6. Вставьте в следующем предложении пропущенные слова из списка: «... тем выше, чем больше ... от ... и чем выше вероятность...»:

а) ущерб;
б) риск;
в) атака.

7. Какие меры могут быть предприняты по отношению к каждому риску при управлении рисками:

а) устранение риска;

б) оценка риска;

в) принятие риска;

г) снижение риска;

д) перенаправление риска;

е) изменение характера риска.

8. Для оценки вероятности возникновения угроз используются:

а) аналитические расчеты;

б) статистические данные;

в) мнения экспертов;

г) анализ факторов риска;

д) экспериментальная проверка.

9. На каком этапе завершается процесс управления рисками:

а) анализ уязвимостей;

б) оценка рисков;

в) анализ угроз;

г) определение критических ресурсов;

д) принятие контрмер;

е) ни на каком.

10. Когда можно не предпринимать никаких действий по отношению к выявленному риску:

а) если риск меньше средств, затрачиваемых на его устранение;

б) если величина риска не превышает принятого на предприятии приемлемого уровня потерь;

в) всегда необходимо принимать меры к уменьшению риска.

11. Какие из перечисленных аббривиатур являются названиями стандартных ме- юдик оценки рисков:

а) CRAMM;

б) RiskWatch;

в) McAfee;

г) OCTAVE;

д) Wireshark;

е) FRAP.

12. Типовой профиль угрозы в методике OCTAVE создается для:

а) каждого пользователя;

б) каждого критического актива;

в) каждого источника уязвимости;

г) каждого нарушителя.

13. Предложите собственный вариант качественных шкал оценки вероятностей, ущерба и риска, а также правило соответствия пар (вероятность, ущерб) уровням риска. Для каждой градации шкалы составьте описание.

3 СИСТЕМНЫЙ ПОДХОД К УПРАВЛЕНИЮ БЕЗОПАСНОСТЬ Ю

Обычно первое, что ассоциируется с информационной безопасностью, — это антивирусные программы, файерволы, системы шифрования, аутентификации, аудита и другие технические средства защиты. Бесспорно, роль этих средств в обеспечении безопасности велика, однако не меньшее, а иногда и большее влияние на безопасность системы оказывают средства, построенные на качественно другой основе.

Видеокамера и надежный замок в офисе, продуманная процедура приема сотрудников на работу, закон, угрожающий хакеру уголовным преследованием, стандарт, помогающий провести анализ возможного ущерба из-за действия нарушителя, — все эти очень не похожие средства одинаково важны для обеспечения безопасности.

Успех в области информационной безопасности может принести только ***системный подход,*** при котором средства защиты разных типов применяются совместно и под централизованным управлением.

Иерахия средств защиты от информационных угроз

Законодательный уровень

Административный уровень

Процедурный уровень

Технический уровень

Рис. 3.1. Многоуровневая модель средств защиты

Общепризнаным является представление множества разных типов средств защиты в виде четырех иерархически организованных уровней, средства каждого из которых могут быть использованы на разных этапах жизненного цикла системы обеспечения информационной безопасности (рис. 3.1).

Законодательный уровень. К нему относятся правовое регулирование, стандартизация, лицензирование и морально-этические нормы, принятые в обществе. При создании системы обеспечения безопасности необходимо учитывать существующие международное и национальные законодательства, а именно, видеть ***возможности,*** вытекающие из гарантированных законом прав, и

■ ***ограничения,*** обусловленные установленными законом обязанностями Законодательство может прямо влиять на концепцию построения системы. Например, выход Закона РФ «О персональных данных», регламентирующего меры по обеспечению безопасности персональных данных при их обработке, потребовал от многих предприятий пересмотра и внесения принципиальных изменений в процедуры и инфраструктуру обработки информации. Не менее принципиальными для продолжения работ могут оказаться требования сертификации средств защиты данных, которые предполагается использовать в проектируемой СОИБ, или необходимость получения лицензии для выбранного вида деятельности.

На протяжении всего жизненного цикла системы, начиная с предпроектной стадии, к проектированию СОИБ должны привлекаться стандартные наработки в виде методик, стандартных требований и практических рекомендаций. Это дает возможность использовать в работе самые передовые, проверенные и одобренные коллегами-профессионалами решения.

Административный уровень. Основу административного уровня составляет политика безопасности, которая определяет стратегические направления информационной защиты предприятия, а именно очерчивает круг критически важных информационных ресурсов предприятия, защита которых представляет наивысший приоритет, предлагает меры, которые могут быть предприняты для устранения или уменьшения связанных с этими ресурсами рисков. На основе найденной стратегии разрабатывается программа обеспечения безопасности ИС, планируется совокупный бюджет, необходимый для выполнения программы, назначаются руководители и очерчивается зона их ответственности.

Процедурный уровень. Средства этого уровня решают задачи, поставленные вышележащим административным уровнем, с использованием технических средств, предоставляемых нижележащим техническим уровнем. В качестве средства процедурного уровня выступает человек, выполняющий взаимосвязанную последовательность действий, направленную на решение той или иной задачи обеспечения безопасности. Любой аспект информационной безопасности предполагает использование средств процедурного уровня. Например, обеспечение нормального режима работы информационной системы осуществляется за счет выполнения множества повседневных процедур: резервного копирования, управления программным обеспечением, профилактических работ и т. п. Другой пример — управление персоналом. Прием, увольнение, организация взаимного контроля, обучение — решение всех этих задач также предполагает выполнение

процедур, реализуемых человеком. Многие процедуры включают в себя использование технических средств. Например, задача учета различных ресурсов (документации, магнитных лент с резервированными данными, программ, оборудования, и др.), как правило, решается с привлечением специально разработанных для этих целей программ.

Технический уровень. К этому уровню относится многочисленный класс технических средств разной природы. Именно этому виду средств защиты в основном посвящена эта книга. Технические средства и методы можно разделить на программные, аппаратные и программно-аппаратные. Программные средства включают защитные инструменты операционных систем (подсистемы аутентификации и авторизации пользователей, средства управления доступом, аудит и др.) и прикладные программы, предназначенные для решения задач безопасности (системы обнаружения и предотвращения вторжений, антивирусные средства, прокси-серверы). Примером аппаратных средств, специализирующихся на информационной защите, являются источники бесперебойного питания, генераторы напряжения, средства контроля доступа в помещения и др. К аппаратно-программным средствам относятся, например, некоторые анализаторы сетевого трафика и межсетевые экраны. И хотя данный уровень средств называется техническим, к нему также относят математические методы (методы криптографии), алгоритмы (эвристический алгоритм расчета времени оборота в протоколе TCP), абстрактные модели (модели контроля доступа) и т. п.

Далее в этой главе будут более подробно рассмотрены средства безопасности всех уровней, кроме самого нижнего — технического, поскольку вопросы, относящиеся к техническому уровню, являются предметом изучения второй, третьей и четвертой частей этой книги.

Законодательный уровень

Законы в области информационной безопасности

Информационные технологии пронизывают нашу жизнь, поэтому совершенно естественно, что правоотношения в данной сфере уже давно регулируются развитым законодательством, в котором немалое внимание уделяется информационной безопасности.

Участниками правоотношений являются ***субъекты правоотношений,*** к которым в области ИБ можно отнести (рис. 3.2):

- обладателей информации;
- потребителей информации;
- операторов информационных систем;
- владельцев интернетовских сайтов;

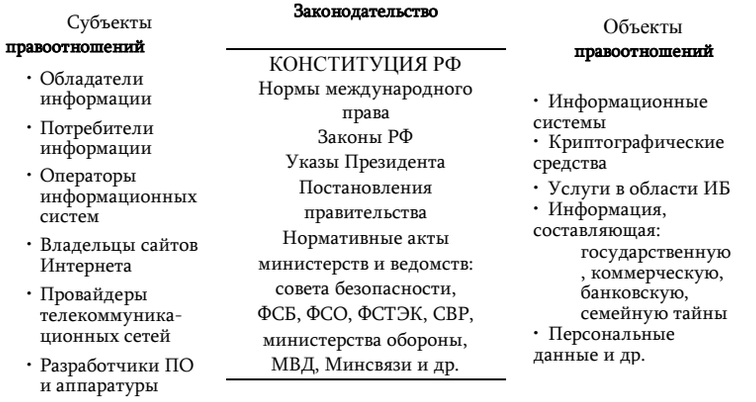


Рис. 3.2. Субъекты и объекты правоотношений в области информационной безопасности

- провайдеров телекоммуникационных сетей;
- разработчиков программных и аппаратных средств информационных технологий и др.

Права и обязанности субъектов правоотношений определяются по отношению ***объектам правоотношений,*** к числу которых могут быть отнесены, например:

- информационные системы (государственные и частные);
- средства защиты информации (алгоритмы шифрования и программы реализующие их, аппаратные ключи, используемые для аутентификации и т. п.);
- услуги, оказываемые в области ИБ;
- информация ограниченного доступа, в том числе информация, составляющая государственную, коммерческую, банковскую, семейную тайну, тайну переписки, персональные данные и др. Необходимость защиты информационных ресурсов и поддерживающей их инфраструктуры диктуется как нашими ***правами*** на защиту (например, каждому гражданину должна быть обеспечена защита от раскрытия его персональных данных), так и ***обязанностями*** ее защищать (каждая организация, оперирующая персональными данными, обязана защищать их от раскрытия). В области ИБ, как и в других областях, реализация прав одного субъекта влечет за собой обязанность соблюдения прав и законных интересов других лиц, в том числе обязанность принимать меры по защите информации.

Правоотношения, регламентируются нормативными правовыми актами (см. рис. 3.2).

Нормативно-правовой акт — это официальный документ, приня

тый компетентным правотворческим органом, устанавливающий общеобязательное государственное предписание, рассчитанное на многократное применение.

Система нормативных правовых актов, действующих на территории страны, принятых законодательным (представительным) органом, называется ***законодательством.***

Законодательство в широком смысле включает Конституцию и общепризнанные принципы и нормы международного права, законы РФ, принимаемые государственной Думой, и подзаконные нормативные акты — указы Президента, постановления Правительства, нормативные акты федеральных органов исполнительной власти (министерств и ведомств) ит.п.

Постановления правительства направлены на реализацию соответствующих федеральных законов. Эту же цель преследуют нормативные акты основных правовых регуляторов: Совета безопасности, ФСБ, ФСО, ФСТЭК, СВР, Министерства Обороны, МВД, Минсвязи и др.

В Приложении 1 приводится краткий обзор нормативно-правовых актов Российской Федерации в сфере информационной безопасности.

Законодательство включает меры ***ограничительно-репрессивного*** характера, направленные на предотвращение нарушений, в том числе путем применения наказаний (например, уголовный кодекс), и меры ***созидательного*** характера, направленные на координацию работ в сфере ИБ, обучение и помощь в создании и использовании средств обеспечения информационной безопасности (например, стандарты).

К числу нарушений законодательства в области ИБ относят как традиционные «компьютерные» преступления, такие как нарушения доступности данных (DDoS-атаки), использование вредоносного ПО, превышение привилегий, несанкционированный доступ и др., так и нарушения регламентирующих правил: отсутствие лицензии на определенный вид деятельности в области защиты информации, использование несертифицированных продуктов там, где это требуется законом (например, средств шифрации при работе с информацией, составляющей государственную тайну).

Определение нарушений и соответствующих наказаний в области ИБ можно найти в Уголовном, Семейном, Гражданском кодексах, кодексе об административно-правовых нарушениях, а также в нормативных актах федеральных органов исполнительной власти.

К сожалению, законодательство в области киберпреступности еще не устоялось — ему всего десять лет. Правовые акты часто не успевают за стремительным развитием информационной сферы. Специалисты находят в новых законах неоднозначность и противоречивость некоторых терминов и положений, неопределенность тою, когда следует применять специально предусмотренные наказания м киберпреступления, а когда — применять другие «традиционные»

6 http://lenta.ru/articles/2013/08/09/kaspersky/

| гатьи уголовного кодекса.

В некоторых случаях специалисты права, соглашаясь с общест- иенным мнением, отмечали неадекватность компьютерных преступлений и соответствующих наказаний. Так, например, в 2012 году братья Попелыши посредством использования фишинга завладели банковскими данными граждан и совершили хищение на сумму в полмиллио- на долларов. Однако в соответствии с нынешним законодательством их преступление не было квалифицировано как кража, так как электронные деньги по закону не являются объектом кражи. В результате первого в истории РФ дела о фишинге преступники были приговорены /шшь к условным срокам за неправомерный доступ к компьютерной информации, а также за использование и распространение вредоносных программ.

Другим примером является дело о DDoS-атаке, проведенной в 2010 году на платежную систему авиакомпании «Аэрофлот». Из-за >той атаки компания понесла убытки почти в 150 миллионов рублей, а преступники получили по два с половиной года тюрьмы. В связи завершением расследования топ-менеджер «Лаборатории Касперского» по юридическим вопросам Игорь Чекунов6 задается риторическими вопросами: «Преступный умысел был направлен на причинение имущественного ущерба посредством DDoS-атаки. А наказываем за что? За неправомерный доступ к охраняемой законом информации. Является это адекватным? Нет. Там же было совершено другое преступление, явно не только информационное. Был там причинен имущественный ущерб? Да. Есть соответствующая статья Уголовного кодекса «Причинение имущественного ущерба» — вот ее и надо применять.» В данном случае мы наблюдаем парадоксальную ситуацию, когда наказание определяется не за то, на что направлен умысел лица, совершающего это преступление (хищение в особо крупном размере), а за инструмент и способ совершения преступления (использование вредоносных программ).

Как правило, расследование компьютерного преступления включает значительную техническую составляющую, поэтому очень важной задачей компьютерной криминалистики является подготовка экспертов и методических рекомендаций для проведения технических судебных экспертиз.

Еще одним характерным примером сложностей применения законов к киберпреступлениям является известное дело против компании Google по поводу сбора ею данных для проекта Street View. Этот проект дал пользователям Интернет замечательную возможность виртуально посетить многие города и посмотреть на улицы и дома глазами пешехода. Для создания такой виртуальной реальности автомобили Google проехали по улицам и переулкам, снимая все на камеру. Однако скоро выяснилось, что сотрудники Google занимались не только съёмкой зданий, но и сбором данных, передаваемых локальными беспроводными сетями, установленными в этих зданиях, в том числе паролей и сообщений электронной почты. В результате во многих странах были возбуждены уголовные дела против Google, некоторые из

которых тянутся до сих пор. Во время одного из таких разбирательств в США адвокаты Google пытались доказать, что сотрудники компании не нарушили американский закон Wiretap Act, потому что собранные данные были подобны данным, передаваемым радиостанциями для публичного потребления, так как радиоволны, которые использовал Google, распространялись за пределы частного жилища. После тщательного рассмотрения аргументов сторон суд отклонил аргументацию Google, и этот конкретный иск все еще находится в стадии рассмотрения.

Стандарты в области информационной безопасности

Важным направлением законодательства является стандартизация.

Стандартизация **—** деятельность, направленная на достижение оптимальной степени упорядочения в определённой области посредством установления положений для всеобщего и многократного использования в отношении реально существующих или потенциальных задач [из закона РФ о техническом регулировании],

Универсальный тезис о пользе стандартизации, справедливый для всех отраслей, особенно справедлив в отношении информационных систем. Основой практически всех информационных систем являются компьютерные сети. Сущностью сети является соединение разного оборудования, а значит, проблема совместимости является здесь одной из наиболее острых. Без согласования всеми производителями общепринятых стандартов для оборудования и протоколов прогресс в деле «строительства» сетей был бы невозможен. Поэтому все развитие компьютерной отрасли вообще и средств информационной безопасности в частности, отражено в стандартах,—любая новая технология только тогда приобретает «законный» статус, когда ее содержание закрепляется в соответствующем стандарте.

Закон РФ N 65-ФЗ «О техническом регулировании» определяет понятие «стандарт» следующим образом:

Стандарт — это «документ, в котором в целях добровольного многократного использования устанавливаются характеристики продукции, правила осуществления и характеристики процессов проекти- цппания (включая изыскания), производства, строительства, монта- •н,|. наладки, эксплуатации, хранения, перевозки, реализации и утилизации, выполнения работ или оказания услуг. Стандарт также мочит содержать правила и методы исследований (испытаний) и изме- ргний, правила отбора образцов, требования к терминологии, символике, упаковке, маркировке или этикеткам и правилам их нанесения».

По умолчанию ***соблюдение стандарта не является обязательным*** (н< ли явно не указана обязательность его исполнения). Однако сущес- тует множество причин, по которым большинство компаний, предприятий, частных лиц, организаций добровольно выбирают следование стандартам. Мы уже говорили о стандартизации как средстве обеспечения совместимости информационных технологий, продуктов и терминологии. Следование стандартам также позволяет создавать более качественные, более

конкурентоспособные технологии, системы и услуги, так как стандарты — это концентрированное выражение передовой технической мысли, они аккумулируют актуальные теоре- (ические знания и так называемые «лучшие практики».

Стандарты дают специалистам единое средство общения, язык, который они понимают одинаково. Это особенно важно в тех случаях, когда решается проблема, лежащая на стыке разных отраслей знаний. Такая ситуация возникает, например, при анализе рисков предприятия, когда в одной рабочей группе оказываются специалисты по компьютерным технологиям и бизнес-процессам. Именно это является одной из причин, по которой так востребованы стандартные методики управления рисками.

Стандартные процедуры оценки систем дают возможность их со-поставления и сравнения, на основании чего может выполняться ***сертификация*** систем на соответствие определенным требованиям. Сертификация информационных систем на соответствие требованиям безопасности регулируется достаточно обширным набором стандартов. В главе 18 мы подробно рассмотрим самые известные из них — «Оранжевую книгу» и «Общие критерии».

Стандарты регулируют самые различные сферы и аспекты обеспечения информационной безопасности, в том числе теоретические концепции и алгоритмы, требования к программным и аппаратным средствам, методики обследования систем и порядок документирования результатов, административные процедуры и вопросы обучения персонала. Стандарты часто образуют «семейства», члены которого объединены неким общим признаком, например они относятся к

одной предметной области, но описывают ее с разной степенью детализации. Работы по стандартизации вычислительных сетей и средств безопасности ведутся большим количеством организаций. В нашей стране главную организационную роль в стандартизации играет ***Федеральное агентство по техническому регулированию и метрологии (Росстандарт).*** Росстандарт создает и координирует рабочие группы по разработке стандартов, организует общественное обсуждение и экспертизу новых стандартов, утверждает и публикует документы по стандартам, ведет учет и распространение национальных стандартов.

В законе РФ о техническом регулировании говорится, что разработчиком стандарта может быть любое лицо, но, как правило, стандарты разрабатываются рабочими группами (техническими комитетами), в состав которых на добровольной основе могут включаться представители органов исполнительной власти, научных, коммерческих и некоммерческих организаций, общественных объединений. Одним из основных принципов стандартизации является ориентация на тех лиц, кто в наибольшей степени заинтересован в существовании стандартов. Поэтому очень часто разработчиками стандартов являются компании и организации, много и успешно работающие в той области, для которой они предлагают стандарты.

Стандарты информационной безопасности могут быть отнесены к одной из следующих групп.

- ***Международные стандарты,*** основными разработчиками которых в области информационных технологий являются Международная организация по стандартизации (International Organization for Standardization, ISO), Международная электротехническая комиссия (International Electrotechnical Commission, IEC) и Международный союз электросвязи (International Telecommunication Union, ITU).
- ***Национальные стандарты.*** К ним относятся, например, государственные стандарты Российской Федерации, ГОСТы. Международный авторитет имеет национальный орган стандартизации — Британский институт стандартов (British Standards Institution, BSI). Большое количество общепризнаных стандартов вышло под эгидой неправительственной некоммерческой организации США Национального института стандартов и технологий (National Institute of Standards and Technology, NIST).
- ***Стандарты специальных*** комитетов и объединений. К таковым относится, например, разветвленная система стандартов консорциума W3C (World Wide Web Consortium), который объединяет крупнейшие мировые компании и корпорации. Этот консорциум занимается стандартизацией Web-технологий, включая решения по обеспечению безопасности этого сервиса.
- ***Корпоративные стандарты отдельных компаний.*** Широкое распространение получили, например, корпоративные стандарты компании Microsoft в области создания безопасного программного обеспечения, управления рисками и др.

Некоторые стандарты, непрерывно развиваясь, могут переходить и 1

одной категории в другую. В частности, корпоративные стандарты на продукцию, получившую широкое распространение, обычно становятся международными стандартами де-факто, так как вынуждают производителей из разных стран следовать корпоративным стандар- |,ш, чтобы обеспечить совместимость своих изделий с этими популярными продуктами. Например, из-за феноменального успеха пер- | опального компьютера компании IBM фирменный стандарт на архи- мжтуру IBM PC стал международным стандартом де-факто.

Более того, ввиду широкого распространения некоторые корпоративные стандарты становятся основой для национальных и международных стандартов де-юре. Например, стандарт Ethernet, первоначально разработанный компаниями Digital Equipment, Intel и Xerox, через некоторое время и в несколько измененном виде был принят как национальный стандарт IEEE 802.3, а затем организация ISO утвердила его в качестве международного стандарта ISO 8802.3.

Административный уровень. Политика безопасности

Определение политики безопасности

В любом целенаправленном деле наличие достаточных материальных ресурсов — это только необходимое, но недостаточное условие; для достижения цели необходимо выработать осмысленный план действий. Так и при построении информационной защиты нельзя целиком полагаться на технические средства — никакие самые современные файерволы, системы обнаружения вторжений, сканеры уязвимостей, централизованные серверы аутентификации не защитят организацию, если не будет выработано руководящей идеи, которая превращает набор отдельных мощных, но часто неэффективно используемых инструментов и методов, в интегрированную систему, работающую на достижение общей цели. Если учесть, что для успешной реализации основополагающей идеи необходимо с самого начала располагать хотя бы самыми общими соображениями, в каком направлении двигаться, то мы приходим к понятию «политика».

В самом широком смысле (безотносительно к информационным технологиям и вообще к технике) понятие «политика» может быть определено следующим образом:

- ***Политика*** — это общее руководство, которое устанавливает главные направления, в которых нужно действовать, чтобы наиболее рациональным путём достичь поставленной цели. Содержание политики выражается в ее целях, программах и ценностях, в проблемах и задачах, которые она решает, в мотивах, механизмах, способах и методах принятия и реализации решений.

Сфера приложения политики может быть любая целенаправленная деятельность. Мы здесь рассматриваем *политику информационной защиты предприятия,* которую будем называть сокращённо «*политикой безопасности»* (ПБ). В сфере информационных технологий используются политики и в других более узких сферах, например политика информационной защиты компьютерных систем предприятия, политика использования средств коммуникаций, политика использования корпоративной электронной почты и т. п.

Как и любая политика, ПБ призвана играть организующую и дисциплинирующую роль. Процесс выработки ПБ приводит к более ясному осознанию целей и путей построения СОИБ.

Политика безопасности разрабатывается с привлечением высшего руководства. Тем самым руководители демонстрируют свою поддержку предлагаемой стратегии обеспечения информационной безопасности, что имеет большое значение, так как ее реализация может потребовать привлечения значительных финансовых средств и ресурсов предприятия.

Будучи принятой, политика безопасности становится «законом», обязательным для исполнения всеми сотрудниками предприятия. Персонал предприятия должен быть ознакомлен с положениями ПБ, в том числе с ответственностью, которая определена за ее нарушения.

Политика безопасности фиксируется в документах. Часто под политикой понимают именно ее *документальное выражение.* Так, например, ГОСТ 50922-2006 дает следующее определение:

- ***«Политика безопасности.*** Совокупность документированных правил, процедур, практических приемов или руководящих принципов в области безопасности информации, которыми руководствуется организация в своей деятельности.»

Поскольку целевая аудитория политики безопасности в общем случае состоит не только из специалистов в области информационной защиты, то документы ПБ должны быть изложены достаточно простым языком, без использования узкоспециальных терминов и технических подробностей.

Как правило, изложение ПБ представляет собой иерархически организованный набор документов разного объема, с разной степенью детализации описывающих стратегические решения, руководящие директивы, инструкции и пр.

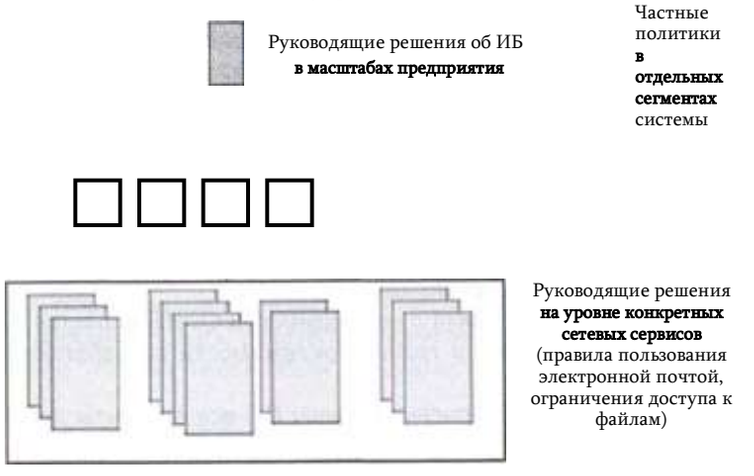


Рис. 3.3. **Иерархическое представление документации политики безопасности** обеспечения ИБ (например, политика доступа сотрудников в Интернет)

Решения, которые должны быть приняты в рамках разработки политики безопасности предприятия, а также документы, их описывающие, могут быть отнесены к одному из следующих уровней (рис. 3.3):

- *верхний уровень* — решения, затрагивающие предприятие в целом, они принимаются высшим руководством, носят общий характер, могут быть описаны компактным документом;
- *средний уровень* — решения, относящиеся к отдельным (частным) аспектам обеспечения информационной безопасности:
- *нижний уровень*— решения, касающиеся регламентации отдельных сервисов.

Каждый документ более высокого уровня раскрывается и дополняется одним или несколькими документами более низкого уровня (см. рис. 3.3). Чем выше уровень документа, тем более компактным и декларативным он является. Граница между уровнями является условной, так, например, документы среднего уровня ПБ, описывающие частные политики, могут частично включать в себя политики защиты сервисов, т. е. политики нижнего уровня.

Верхний уровень политики безопасности

К верхнему уровню относят решения и соответствующие документы самого общего характера.

Первым шагом на пути к разработке политики безопасности должно стать принятие руководством предприятия принципиального решения о том, что оно действительно нуждается в системе обеспечения информационной безопасности. Разные предприятия имеют разные побуждающие причины для внедрения новой или пересмотра

текущей СОИБ. Это может быть, например, появление нового закона, ужесточающего требования к защите информации, например закона о защите персональных данных, которое вынуждает предприятие изменить бизнес-процессы. Или же — условие, выдвинутое партнерами по бизнесу, о том, что предприятие должно пройти сертификацию на соответствие некоему стандарту безопасности, а это невозможно при текущем состоянии СОИБ. В любом случае политика безопасности должна включать аргументированное обоснование необходимости информационной защиты предприятия.

После констатации факта о ***необходимости СОИБ*** решается очень важная задача определения ***границ*** приложимости разрабатываемой политики безопасности:

Должны ли быть включены в сферу ПБ все сегменты предприятия?

Или следует ограничится только самыми критичными подсистемами?

Надо ли учитывать компьютеры сотрудников, работающих из дома?

От того, какое решение принято на данном этапе, существенно зависит то, какой будет система обеспечения безопасности.

Когда границы определены, выполняется всестороннее ***обследование*** предприятия: выявляются критически важные активы, более всего нуждающиеся в защите, устанавливаются правила разграничения доступа к информационным ресурсам, определяются наиболее вероятные угрозы, оцениваются возможные потери, принимаются концептуальные решения относительно методов обеспечения защиты. То есть значительная часть политики безопасности базируется на результатах анализа рисков.

На верхнем уровне принимаются организационно-административные решения, а именно определяются должностные позиции (роли), создаются административные подразделения, комитеты, рабочие группы, функциями которых является проведение политики безопасности в жизнь, устанавливаются границы ответственности всех этих административных единиц.

Политика безопасности верхнего уровня может содержать указания на приверженность предприятия тем или иным стандартам и нормативно-правовым актам, принципы обучения персонала, порядок реагирования на нарушения режима безопасности и др.

Высокоуровневая часть политики безопасности описывается лаконично и может быть представлена документом, состоящим всего из 2-5 страниц. Краткость достигается за счёт ссылок на другие информационные ресурсы: стандарты и законы, частные политики и директивы и т. п.

Средний уровень политики безопасности

Этот уровень называют также уровнем *частных политик,* так как к нему относят решения и соответствующие документы, касающие- | ч частных аспектов информационной безопасности таких, например, мк политика использования средств криптографической защиты, по- мигика антивирусной

защиты, политики мониторинга и менеджмента инцидентов информационной безопасности, политика защиты коммуникационных каналов, политика физической защиты и др. По сравнению с верхним уровнем разработка политики среднего уровня требует большего участия технических специалистов (из числа руководителей).

Обычно в частной политике имеется упоминание о запрещенных действиях и наказаниях за них. Например, в рамках частной политики компьютерной инфраструктуры предприятия предусматриваются меры, защищающие сеть от вирусов и других вредоносных программ.

Для этого ПБ устанавливает запрет сотрудникам использование стандартного и поддерживаемого предприятием программного обеспечения, для чего следует включить в политику безопасности следующее:

- указание на то, что ПО перед его использованием должно тестироваться сотрудниками отдела безопасности;
- запрет персоналу устанавливать и запускать ПО без предварительного разрешения отдела безопасности;
- требование регулярного обновления всего используемого на корпоративных компьютерах ПО.

С этой же целью необходимо регламентировать использование в корпоративной сети собственного оборудования сотрудников — ноутбуки, планшетов, портативных переносных USB-накопителей, флэш- карт и т. п. Несанкционированное подключение устройств памяти помимо угрозы заражения повышает *угрозу утечки данных,* поэтому в организациях с повышенными требованиями к секретности вводится полный запрет на использование любых устройств (в том числе телефонов с внешним доступом, смартфонов с фотокамерой), с помощью которых преступник может вынести (или передать) секретные данные за пределы организации. Например, с 2008 года такой запрет был введен во всех организациях Министерства обороны США, нарушение этого правила влечет за собой немедленное увольнение7.

7 Information Week Security, Mathew J. Schwartz, «Thumb Drive Security: Snowden 1, NSA 0».

Как же соотносится общепринятая практика запрета использования USB-памяти с тем широко известным фактом, что именно с помощью устройства такого типа контрактник Сноуден похитил 700 тысяч суперсекретных документов Национального агентства безопасности США? Коллеги Сноудена говорят, что в общем случае такой запрет включен в политику безопасности агентства, однако «всегда есть исключения». Есть некоторые люди, которым по работе требуется использовать USB-память, и ради них делаются исключения. На одной из таких должностей — системного администратора — работал по контракту Сноуден. За последние годы подобные случаи происходили не раз. Так, в 2012 году аналогичным образом, пользуясь привилегиями системного администратора, сотрудник швейцарской разведывательной службы вынес на портативном накопителе секретную информацию, которая была передана Швейцарии ан- титеррористическими службами США и Великобритании. Все это говорит о том, что политика безопасности должна быть законом, обязательным для всех, без исключений.

Документы, описывающие частные политики, не имеют жесткого формата, их содержание зависит от специфики предприятия, от того, какой аспект СОИБ они затрагивают. Частные политики иногда играют роль корпоративных стандартов и могут являться конфиденциальными. Далее, в разделе «Пример политики безопасности» приводится документ, описывающий принципы и подходы к защите компьютерной инфраструктуры предприятия.

Нижний уровень политики безопасности

Политики нижнего уровня определяют действия по обеспечению безопасности на уровне сетевых сервисов и могут представлять собой руководства, инструкции, регламенты и правила, связанные с администрированием и использованием сервисов.

Выше рассматривался пример политики среднего уровня в отношении защиты компьютерной сети от вирусов и других вредоносных программ. Среди прочего, там были перечислены правила, регламентирующие использование в корпоративной сети программного обеспечения. Однако каждое из этих правил оставляет много вопросов.

Кто и как учитывает программное обеспечение? Какие методики должны использоваться при тестировании? Должны ли тестироваться программы, разработанные специалистами компании? Как надо интерпретировать результат тестирования?

В какой форме надо подавать запрос на разрешение? Кому и при каких условиях может быть дано разрешение? Должны ли быть установлены ограничения на время получения разрешения?

Что конкретно означает «регулярное» обновление? Кто должен его выполнять?

Ответы на большинство этих и подобных им вопросов должны содержаться в политиках безопасности нижнего уровня. В отличие от верхних двух уровней, многие положения которых носят общий характер, на нижнем уровне приводятся более специфические, более де- гальные, более формализованные рекомендации, учитывающее особенности конкретного сервиса.

С другой стороны, частные политики безопасности нижнего уровня нельзя смешивать с обычными инструкциями. В первом случае объекты регламентации — процессы использования базы данных, файлового сервиса,

корпоративной телефонной связи, электронной почты, антивирусных программ и т. д. и т. п. — оказывают непосредственное влияние на безопасность предприятия и поэтому должны разрабатываться централизовано, с участием руководства. Во втором случае инструкцию может составить любой специалист, обладающий достаточным для этого уровнем профессиональных знаний.

Пример политики безопасности

Реально существующее предприятие, которое мы здесь будем условно называть *NNN,* является интернет-провайдером национального масштаба, предоставляющим услуги академическим организациям с помощью собственной глобальной сети *NNET* (также условное название). Предприятие определило собственную Политику безопасности, которая отражена в комплексе документов.

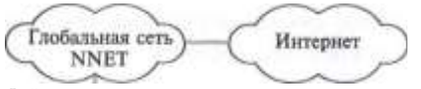
Политика безопасности предприятия (верхний уровень) описана в четырехстраничном документе, преамбула которого приведена ниже.

«Эффективность работы компании будет тем выше, чем меньше ограничений будет установлено на разработку или использование новых приложений и сервисов. Использование строгих процедур мандатного доступа, скорее всего, затруднит текущее использование сети, а также может вызвать проблемы в будущих проектах, поэтому этот режим следует рассматривать только в тех случаях, где это явно необходимо. Ограниченный режим доступа к сети может быть предложен в качестве опционального сервиса для присоединяемых сетей организаций-клиентов, однако основная сеть NNET должна предоставлять настолько открытый доступ, насколько это позволяют производственные и законодательные ограничения.

Однако презумпция открытости привносит в работу NNN связанные с ней риски. Инциденты нарушения защиты или неправильное использование могут серьезно снизить эффективность работы сети [ссылка на обзор рисков, приведенный в приложении к документу]. *Результаты инцидентов могут быстро распространиться на всю сеть NNET далеко за пределы той организации-клиента или компьютера, которые явились источником этого нарушения. Чтобы сеть NNET могла выполнять свои задачи, необходимо предусмотреть реакцию на эти риски. С этой целью предприятие NNN принимает данную Политику безопасности, чтобы защитить сеть NNET и организации, ее*

Технологии:

- MS Active Directory
- CitrixAnywhere
- Токены RSA
- Sophos Anti-Virus
- Outlook, Entourage
- Staffmail



Файервол

«|» Экспериментальная проводная сеть

«2» Беспроводная сеть университетского роуминга Eduroam

использующие. Требование соответствия положениям данной Политики являются обязательными для всех организаций, подключенных к сети NNET, а также для всех пользователей сети NNET.»

Компьютерная инфраструктура предприятия NNN состоит из двух проводных и трех беспроводных локальных сетей (рис. 3.4):

- проводная производственная сеть, к которой подключены все компьютеры пользователей и серверы предприятия, поддерживающие бизнес предприятия;
- экспериментальная проводная сеть, к которой подключены компьютеры и серверы, а также коммутаторы и маршрутизаторы, используемые для тестирования и отладки новых технических решений и конфигураций;
- беспроводная производственная сеть, к которой могут подключаться ноутбуки сотрудников;
- беспроводная гостевая сеть;
- беспроводная сеть университетского роуминга, позволяющая студентам из любого университета, участвующего в проекте Eduroam, логически входить в сеть с помощью учетной записи своего университета.

Все локальные сети соединены с Интернет через глобальную сеть предприятия NNET.

Комплекс документов включает также частные политики предприятия NNN, одной из которых является «Политика безопасности компьютерной инфраструктуры» (политика среднего уровня)*.

«Политика безопасности компьютерной инфраструктуры»
Классификация документа — неклассифицированный Введение

Целью этого документа является установление политики безопасности компьютерной инфраструктуры предприятия NNN и вы-

Здесь приведен перевод с английского полного текста реального документа за исключением описания процедуры его обновлений.

Беспроводная гостевая сеть

Беспроводная производственная сеть (((*

Проводная производственная сеть

Рис. 3.4. **Схема компьютерной инфраструктуры предприятия NNN**

мча ***рекомендаций пользователям в виде лучших практик, которые они должны принять во внимание при использовании корпоративных компьютеров, администрируемых подразделением «ИТ- поддержки».***

Учитывая, что большое количество должностей на предприятии валяются инженерно-техническими, политика устанавливает минимально возможный набор ограничений, создающих как можно меньше помех для производственной деятельности.

Эта политика относится ко всему персоналу и нацелена на то, чтобы:

- ***поощрить сотрудников действовать ответственно и создать все условия для их соответствующего поведения;***
- ***поддержать усилия компании в выполнении ее основной бизнес-задачи — обеспечение контролируемого доступа к сети NNET;***
- ***защитить ресурсы компании от атак и предотвратить организацию атак с их помощью;***
- ***создать базис, способствующий расследованию любых инцидентов, связанных с безопасностью.***

Персонал также должен следовать правилам документа «Политика безопасности предприятия NNN» (ссылка на документ).

Безопасность компьютеров

1. Учетные записи пользователей и разрешения

Доступ к корпоративным компьютерам, а также к некоторым компьютерным сервисам контролируется на основе учетных записей пользователей, с которыми связаны определенные права пользователей. Эти права описаны в рабочей инструкции «Права пользователей корпоративных компьютеров (указан номер инструкции). ***Уровень прав, назначаемых каждому пользователю, является таким, чтобы позволить пользователю выполнять свои служебные обязанности без нанесения ущерба безопасности.***

2. Шифрование данных, хранящихся на компьютерах и других внешних носителях

Отдел ИТ-поддержки обеспечивает защиту данных на всех новых корпоративных ноутбуках за счет шифрации их на жестком диске, чтобы предотвратить неавторизованный доступ к этим данным. Сотрудники предприятия, имеющие компьютеры без зашифрованных данных на жестком диске, притом, что эти компьютеры относятся к категории, которые требуют шифрации, должны обращаться в корпоративную службу «Поддержка сервисов».

Отдел ИТ-поддержки также может снабдить сотрудников компании, которым требуется переносить чувствительные данные, специальными портативными переносными накопителями (memory

stick, MS) с аппаратной поддержкой шифрации. Сотрудник предприятия, которому нужно MS с аппаратной шифрацией, должны обращаться в корпоративную службу «Поддержка сервисов».

3. Обязанности персонала

- *Сотрудник должен блокировать свой компьютер в тех случаях, когда он отлучается со своего рабочего места, чтобы предотвратить неавторизованный доступ к программам и данным.*
- *Персонал не должен оставлять компьютерное оборудование в потенциально опасных ситуациях (например, оставлять ноутбук на видном месте в припаркованном автомобиле).*
- *Сотрудники не должны загружать данные в корпоративные компьютеры из MS или других переносных носителей, если они не уверены в их происхождении.*

Безопасность сетевых сервисов

Доступ к сетевым сервисам таким, например, как файловый сервис, контролируется с помощью инфраструктуры идентификатора пользователя и его пароля системы справочной системы Microsoft Active Directory. Доступ с помощью беспроводных сетей дополнительно защищается с помощью технологии 802.1x.

Некоторые сервисы, такие как база данных клиентов предприятия, требуют дополнительных идентификаторов и паролей. Услуги виртуальных частных сетей VPN и удаленного управления Citrix Anywhere требуют использования токена B.SA, который обеспечивает строгую двухфакторную аутентификацию.

1. Подсоединение к корпоративным сетям

Только корпоративные компьютеры и IP-телефоны должны присоединяться к проводным и беспроводным производственным сетям. Гости и люди, работающие по контрактам, а также сотрудники, использующие некорпоративные компьютеры, не должны пользоваться производственными сетями (как проводными, так и беспроводными или VPN) без предварительного разрешения отдела ИТ-подцержки. Если им вдруг понадобится доступ в Интернет, то они должны использовать гостевую беспроводную сеть или беспроводную сеть Eduroam.

2. Подсоединение к сторонним сетям

Персонал должен проявлять осторожность при присоединении своих компьютеров к сторонним сетям (проводным или беспроводным), особенно неизвестным или незащищённым сетям, предпочтительно использовать сеть Eduroam, если она доступна, и использовать сервис VPN, чтобы повысить безопасность соединения. Безопасность телефонной связи

Предусмотрено два режима использования настольных телефонов. Один используется для внутренних соединений и является незащищенным. Второй режим применяется для связи с внешними абонентами и защищен процедурой логического входа пользователя, требующей ввода идентификатора и пароля. Покидая свое рабочее место, сотрудники должны выполнять процедуру логического выхода.

Телефоны в местах общего доступа (холлах, комнатах для совещаний и т. п.) не позволяют делать международные звонки и звонки с высоким тарифом оплаты. Такое же ограничение имеет гостевая учетная запись настольных телефонов сотрудников.

Доступ к электронной почте

Доступ к электронной почте контролируется с помощью инфраструктуры идентификатора пользователя и его пароля системы справочной системы Microsoft Active Directory; для доступа можно использовать клиентские программы Outlook, Entourage или Staff- mail (Outlook Web App).

1. Электронные календари

Все электронные календари сотрудников должны быть всегда открыты для просмотра другими сотрудниками.

2. Фильтрация вирусов и спама

Все сообщения электронной почты, отправляемые внешним адресатам или получаемые от них, автоматически проверяются антиспамовыми и антивирусными фильтрами. Все электронные письма, содержащие ненадлежащий контент, могут быть помещены в карантин или заблокированы.

3. Обязанности персонала

- *Персонал должен выполнять проверку на вирусы всех получаемых по электронной почте файлов, используя щелчок правой клавишей мыши и выбирая соответствующую опцию;* в *результате вызывается программа SophosAnti- Virus. Кроме того, сотрудники должны уведомлять службу поддержки сервисов о подозрительном поведении компьютера.*
- *Персонал должен обращаться максимально осторожно со всеми файлами, полученными от неизвестных отправителей.*
- *Персонал не должен рассматривать электронную почту как безопасное средство передачи данных — если необходимо отправить защищенным образом электронное письмо, то отдел ИТ-поддержки может обеспечить сотрудника по его запросу программой шифрования электронных сообщений.*

Безопасность Интернета

1. Доступ в Интернет

Доступ в Интернет осуществляется через глобальную сеть NNET и предоставляется всем корпоративным компьютерам с минимальными ограничениями.

2. Обязанности персонала

- ***Сотрудники должны действовать с осторожностью при открытии страниц тех веб-сайтов, которые они не используют регулярно, а также не посещать сайты с неподобающим контентом.***
- ***Каждый сотрудник должен выполнять проверку всех загружаемых из Интернета файлов на вирусы, используя щелчок правой клавишей мыши и выбирая соответствующую опцию; в результате вызывается программа SophosAnti-Virus. кроме того он должен уведомлять службу поддержки сервисов о подозрительном поведении своего компьютера.***

Файерволы и антивирусные программы

1. Использование файерволов и антивирусных программ

На всех корпоративных компьютерах (настольных, ноутбуках, серверах) установлены антивирусные программы и программные файерволы. Инструкции по использованию этих средств размещены на корпоративном сайте на странице отдела ИТ-поддержки.

Антивирусное программное обеспечение использует автоматические обновления, чтобы обеспечить защиту от самых последних угроз, и постоянно сканирует все данные, обрабатываемые компьютером, а также выполняет регулярные сканирования всего компьютера. Все обнаруженные вирусы, а также вредоносные программы могут быть удалены или отправлены в карантин. При этом отдел ИТ-поддержки автоматически уведомляется о таких событиях.

Корпоративные сети защищены файерволом, который работает на основе пакетной фильтрации.

2. Обязанности персонала

- ***Персонал должен немедленно сообщать о любом подозрительном поведении компьютера в службу поддержки сервисов предприятия Это относится к сообщениям любого программного обеспечения, кроме корпоративной антивирусной программы, которая делает это автоматически.***
- ***Сотрудники не должны пытаться отключать антивирусные программы, файерволы и все другие программы обеспечения безопасности, если только они не уполномочены на это отделом ИТ-поддержки. Любые такие попытки будут рассматриваться как нарушения данной политики безопасности***

Резервное копирование

Соглашение об уровне сервиса (SLA) отдела ИТ содержит информацию о том, какие данные должны резервироваться для того, чтобы обеспечить их восстановление в случае потери. Резервные данные хранятся в различных местах и передаются туда зашифрованными.

Мобильные телефоны и планшеты

Те мобильные телефоны и планшеты, которые имеют доступ к ресурсам предприятия, должны быть заблокированы на то время,

когда они не используются для этих целей.

Такие устройства, по возможности, должны поддерживать удаленное стирание данных в случае потери устройства.

Использование личного оборудования

Если сотрудники используют личное оборудование для доступа к ресурсам компании, то это оборудование не должно использоваться для хранения любых чувствительных данных и должно блокироваться, когда не используется.

Персонал не должен подключать свое личное оборудование к производственным сетям предприятия без предварительного разрешения отдела ИТ-поддержки.

Защита паролей

Идентификаторы и пароли пользователей используются для контроля доступа к некоторым элементам компьютерной инфраструктуры предприятия, и, следовательно, важно, чтобы персонал управлял своими паролями эффективно.

Сотрудники всегда должны изменять свои пароли от их значений по умолчанию.

В документе «Использование паролей» [ссылка на документ] ***даются рекомендации о том, как выбирать и пользоваться паролями.***

Обязанности персонала

Сотрудники не должны раскрывать друг другу пароли; они несут ответственность за любые действия, выполненные с использованием их учетной записи.

Безопасность службы печати

В том случае, когда необходимо распечатать чувствительные документы, необходимо посылать документ в специальную защищенную очередь, которая гарантирует, что документ будет распечатан только в случае физического присутствия пользователя. Что подтверждается набором его личного PIN-кода на пульте принтера.

Безопасность программного обеспечения

1. Стандартное и поддерживаемое программное обеспечение

Отдел ИТ-поддержки ведет перечень стандартного и поддерживаемого программного обеспечения, которое было протестировано на предмет безопасности.

2. Использование другого программного обеспечения

Сотрудники не должны запускать или устанавливать программное обеспечение без предварительного разрешения отдела ИТ-поддержки за исключением обстоятельств, определенных рабочей инструкцией «Права пользователя» [ссылка на документ]. Если сотруд

ники отдела ИТ-поддержки обнаружат программу, запущенную без разрешения, то они могут блокировать ее ***работу, удалить ее из компьютера и возбудить дисциплинарную процедуру нарушения политики безопасности.***

3. Обновление ПО

Любое ПО, используемое на корпоративных компьютерах, должно регулярно обновляться для того, чтобы защищаться от любых новых угроз. Во всех возможных случаях отдел ИТ-поддержки будет автоматизировать процесс обновления ПО (например, используя функциональность Windows update) в соответствии с инструкцией «Обновления программного обеспечения» (ссылка на документ).

Предотвращение краж и восстановление данных

1. Предотвращение физических краж.

Ключевые элементы компьютерной инфраструктуры предприятия защищены от кражи и неавторизованного доступа путем размещения их в комнатах с доступом через контролируемые двери, которые позволяют физический доступ только авторизованному персоналу. Все случаи доступа журнализируются, в таких комнатах может выполняться мониторинг камерами наблюдения.

Некоторые элементы компьютерной инфраструктуры снабжаются замками, ключи от которых имеют только сотрудники отдела ИТ-поддержки.

2. Отслеживание краж

Некоторое компьютерное оборудование защищено технологиями поиска украденного, которые позволяют определить географическое положение оборудования и удаленно стереть записанные на нем данные. Отдел ИТ-поддержки работает совместно с полицией для того, чтобы отследить украденное оборудование и возбудить против преступников уголовное дело.

Списание оборудования

Данные, хранящиеся в устройствах, подлежащих списанию, должны быть стёрты, если это возможно.

Регистрация и расследование инцидентов, связанных с нарушениями безопасности

Любые инциденты или даже подозрения на них, связанные с нарушениями безопасности, о которых было доложено корпоративной службе безопасности, должны расследоваться с наивысшим приоритетом. В процессе расследования отдел ИТ-поддержки должен работать с группой компьютерной безопасности, сторонними организациями или с полицией.

В случае серьёзного нарушения безопасности отдел ИТ-поддержки может остановить любой компьютерный сервис для того, чтобы защитить другие сервисы и предотвратить нанесение ущерба компьютерной инфраструктуре предприятия и сети NNET (а также репутации предприятия).

Отдел ИТ-поддержки осуществляет мониторинг и журнализацию некоторых видов пользовательской активности.

Взаимодействие с персоналом

Отдел ИТ-поддержки взаимодействует с персоналом по поводу инцидентов безопасности и рекомендуемых приемов безопасности с помощью электронных писем и других средств.

Сотрудники должны докладывать немедленно о любых инцидентах корпоративной службе безопасности (e-mail и телефон).

Дисциплинарные процедуры

Сотрудники, замеченные в нарушениях политики безопасности, подвергаются дисциплинарным наказаниям, определенным в [ссылка на документ].»

Процедурный уровень

Как мы уже обсудили, главной движущей силой создания безопасной системы является стратегический план (политика), который затем детализируется в более частные планы. Таким частным планом (политикой) может быть, например, регламентация того, какие программы сотрудники могут запускать у себя на работе, а какие нет. Пусть, например, в этой частной политике сказано, что «сотрудники предприятия имеют право загружать и выполнять только те программы, которые были протестированы и разрешены к использованию». Чтобы воплотить данное (принятое на административном уровне) решение о правилах использования ПО в жизнь, предприятию обязательно потребуется человек, который будет:

- в соответствии с данными ему инструкциями вести учет программ, разрешенных для выполнения пользователям корпоративной сети;
- в соответствии с порядком, принятым на предприятии, принимать новые программы и передавать их на тестирование, а затем в зависимости от результата включать или не включать их в список разрешенных;
- в соответствии с определенной процедурой принимать запросы от сотрудников, которым необходимо разрешение на использование нестандартного ПО;
- решать на основании некоторых заранее определенных правил, может ли быть просьба удовлетворена;
- используя некоторую процедуру, фиксировать нарушение и передавать сведения о нарушителе в соответствующее подразделение.

То есть задача правильного использования ПО, сформулированная в политике, может быть решена только человеком, действующим в соответствии с определёнными инструкциями. Абсолютно такие же рассуждения могут быть приведены относительно других задач административного уровня. Таким образом, мы приходим к выводу, что следующий шаг на пути к безопасной системе — разработка процедур, реализуемых с помощью универсального инструмента — человека.

В общем случае ***«процедура»*** — это взаимосвязанная последовательность действий, направленная на решение некоторой задачи.

Процедура может быть формальной, как, например, при ее представлении в виде алгоритма или программы на языке программирования, так и неформальной, в виде в той или иной мере расплывчатых указаний. Формальные процедуры могут быть реализованы механическими или электронными устройствами, но только человек может действовать в условиях, когда невозможно абсолютно однозначно определить все детали процедуры, а именно такими являются большая часть процедур, связанных с поддержанием работоспособности системы, управлением персоналом, физической защитой, управлением документацией и др.

Именно поэтому между слоем стратегий и слоем технических средств (способных реализовывать только формальные процедуры) с необходимостью появляется промежуточный слой — *слой неформальных процедур, приводимых в действие человеком.* Например, мы относим формальную программно-аппаратную процедуру автоматического подключения резервного источника питания к техническому уровню, а инструкцию, описывающую действия оператора, подключающего этот же прибор вручную, — к процедурному уровню. Исполнителями процедур являются ИТ-специалисты, сотрудники отделов информационной безопасности, пользователи и другие сотрудники, связанные с информационной защитой.

Процедуры могут иметь иерархическую структуру, при которой одна процедура включает в себя другие, вложенные процедуры. Например, общая процедура резервного копирования может ссылаться на другие документы, содержащие инструкции по выполнению следующих, более частных задач: создание резервной копии, контроль резервных копий, ротация носителей, восстановление информации из резервных копий. В общем случае каждая из вложенных процедур может выполняться разными сотрудниками.

Процедуры управления персоналом,

Управление персоналом включает подбор персонала, прием на работу, увольнение, текущий контроль и др. Каждое из перечисленных действий имеет отношение к безопасности. При подборе работников следует проверять прошлое кандидатов, их рекомендации, в некоторых случаях требуется дополнительная проверка профессиональных сертификатов, кредитной истории, записей в базах данных о преступниках и другие более тщательные проверки.

В процедуре приема на работу должно быть предусмотрено ознакомление сотрудника с мерами ответственности за нарушения правил информационной безопасности. В частности, работник должен официально подтвердить свое согласие следовать политике безопасности предприятия и нести ответственность за ее нарушения, подписать соглашение о неразглашении конфиденциальных данных. Такой порядок помогает разрешать потенциальные конфликтные ситуации на правовой основе.

Особое внимание служба безопасности должна уделять процедуре увольнения, так как по статистике большое число нарушений информационной безопасности совершается как раз лицами, потерявшими работу. Каждый уволенный по негативным обстоятельствам представляет собой угрозу

раскрытия конфиденциальной информации, к которой он имел допуск. Процедура увольнения должна включать немедленную блокировку всех учетных записей, смену паролей, блокировку удаленного доступа. Особенно серьезная угроза исходит от уволенного системного администратора, в таких случаях должна быть применена специальная процедура, включающая помимо обычных мер, полный аудит системы.

Должны быть также предусмотрены процедуры текущего контроля сотрудников. Контроль включает процедуры отчетности в соответствии с административной субординацией, оценку работы (ежегодная аттестация сотрудников), продвижение по служебной лестнице. Рабочая, доброжелательная атмосфера на предприятии является важным фактором производительности и безопасности. Однако необходимо соблюдать разумный компромисс между культивируемой в большинстве компаний атмосферой непринужденности и доверия между сотрудниками, с одной стороны, и необходимой для обеспечения безопасности атмосферой взаимного контроля и подозрительности — с другой.

При управлении персоналом следует придерживаться нескольких общепризнанных принципов.

Принцип разграничения обязанностей преследует несколько целей: во-первых, он способствует повышению производительности за счет специализации, во-вторых, устраняет ненужное дублирование, а в-третьих (что важно для безопасности), не дает концентрировать слишком много полномочий в одних руках. В особо критичных случаях такое разграничение вводится для того, чтобы некоторое действие могло быть произведено только с участием двух (или более) человек. Пример такой процедуры — доступ к банковской ячейке, которая от

крывается двумя ключами — ключом владельца содержимого ячейки и ключом представителя банка.

Правило обязательного отпуска — это правило, помимо заботы о здоровье сотрудника, дает возможность в его отсутствие основательно проверить, нет ли нарушений в его работе -одним из косвенных признаков этого может служить исчезновение какой-либо проблемы одновременно с его уходом в отпуск. Кроме того, появляется возможность узнать, не использует ли какой-либо злоумышленник его учетную запись. Доказательством этого служит то, что даже после ухода данного сотрудника в отпуск, в журнале регистрации продолжают появляться записи о связанных с его учетной записью событиях).

Принцип минимально необходимого уровня привилегий означает, что каждый сотрудник должен иметь только тот тип доступа и только к тем ресурсам, которые ему необходимы для выполнения его служебных обязанностей.

Нарушение этого принципа мы могли наблюдать в известном случае Брэдли Мэннинга, рядового американской армии, который находясь на удаленной военной базе в Ираке, сумел передать WikiLeaks огромный объем секретных документов. Для того чтобы добыть эту информацию, Мэннинг использовал удаленный доступ к сертифицированной сети министерства обороны США. И хотя Мэннинг и его сослуживцы имели допуск к секретным данным, те сведения, которые он смог получить по сети и впоследствии раскрыл (например, дипломатическую переписку), явно выходили за рамки того, что ему и его сослуживцам необходимо было знать.

Принцип непрерывного обучения правилам безопасности. Знакомство с политикой безопасности предприятия при поступлении на работу должно дополняться регулярными разъяснениями всех вносимых в нее изменений, необходимо донести сотрудникам важность обеспечения безопасности и объяснить, что в связи с этим ожидается от них. Кроме того, должны проводиться курсы и тренинги, повышающие квалификацию сотрудников в области безопасности. Конкретная направленность учебных курсов должна учитывать специфику профессиональных групп и должностных позиций. Особое внимание уделяется подготовке и переподготовке ИТ-персонала, которым должна предоставляться возможность изучения самых современных программных и аппаратных продуктов обеспечения безопасности. Хорошим стимулом может стать стремление получить один из общепризнанных сертификатов, подтверждающих профессиональную квалификацию специалиста в области информационной защиты.

Процедуры реагирования на нарушения безопасности

Процедура реагирования на инциденты в качестве начальных шагов должна рассматривать констатацию, распознавание и определение степени серьёзности нарушения.

При том, что конкретные действия существенно зависят от типа нарушения, во всех случаях требуется *немедленное оповещение ответственных лиц предприятия* (для этого в инструкции должны быть даны краткие и ясные указания о том, к кому и в каких случаях обращаться). В некоторых случаях должны быть также оповещены пользователи сети, другие сотрудники предприятия, другие организации, вовлечённые в данный инцидент, СМИ. На следующем шаге процедура может содержать

рекомендации по оценке масштабов и локализации нарушения. Например, для того чтобы ограничить распространения вируса, может быть предусмотрено отключение части сетевых сервисов, связанных с файловым обменом, сегментация сети, отключение сегментов от Интернета. Если риску подвергается критически важная информация, то она должна быть изолирована, вплоть до физического отключения серверов и накопителей. Процедура реакции также включает регистрацию инцидента, идентификацию преступника (когда это возможно), анализ происшествия, принятие мер, предупреждающих его повторение.

Для того чтобы персонал был готов выполнять действия, предписанные инструкцией на случай инцидента, время от времени должны проходить «боевые учения», чтобы люди могли на практике отработать свои задачи в различных ситуациях.

Следует отметить, что процедуры реакции на нарушения должны быть нацелены не только на то, чтобы по возможности снизить потенциальный ущерб предприятия от нарушения, но и на создание юридических оснований для преследования преступника. Доказательствами преступления могут оказаться «отпечатки» атаки в оперативной памяти, в дисковом накопителе, в журнале регистрации событий. Для наиболее популярных типов атак (DDoS-атак, фишинга, спама, вирусов и др.) последовательности действий по сохранению улик должны быть описаны заранее.

При выявлении физического вторжения должна быть вызвана охрана, которая должна иметь свои инструкции для широкого спектра потенциально опасных действий.

Правильная реакция на нарушение создаёт условия для эффективности последующих действий по восстановлению системы.

Поддержка работоспособности предприятия

Поддержка работоспособности информационной системы предприятия заключается как в выполнении повседневных рутинных процедур, так и в разработке процедур на случай сбоев или аварий системы. В последнем случае различают превентивные меры, которые

направлены на смягчение возможных последствий аварии, процедуры немедленного реагирования, предназначаемые для оперативного восстановления системы, и планы восстановления, рассчитанные на достаточно длительный период нахождения информационной системы в нерабочем состоянии (поддержка непрерывности бизнеса).

К числу повседневных процедур, обеспечивающих нормальное функционирование системы в обычных условиях, когда никаких аварийных событий не происходит, относятся, например:

- *процедуры поддержки системного и прикладного ПО* (учет, тестирование, обновление, конфигурирование, отслеживание несанкционированной загрузки и запуска программ);
- *процедуры поддержки оборудования* (учет, профилактические и ремонтные работы);
- *процедуры поддержки пользователей* (консультирование, ознакомление с новыми сервисами, новыми угрозами);
- *процедуры поддержки документации* (учет, хранение, создание резервных копий наиболее важных документов, таких, например, как инструкции по действиям в непредвиденных ситуациях). Доступ к некоторым документам, таким, например, как отчет об исследовании уязвимостей системы, информация о клиентах, финансовые документы и т. п., может быть ограниченным. Для работы с классифицированными документами должны быть определены особые процедуры, регламентирующие регистрацию, хранение, определение и изменение уровня секретности, порядок оформления допуска;
- *процедуры безопасной утилизации носителей с* конфиденциальными данными. Они являются критически важными для обеспечения безопасности. Бумажные носители уничтожаются путем шрединга, размачивания, сжигания. Чувствительная информация на жестких магнитных дисках должно быть стерта с помощью специального ПО, так как обычная операция delete уничтожает файл только логически, помечая его уничтоженным, но физически содержимое файла остается на диске и может быть легко прочитано. Самым надежным в таких случаях является физическое разрушение диска.

«Хочешь мира — готовься к войне», «предупрежден — значит вооружен», «знал бы, где упадешь, соломку подстелил» — все эти поговорки говорят о важности *превентивных* мер. К сожалению, когда дело касается информационных систем, основная часть работы по подготовке к авариям сводится к созданию процедур, рассчитанных на то, что авария *уже произошла* и надо восстанавливать систему. В число работ, которые *упреждают* аварию или *смягчают* ее последствия, относятся дублирование жизненно важных ресурсов на случай непредвиденного их выхода из строя, а также регулярное резервное копирование критически важных данных для последующего восстановления.

Избыточные, резервные устройства используются для некоторых серверов, маршрутизаторов и коммутаторов, телекоммуникационных каналов и

оборудования доступа к Интернету (как минимум, к двум разным провайдерам), а также источников бесперебойного питания и электрогенераторов.

В зависимости от схемы подключения резервного оборудования в случае аварии система либо может немедленно переключиться на дублирующее устройство, либо процесс переключения займет какое- то время, но в любом случае негативный эффект от аварии будет снижен.

В некоторых случаях необходимо готовить «дублеров» и для сотрудников, занимающих ключевые позиции в обеспечении работоспособности предприятия.

В качестве примера приведем структуру типовой инструкции по проведению *резервного копирования данных.* Инструкция включает следующие разделы:

- перечень лиц, ответственных за работу на разных этапах резервного копирования;
- состав и объем информации, которая подлежит резервному копированию (указываются серверы и каталоги на них);
- тип копирования (полностью или только изменения), определение наборов резервных копий (месячная, недельная, ежедневная копии), периодичность проведения резервного копирования; максимальный срок хранения резервных копий, состав и формат информации на этикетках для носителей;
- инструкции по работе с программными системами, поддерживающими процедуру резервного копирования;
- инструкция о действиях, которые должны быть предприняты в случае выявленных попыток несанкционированного доступа к резервируемой информации, а также иных нарушениях информационной безопасности произошедших в процессе резервного копирования;
- порядок контроля резервного копирования (для проверки качества резервных копий на ленте нужно регулярно восстанавливать скопированную информацию на другом компьютере): периодичность, учет ошибок копирования;
- порядок ротации носителей, которая заключается в загрузке, выгрузке носителей из системы резервного копирования, перемещении их в места хранения, уничтожении неиспользуемой больше

конфиденциальной информации с носителей, использовании новых носителей, а также в обеспечении хранения резервной копии вне офиса на случай катастрофы;

- инструкции по полному или частичному восстановлению информации из резервных копий с помощью программных систем резервного копирования ПО.

Если все же произошла авария системы, то вступают в действие *реактивные* процедуры восстановления. Диапазон планируемых действий простирается от простого перезапуска компьютерной сети, который потребовался из-за часового отключения электропитания, до переноса бизнеса на резервные площадки с полным переключением всей инфраструктуры.

При разработке процедур восстановления должно учитываться множество факторов, отражающих специфику бизнеса, информационной инфраструктуры, характер аварии и др. Все эта совокупность факторов определяет содержание восстановительного процесса. Важным параметром, который существенно влияет на существо и последовательность этапов восстановления, является время, которое дается специалистам для проведения восстановительных работ. Оценку этого времени можно получить, основываясь на результатах анализа рисков, проведенного для данной системы. Для каждого критичного сервиса и ресурса строится зависимость ущерба от времени простоя системы. Исходя из приемлемого ущерба могут быть определены оценки приемлемой длительности периода восстановления для каждого критичного ресурса. Эти времена должны быть взяты в качестве ориентира для разработчиков процедуры восстановления.

Для случаев длительных простоев разрабатываются планы и процедуры масштабных действий, предусматривающие временный перевод отдельных подразделений, а иногда и всего предприятия в другое место. Наиболее часто используется один из следующих подходов к резервированию инфраструктуры:

- полное зеркальное отражение инфраструктуры;
- воссоздание рабочей среды для небольшой, критически важной части системы, потерпевшей аварию (в том числе с возможностью базирования на автомобилях);
- распределенная инфраструктура, обладающая свойством живучести, когда при аварии система продолжает функционировать, хотя и в деградированном состоянии. В таких случаях при аварии отдельные компоненты системы берут на себя функции других частей. Предприятие может создавать распределенную структуру не только на базе собственных ресурсов, но и привлекать ресурсы сторонних организаций.

В процедурах, которые предполагается использовать во время аварийной остановки системы, надо учитывать то обстоятельство, что в таких ситуациях резко снижается потенциал штатных средств безопасности, а значит, надо предусмотреть временные меры защиты от несанкционированного доступа к наиболее ценным ресурсам системы (документам, серверам).

Физическая защита

В этой книге в основном будут обсуждаться угрозы ***техногенной*** природы, когда неправильная работа системы вызывается техническими причинами. Сейчас же коротко остановимся на угрозах физической характера и средствах процедурного уровня, используемых для защиты от физических угроз.

Физические угрозы — это природные катаклизмы и катастрофы, наводнения, землетрясения, пожары, ураганы, слишком высокая или слишком низкая температура, повышенная влажность, радиоактивность, электромагнитное излучение и прочее. В результате осуществления такого рода угроз страдает физическое окружение ИТ-системы: помещения, энергопитание, кондиционирование, телекоммуникации. Физические повреждения наносятся также всей информационной инфраструктуре, в том числе компьютерам, маршрутизаторам, кабелям, физическим носителям информации.

Помимо природных явлений, физические угрозы могут быть связаны с деятельностью человека: умышленной — проникновение в серверную комнату и вмешательство в его работу, или неумышленной — случайный разрыв соединительного кабеля.

Последствия физических повреждений могут сказаться на работе ИС даже более серьезно, чем те, которые вызываются техническим вмешательством в работу системы. Все мы знаем о трагических последствиях урагана Катрина, опустошившим Новый Орлеан и Луизиану, о цунами, разрушившим АЭС в Фукусиме, о страшной катастрофе башен Международного торгового центра в Нью Йорке — во всех этих случаях вместе с огромными людскими потерями было физически уничтожено и повреждено большое количество зданий, напичканных компьютерными системами, каждая из которых очевидно имела надёжные файерволы и системы обнаружения вторжений. И хотя в приведённых здесь случаях вряд ли можно было найти шанс для спасения, существует немало примеров, когда вычислительные центры успешно противостояли разрушительным силам природы.

Угрозы физической природы требуют особых средств защиты, среди которых есть как технические средства (ограждения, освещение, коробы для укрытия кабелей, замки, камеры слежения, огнетушители и др.), так и процедурные решения (пропускной режим на территорию предприятия, охрана границ территории и др.).

Вот, например, какие рекомендации по физической защите серверов дает компания Microsoft:

«Каждый сервер необходимо поместить в помещение в закрывающимися на ключ дверями. Доступ в такое помещение должен предоставляться только специально уполномоченным на это лицам. Перед тем как прекратить работу за системной консолью сервера и покинуть помещение, сотрудник должен выполнить блокировку или процедуру логического выхода, т. е. сделать так, чтобы для доступа к серверу потребовалось вводить пароль. Серверная комната должна быть устроена таким образом, чтобы никто, находящийся наружи, не мог видеть клавиатуру (и не смог подглядеть набираемые на клавиатуре пароли).»

К физическим процедурным средствам защиты относятся также многие работы, связанные с восстановлением системы после аварии, такие, например, как перемещение различных ресурсов системы в безопасное место, тушение пожара и др.

Первым очевидным шагом по физической защите предприятия является охрана границ его территории (охрана периметра) и внутренних построек и помещений. Для этого заблаговременно устанавливаются соответствующие технические охранные средства (ограждения, датчики, видеокамеры, замки, блокирующие устройства, и пр.).

Традиционно самым эффективным средством охраны всегда считались профессиональные охранники. Да и теперь именно человек является центральным элементом всей системы обеспечения физической безопасности, поскольку наблюдение основано на чувственном восприятии и современные охранные средства и системы требуют активного участия человека.

В число обязанностей охранника может входить:

- реакция на инциденты;
- проверка пропусков людей разных категорий (собственных сотрудников, сотрудников предприятий-партнеров и субподрядчиков, клиентов, работников обслуживающих и контролирующих организаций, гостей и др.);
- работа с системами мониторинга, проверка безопасности окон, дверей, хранилищ;
- контроль противопожарных датчиков;
- участие в процедурах, препятствующих кражам (проверка на выходе).

Работник охраны в своей работе должен следовать подробным инструкциям, определяющими зону его ответственности, контактную информацию для оповещения ответственного лица о возникновении непредвиденной ситуации, правила действий в разных ситуациях. Работники охраны должны проходить тренинги по пользованию техническими средствами защиты, даже такими простыми, как ключи. Например, должны быть чётко и ясно определены операции учета, хранения, выдачи ключей, а также процедуры, выполняемые в случае потери ключа и в случае списания ненужного ключа. Правила использования кодовых замков должны определять регулярность смены кодовых комбинаций, оповещения о новом коде и требования конфиденциальности.

Принципы защиты информационной системы

Ниже будут рассмотрены принципы построения системы обеспечения информационной безопасности, многие из которых имеют универсальный характер и применимы не только в информационной, но и в других самых разных сферах защиты.

Подход сверху вниз

Проектирование системы защиты должно идти сверху вниз

Подход «сверху-вниз», где понятие «верх» означает руководство предприятия, а «низ» — уровень рядовых сотрудников, соответствует универсальному принципу движения от общего к частному. При таком подходе все принципиальные решения принимаются топ- менеджментом, затем руководители промежуточных уровней преобразуют их в более развернутые планы и частные решения, которые, наконец, доводятся в виде инструкций и в разной степени формализованных процедур до уровня исполнителей.

Именно руководители предприятия определяют стратегически важные объекты защиты, они оценивают риски, которые может понести предприятие в результате разрушения тех или иных информационных активов и намечают стратегию защиты информационных ресурсов.

Такой подход является эффективным, так как, во-первых, руководители хорошо знают бизнес предприятия и могут правильно оценить риски, а следовательно, определить, какие именно информационные ресурсы нужно защищать особенно тщательно, а какие могут быть оставлены «без присмотра». Во-вторых, эти люди непосредственно несут ответственность за то, насколько эффективной окажется проектируемая система, а также обладают полномочиями для принятия критически важных решений.

Противоположный подход— «от частного к общему», или «снизу вверх», успешно используемый в некоторых сферах деятельности (например, в научных исследованиях), совершенно не применим для проектирования сложных технических систем, к которым относится система обеспечения безопасности. Решения, принимаемые

на уровне специалистов отдельных подразделений, могут оказаться несогласованными и не способствовать глобальной цели. Например, системный администратор на свой страх и риск, приложив большие усилия и потратив значительные средства, обеспечил надежную защиту базы данных, а в ней, как в последствии выяснилось, хранилась легко восстанавливаемая информация, которая не представляла для бизнеса особой ценности. Или другой пример. Руководитель отдела сетевой безопасности предприятия-провайдера Интернета для защиты транспортной подсистемы от внешних атак приобрел на основании собственного решения некое средство анализа сетевого трафика, способное собирать метаданные8 о трафике с помощью установленных в сети маршрутизаторов, в том числе IP-адреса отправителей и получателей каждой сессии некоторого клиента с некоторым сервером. Однако после того как система была внедрена, оказалось, что метаданные, собираемые данным сетевым анализатором, не обеспечивают анонимности, так как позволяют сопоставить имена пользователей и их IP-адреса, что противоречит федеральному закону «Об информации, информационных технологиях и о защите информации», который гласит о том, что в области защиты информации должны соблюдаться «неприкосновенность частной жизни, недопустимость сбора, хранения, использования и распространения информации о частной жизни лица без его согласия». Если бы решение о выборе сетевого анализатора принималось в рамках нисходящего процесса проектирования, то законодательное ограничение о персональных данных, зафиксированное на самом верхнем уровне в политике безопасности, последовательно передаваясь вниз по »лестнице» принятия решений, почти наверняка достигло бы уровня отдела сетевой безопасности и было бы учтено при выборе продукта.

Защита как процесс

Защита должна представлять собой непрерывный, циклический, проактивный процесс

Задача информационной защиты не может быть решена раз и навсегда, напротив, работа по защите ИС должна идти ***непрерывно*** на протяжении всего существования защищаемой системы. Защита должна адаптироваться к изменениям защищаемой системы и среды, в которой та существует.

Все системы безопасности уникальны, поскольку отражают специфику конкретных предприятий, для защиты которых они предназначены. Но какие бы различные цели не преследовались при их создании, какие бы различные технологии не использовались, какие бы индивидуальные решения не принимались, общий ход проектирования для всех правильно построенных систем защиты должен иметь ***циклический*** характер, поскольку основой процесса обеспечения безопасности является следующая повторяющаяся

8 Метаданные — это служебная информация, используемая при проведении сеанса связи. Например, для электронной почты метаданными являются e-mail-адреса и IP-адреса отправителя и получателя, название темы письма, дата, время отправления и получения и др. То есть все, кроме содержания письма.

последовательность:

1) анализ состояния защищаемой системы, ее уязвимостей и угроз;
2) оценка рисков и управление рисками;
3) разработка политик всех уровней;
4) реализация принятых решений, направленных на снижение рисков;
5) возвращение к пункту 1.

Процесс обеспечения безопасности по возможности должен иметь ***проактивный*** (т. е. упреждающий), а не реактивный характер. При реактивном подходе защита заключается в принятии мер уже после того, когда нарушение безопасности произошло. Очевидно, что для успешности отражения атаки в первую очередь важны правильно выбранные действия и скорость их выполнения, а как раз этого трудно ожидать в ситуации кризиса. Поэтому более предпочтительным является проактивный подход, когда для защиты от вероятных угроз в спокойной обстановке проводится основательная подготовка оборонительных мер: устанавливаются необходимые технические средства, продумываются действия персонала, составляются и документируются инструкции — т. е. делается все, что только может быть сделано заранее.

Эшелонированная защита

Эффективная защита обеспечивается многократным резервированием средств безопасности

Надежность решения любой задачи повышается, если использовать резервирование. Задача обеспечения безопасности не является здесь исключением. Так, например, для обеспечения физической сохранности важного документа могут быть использованы самые разные средства защиты: дверные замки, датчики разбития окон, противопожарные сигнальные устройства, тревожная кнопка, сейф и масса других полезных приспособлений.

Рис. 3.5. Рубежи обороны ИТ-системы

Информационная система существует в окружении гораздо более изощрённых и многообразных угроз, здесь тем более невозможно найти панацею — одно-единственное средство, которое могло бы со 100 %-ной надежностью противостоять всем видам атак. Поэтому на пути к защищаемому информационному ресурсу, как правило, устанавливают несколько барьеров. Вместе с тем, возникает резонный вопрос: если ни одно из средств безопасности не является абсолютно надежным и в принципе может быть преодолено злоумышленником, то в чем смысл использования нескольких защитных рубежей? Ответ состоит в том, что многократное резервирование используется в системах защиты не столько для того, чтобы какое-то из защитных средств продублировало отказавшее, но главным образом для того, чтобы заставить преступника потратить как можно больше времени на преодоление череды защитных барьеров. Замедление атаки увеличивает шансы ее обнаружения и принятия адекватных мер.

Рассмотрим, например, как реализуется принцип эшелонированной защиты в случае, когда необходимо обеспечить безопасность данных, хранящихся на одном из хостов внутренней локальной сети предприятия. На рис. 3.5 концентрическими окружностями представлены рубежи обороны, каждый из которых добавляет к уже накопленному защитному потенциалу собственные средства защиты (некоторые виды этих средств обеспечения безопасности будут рассмотрены в последующих главах).

Самый внешний слой (организационно-административный) решает задачу безопасности данных, затрудняя злоумышленникам физический доступ к данным, с этой целью разрабатываются и применяются административные и организационные меры безопасности, такие как проверка персонала при приеме на работу, взаимный контроль персонала, ограничение использования переносных портативных носителей и др.

Следующий слой также направлен на защиту от физического проникновения, но другими средствами — средствами физической защиты: ограждения, освещение, видеокамеры, контроль входа в здание, двери с кодовыми замками и т. п.

Далее вступают в действие технические средства безопасности, которые для сети с типовой структурой включают следующие рубежи защиты:

- внешняя сеть — для защиты от проникновения применяет средства регистрации входа, аудит, защитные свойства VPN;
- периметр внутренней сети — усиливает защиту путем использования файервола и прокси-серверов;
- внутренняя сеть — здесь добавляются системы обнаружения и предотвращения вторжений сетевого уровня;.
- хост — дополнительно проводятся процедуры аутентификации и авторизации, работает программный файервол, антивирус, системы обнаружения и предотвращения вторжений уровня хоста;
- данные — механизм контроля доступа и шифрование.

Сбалансированная защита

Степень защищенности системы измеряется защищенностью ее самого слабого звена

Этот принцип можно сформулировать и несколько по-другому: при построении системы безопасности необходимо обеспечивать баланс стойкости всех ее компонентов. Например, если в сети все сообщения шифруются, но ключи легкодоступны, то эффект от шифрования нулевой.

Из данного принципа можно сделать также следующее заключение: если у злоумышленника существует несколько путей нанести урон системе и один из этих путей имеет слабую защиту, то нет смысла добиваться высокого качества

защиты других путей. То есть, если внешний трафик сети, подключенной к Интернету, проходит через мощный сетевой экран, но пользователи имеют возможность связываться с узлами Интернета по коммутируемым линиям через локально установленные модемы, то деньги (как правило, немалые), потраченные на сетевой экран, можно считать выброшенными на ветер.

В таких случаях оказывается полезным еще один принцип — ***принцип единого контрольно-пропускного пункта,*** который заключается **в** том, что весь входящий во внутреннюю сеть и выходящий во внешнюю сеть трафик проходит через единственный узел сети, например через межсетевой экран. Только это позволяет в достаточной степени контролировать трафик. В противном случае, когда в сети имеется множество пользовательских станций, имеющих независимый выход во внешнюю сеть, очень трудно скоординировать правила, ограничивающие права пользователей внутренней сети на доступ к серверам внешней сети и обратно — права внешних клиентов на доступ к ресурсам внутренней сети.

Необходимость баланса стойкости разных компонентов системы безопасности особенно ярко иллюстрируется провалами силовых ведомств, когда мощные организации, располагающие гигантскими ресурсами защиты, допускают существование явных прорех в своих системах обеспечения безопасности. Так известен случай, когда дисковый накопитель с классифицированными данными вооруженных сил США был случайно обнаружен продающимся на базаре в Ираке. Причина — использование ненадежных процедур утилизации данных и аппаратуры, при том что для остальных стадий существования данных — хранения и передачи — использовались мощные алгоритмы шифрации. Еще два примера связаны с масштабными утечками сверхсекретных данных — Мэннинг (2010 г.) и Сноуден (2013 г.). В обоих случаях очевидным слабым звеном стало управление персоналом: несмотря на то что незадолго до инцидентов в поведении потенциальных нарушителей их коллегами отмечались «странности», а также то, что в карьере каждого из них произошли события, которые обычно квалифицируются как провоцирующие факторы, в отношении них не было предпринято никаких расследований. Кроме того, в обоих случаях не сработал контроль использования портативных запоминающих устройств. В результате за пределы зоны безопасности были вынесены сотни тысяч секретных файлов, записанные Мэннингом на CD-дисках и Сноуденом на устройствах memory stick.

Принцип баланса означает не только «подтягивание» наименее слабых компонентов защиты, но и устранение неадекватно сильной защиты в тех случаях, где она не требуется или ее невозможно реализовать. Некоторых наших соотечественников иногда удивляют стеклянные входные двери в частных домах европейских граждан, хотя это решение вполне логично: какой смысл укреплять одно звено (ставить прочную дверь), если рядом другие звенья (застекленные окна первого этажа) все равно останутся слабыми. Следующий пример совсем из другой области. Известно9, что допуск к сверхсекретным данным (top-secret) США имеют 1,4 миллиона людей. Однако вряд ли можно считать сколько-нибудь секретными данные, авторизованными пользователями которых являются полтора миллиона человек. В такой ситуации, возможно, имело бы смысл привести формальные ограничения доступа в соответствие фактическому положению вещей, т. е. ослабить защиту значительной части документов.

Компромиссы системы безопасности

Система обеспечения безопасности создается в результате компромисса между качеством защиты, с одной стороны, и затратами и,! разработку этой системы — с другой. Под качеством здесь понимается комплекс характеристик

9 http://www.usatoday.com/story/news/p...table/2428809/

Здесь идет речь только об остановке в пределах данного цикла создания системы безопасности, на новом витке развития системы могут быть приняты другие решения.

— функциональное разнообразие, надежность защиты, удобство работы сотрудников, поддерживающих ' игтему безопасности, и сотрудников других подразделений предпри- иия. Поясним природу необходимых компромиссов, используя очень упрощенную модель, описывающую финансовую сторону внедрения системы обеспечения безопасности.

Цель системы обеспечения безопасности — снизить прогнозиру- пмый совокупный ущерб, который мог бы быть нанесен предприятию, если бы система защиты отсутствовала. Пусть в исходном состоянии, до внедрения системы защиты, ущерб предприятия от атак (риск) оценивался как ***L-before.***

При внедрении системы защиты возможный ущерб от атак сни- 1ился и оценивается теперь как Laffer, однако в позиции «убытки» у предприятия добавились затраты ***N*** на внедрение системы безопасности. Кроме того, к убыткам предприятия должны быть отнесены те потери ***U,*** которые предприятие понесло из-за снижения производительности в результате внедрения системы безопасности. (Такое снижение может быть вызвано как дополнительными затратами вычислительных ресурсов, так и необходимостью выполнения сотрудниками предприятия дополнительных, связанных с безопасностью процедур.)

Очевидно, что решение о внедрении системы безопасности можно считать экономически обоснованным только в том случае, если риск Lafter в совокупности с затратами на систему безопасности ***N*** и потерями из-за снижения производительности ***U*** окажутся меньше исходного значения риска ***L-before:***

$$N + U + L\text{-}after < L.before. \quad (1)$$

Из этого соотношения следует, что надо стремиться уменьшать каждое из слагаемых в левой части неравенства. Проблема, однако, состоит в том, что они не являются независимыми и уменьшение одного из них может вызвать увеличение других. Так, например, затраты на систему безопасности положительно влияют на защищенность предприятия, т. е., чем больше ***N,*** тем меньше ***L.after*** (что, к сожалению, не всегда справедливо в реальных разработках). Значит, при создании системы обеспечения безопасности необходимо стремиться к некоторому компромиссному варианту, который минимизирует выражение в левой части неравенства.

При создании системы безопасности необходим компромисс между затратами и рисками

В соответствии с нашей идеальной моделью слагаемые ***L^after*** и ***N*** действуют разнонаправленно, а значит, вложение денег в защиту выгодно, только если ущерб ***L_after*** снижается быстрее, чем растут затраты ***N.*** Реальность, однако, намного сложнее, и рекомендация ориентироваться на соотношение скоростей является сугубо абстрактной. В то же время в практической деятельности можно использовать тот факт, что возможный ущерб ***L-after*** никогда не снижается ниже некоторого порога.

Действительно, создание абсолютно непроницаемой защиты невозможно, так как у атакующих всегда остается теоретическая возможность взломать любую защиту, это вопрос только времени и тех средств, которыми располагают злоумышленники, а значит, рано или поздно наступает такой момент, когда становится бессмысленным продолжать вкладывать деньги в систему безопасности. Остается вопрос: на каком уровне затрат надо остановиться*? В первом грубом приближении ответ следует из неравенства (1), которое выражает условие экономической обоснованности затрат на систему защиты. Поскольку слагаемые ***U*** и ***L^after*** — положительные числа, то необходимым, но не достаточным условием справедливости неравенства ***N + U + Ljafter < LJaefore*** является ***N < LJbefore.***

Этот результат часто формулируют в форме ***принципа разумной достаточности:***

- Затраты на обеспечение безопасности информации должны быть по крайней мере не больше, чем величина потенциального ущерба от ее утраты.

Вопрос «Когда следует остановиться?», существует и в другой постановке: какой уровень защищенности системы является достаточным? Для ответа на него разработчикам системы безопасности предлагается встать на место злоумышленника и попытаться оценить, какой уровень защиты злоумышленник мог бы посчитать неприемлемым для себя. Так, например, вряд ли имеет смысл браться за добычу конфиденциальных данных, если эта работа настолько длительная, что к тому времени, когда секретная информация попадет в руки, она уже устареет и не будет представлять никакой ценности. Аналогично, никто (из экономически мотивируемых преступников) не будет заниматься взломом системы, если выгоды от обладания защищаемым

Рис. 3.6. Компромиссы системы безопасности

ресурсом меньше, чем средства, потраченные на проведение атаки. Исходя из этих соображений, формулируются еще два варианта принципа разумной достаточности, относящихся к уровню защищенности, обеспечиваемому системой безопасности (стойкости):

- стойкость системы обеспечения безопасности считается достаточной, если время преодоления защиты превосходит время старения информации;
- стойкость системы обеспечения безопасности считается достаточной, если стоимость ее преодоления злоумышленниками превосходит стоимость полученной ими выгоды.

Таким образом, проектирование системы безопасности требует нахождения множества компромиссов (рис. 3.6) между возможными затратами и возможными рисками. Определяя политику безопасности, администратор должен взвесить ущерб, который может понести предприятие в результате нарушения защиты данных, и соотнести его с затратами, требуемыми на обеспечение безопасности этих данных. Так, в некоторых случаях можно отказаться от дорогостоящего межсетевого экрана в пользу стандартных средств фильтрации обычного маршрутизатора, в других же приходится идти на беспрецедентные затраты. Главное, чтобы принятое решение было обосновано экономически.

При создании системы безопасности необходим компромисс между ее эффективностью и эффективностью защищаемого бизнеса.

При внедрении любой системы защиты производительность предприятия может только уменьшаться, так как, во-первых, дополнительные задачи СОИБ обременяют ИТ-инфраструктуру: часть вычисли

тельных ресурсов — процессорное время, память, пропускная способность линий связи, затраты на обслуживание и администрирование и др. — идет на решение задач безопасности; во-вторых, сотрудникам предприятия в связи с внедрением (или усовершенствованием) системы обеспечения безопасности приходится выполнять дополнительные требования, например требование блокирования компьютера при каждой отлучке от рабочего места, дополнительный контроль на входе путем анализа сетчатки глаза, получение разрешения на использование ресурсов Интернета и т. п. Такая дополнительная работа доставляет сотрудникам неудобства и приводит в конечном счете к снижению производительности компании.

Следует заметить, что существуют примеры, когда внедрение системы безопасности благоприятно сказывалось на удобстве работы сотрудников и производительности основного бизнеса. Это никак не противоречит всему выше сказанному: так случается, когда в рамках системы безопасности внедряется технология многоцелевого назначения. Например, технология виртуальных частных сетей на основе MPLS предоставляет помимо защищенности ряд других преимуществ, в том числе возможность организации удобного и производительного удаленного доступа к серверам корпоративной сети.

В большинстве случаев процедуры безопасности только затрудняют работу. Поэтому так важно проводить разъяснительную работу, убеждать работников предприятия в необходимости следования правилам и процедурам безопасности. В некоторых случаях могут быть применены директивные и дисциплинарные меры.

Однако все это справедливо только до тех пор, пока привнесённые системой безопасности неудобства не начнут оказывать очевидного негативного влияния на бизнес предприятия. Пусть, например, одной из мер, предлагаемых программой безопасности, является уменьшение числа сотрудников, работающих из дома, однако для этого предприятие должно пойти на существенные затраты, чтобы создать дополнительные рабочие места в офисе. Или предлагаемое средство защиты настолько сложное, что, для того чтобы его использовать, сотрудники предприятия должны пройти дорогостоящее и длительное обучение, возможно с отрывом от работы. В такой ситуации необходимо провести анализ, на основании которого должен быть найден компромисс между качеством системы безопасности и ее влиянием на удобство работы персонала и производительность бизнеса.

Вопросы к главе 3

1. К числу субъектов правоотношений в области ИБ можно отнести:
а) провайдеров телекоммуникационных сетей;
б) информационные системы (государственные и частные);
в) услуги, оказываемые в области И Б;
г) информация, составляющая государственную тайну;
д) разработчики программных и аппаратных средств;
е) персональные данные.
2. К числу объектов правоотношений в области ИБ можно отнести:
а) алгоритмы шифрования;

б) аппаратные ключи;
в) обладателей информации;
г) потребителей информации;
д) операторов информационных систем;
е) владельцев интернетовских сайтов.
3. Что из перечисленного при определенных условиях может быть квалифицировано как нарушение законодательства в области информационной безопасности:
а) организация и проведение атаки DDoS;
б) отсутствие лицензии на разработку средств шифрования;
в) использование несертифицированных программных продуктов;
г) нарушение правил пажарной безопасности в помещении информационного центра;
д) рассылка спама.
4. Соблюдение стандарта по умолчанию является:
а) добровольным;
б) обязательным.
5. Отметьте правильные на ваш взгляд продолжения тезиса «Политика информационной безопасности это:
а) режим работы предприятия»;
б) общее руководство, определяющее главные направления деятельности в области защиты ИС»;
в) основные концепции защиты ИС предприятия»;
г) закон, обязательный для исполнения всеми сотрудниками предприятия»;
д) совокупность документированных принципов, правил, процедур и подходов»;
е) подробная пошаговая инструкция, объясняющая последовательность действий персонала при возникновении угрозы».
6. К какому уровню политики безопасности предприятия может быть отнесен документ «Политика безопасности компьютерной инфраструктуры предприятия»:
а) верхнему;
б) среднему;
в) нижнему.
7. К процедурному уровню средств безопасности могут быть отнесены:
а) неформальные процедуры, приводимые в действие человеком и направленные на поддержку безопасности;
б) программные процедуры, обеспечивающие безопасность в программных комплексах;
в) автоматическая процедура подключения резервного источника питания;
г) инструкция оператора по проведению резервного копирования данных.
8. Какие цели преследует разграничение обязанностей персонала:
а) повышение производительности труда;
б) изоляция сотрудников друг от друга;
в) избежание конфликта интересов;
г) устранение ненужного дублирования;
д) избежание концентрации слишком больших полномочий в одних руках;

е) создание ситуации, когда некоторое действие может быть произведено только с участием нескольких сотрудников.

9. Какие нарушения принципов безопасности можно обнаружить в следующем утверждении: «На предприятии силами сотрудников отдела ИТ-поддержки была проведена процедура анализа рисков, полностью завершившаяся год назад»:

а) при создании системы безопасности необходим компромисс между затратами и рисками;

б) степень защищенности системы измеряется защищенностью ее самого слабого звена;

в) эффективная защита обеспечивается многократным резервированием средств безопасности;

г) защита должна представлять собой непрерывный, циклический, проактивный процесс;

д) проектирование системы защиты должно идти сверху вниз;

е) ни один принцип не нарушен.

10. Какие из следующих утверждений являются ошибочными:

а) нет смысла предусматривать подсистему аутентификации приложения, основанную на использовании многоразовых паролей, если многоразовые пароли уже используются и при логическом входе в систему;

б) надежная система авторизации может компенсировать недостаточную стойкость паролей пользователей, которые они использовали при входе;

в) стойкость системы защиты считается достаточной, если затраты на нее превысили запланированный уровень;

г) затраты на обеспечение безопасности информации должны быть по крайней мере не больше, чем величина потенциального ущерба от ее утраты;

д) иногда стойкость системы обеспечения безопасности считают достаточной, если стоимость ее преодоления злоумышленниками превосходит стоимость полученной ими выгоды.

Часть II

Базовые технологии компьютерной безопасности

Эта часть книги посвящена базовым технологиям безопасности компьютерных сетей, т. е. таким технологиям, которые могут использоваться самыми разными средствами обеспечения защиты. Такими основополагающими технологиями являются, в частности, аутентификация, авторизация, управление доступом, аудит, фильтрация и сегментация. Все эти технологии используются в различных компонентах компьютерной сети: в операционных системах и приложениях, в транспортных протоколах и в сетевых устройствах. И хотя в конкретных продуктах могут быть применены различающиеся реализации этих технологий, все они построены в соответствии с едиными принципами, используют схожие методы и приемы. Знание базовых технологий позволяет быстро «разобраться» с конкретными реализациями программных и аппаратных средств безопасности. Тут как нельзя лучше подходит известное выражение «знание нескольких принципов освобождает от знания многих фактов».

4 *Криптография*

Криптография имеет глубокие исторические корни, некоторые свидетельства говорят даже о том, что она возникла одновременно с появлением письменности, по меньшей мере 4 тысячи лет назад.

Испокон веков зашифрованные тексты использовались в дипломатической и частной переписке, в работе спецслужб, при проведении торговых операций, для описания секретных изобретений. Огромный вклад в развитие криптографии внесли военные коммуникации, первый математический труд в этой области, написанный в 1883 году Огюстом Керкгоффсом (Auguste Kerckhoffs) так и назывался «Военная криптография». Именно в этой работе был сформулированы несколько основополагающих принципов, в том числе знаменитое правило Керкхоффса о стойкости шифра. Новым вызовом криптографии стало появление Интернета и связанного с ним взрывного роста коммуникаций через общедоступные каналы связи. Объективная потребность в защите данных, передаваемых через публичную сеть, в принципе не могла быть удовлетворена имеющимися в то время средствами, построенными на основе симметричного шифрования; нужен был совершенно новый подход, и этот подход был найден — в середине 70-х годов прошлого столетия Диффи и Хеллман совершили революционный прорыв в области криптографии, предложив асимметричный метод шифрования.

Основные термины и понятия

Прежде чем перейти к конкретным методам и алгоритмам шифрования, давайте определим некоторые базовые понятия криптографии применительно к безопасности 1/1Т-систем.

Шифрование — это средство обеспечения конфиденциальности данных, хранящихся в памяти компьютера или передаваемых по проводной или беспроводной сети.

Шифрование является краеугольным камнем всех служб информационной безопасности, будь то система аутентификации или авторизации, защищенный канал или средства безопасного хранения данных.

хадЛххххгяяЧХ10 А язю

Рис. 4.1. Зашифрованное сообщение из рассказа Артура Конана Дойля «Пляшущие человечки»

Любая процедура шифрования, превращающая информацию из обычного «понятного» вида в «нечитабельный» зашифрованный, естественно должна быть дополнена процедурой *дешифрования,* которая, будучи примененной к зашифрованному тексту11, снова приводит его в понятный вид.

Пара процедур — шифрование и дешифрование — называется *криптосистемой* Обычно криптосистема предусматривает наличие специального элемента — *секретного ключа.*

В качестве ключа может выступать некоторый предмет, например книга, или число, или рисунок (рис. 4.1). Простейший метод шифрования — это замена букв в шифруемом тексте в соответствии с тем или иным правилом. Например, каждой букве алфавита может ставится в соответствие другая буква этого алфавита, сдвинутая на некоторое число позиций влево или вправо. В качестве секретного ключа здесь выступает ***число,*** определяющее сдвиг. Сдвиг может быть переменным, а правило преобразования задано полной таблицей соответствия букв, тогда ключом шифра является эта ***таблица.*** Очень широко распространен также «книжный» метод шифрования, когда шифрованный текст представляет собой тройки чисел {номер страницы, номер строки, номер буквы}, задающих позицию буквы в ***книге,*** которая в данном случае является секретным ключом.

Еще один древнейший метод шифрования основан на использовании предмета в качестве ключа, а именно скиталы — конуса с намотанной на него по спирали узкой лентой (рис. 4.2). Текст, который надо передать в нечитаемом виде, пишется на ленте вдоль конуса. В размотанном виде текст на ленте представляет собой бессмысленную последовательность букв. Чтобы прочитать его, получатель должен таким же образом намотать ленту на конус, имеющий в точности те же параметры. В данном случае ключом является ***скитала.***

Криптосистема считается ***раскрытой,*** если найдена процедура, позволяющая подобрать ключ за реальное время.

шифрования, будем условно называть «текстом», учитывая, что это может быть также числовой массив или графические данные.

11 Информацию, над которой выполняются функции шифрования и де

Методы раскрытия криптосистемы, процедуры выявления уязвимости криптографических алгоритмов, выяснение секретного ключа называют *криптоанализом* или взломом шифра. Попытку раскрытия

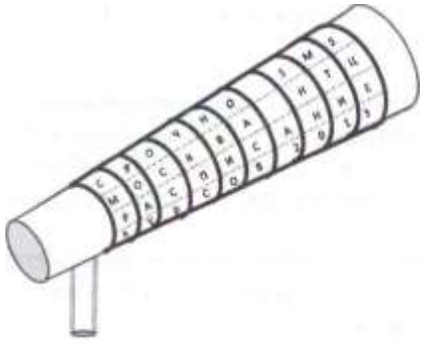


Рис. 4.2. Скитала

Рис. 4.3. Огюст Керкгоффс— нидерландский криптограф второй половины XIX века

конкретного шифра с применением методов криптоанализа называют *криптографической атакой.*

Одним из классических методов криптоанализа, применяемых для раскрытия шифров, основанных на перестановке или замене букв, является ***частотный анализ.*** Для текстов, написанных на определенном языке, относящихся к определенной сфере знаний, существует устойчивые статистические данные о частоте, с которой встречается в тексте та или иная буква или последовательность букв, включая некоторые слова. Обладая такими данными и проведя статистический анализ зашифрованного текста, можно выполнить обратную замену символов.

Сложность алгоритма раскрытия является одной из важных характеристик криптосистемы и называется *криптостойкостью.*

В криптографии принято *правило Керкгоффса,* заключающееся в том, что ***стойкость шифра должна определяться только секретностью ключа.*** Так, все стандартные алгоритмы шифрования (например, AES, DES, PGP) широко известны*, их детальное описание содержится в легкодоступных документах, но от этого их эффективность не снижается. Система остается защищенной, даже если злоумышленнику известно все об алгоритме шифрования, но он не знает секретный ключ.

Существуют два класса криптосистем — *симметричные* и *асимметричные.* В симметричных схемах шифрования (классическая криптография) секретный ключ шифрования совпадает с секретным ключом дешифрования. В асимметричных схемах шифрования (криптография с открытым ключом) ключ шифрования не совпадает с ключом дешифрования.

* Вместе с тем существует немало фирменных алгоритмов, описание которых не публикуется для того, чтобы усилить защиту.

Симметричные алгоритмы шифрования

Теоретические основы *симметричной криптосистемы* впервые были изложены в 1949 г. в работе Клода Шеннона.

На рис. 4.4 приведена модель симметричной криптосистемы. В данной модели три участника: два абонента, желающих, обмениваться шифрованными сообщениями, и злоумышленник, который хочет перехватить и каким-либо образом расшифровать передаваемые сообщения.

При объяснении алгоритмов шифрования здесь и далее мы будем называть участников обмена Алисой и Бобом, а злоумышленника, старающегося перехватить их сообщения — Евой. Эти имена традиционно используются в криптографии для таких целей.

В распоряжении Алисы и Боба имеется незащищенный канал передачи сообщений, который в принципе может прослушиваться злоумышленником. Поэтому они договариваются использовать шифрование, и для этого им нужен секретный, известный только им двоим

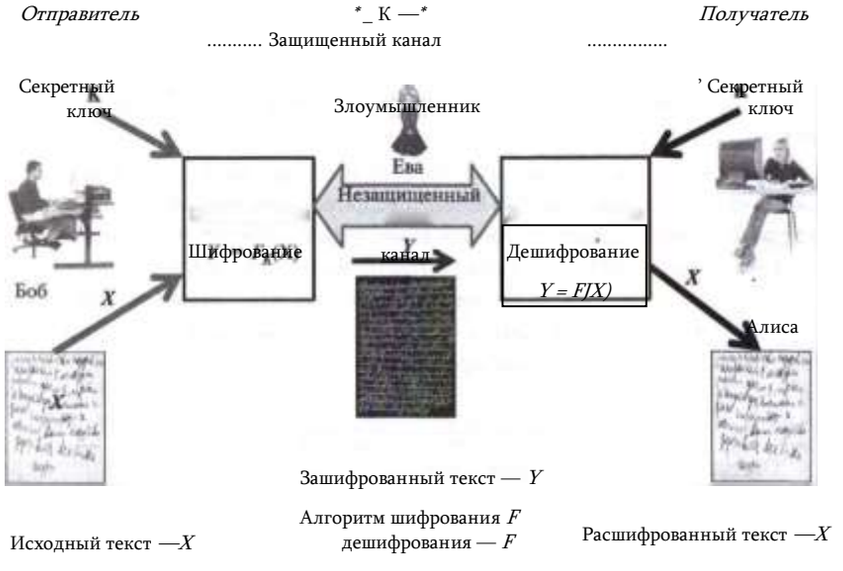


Рис. 4.4. Модель симметричного шифрования

ключ. Этот ключ им был передан (или один из них послал его другому) заранее ***по надежному каналу.*** Боб и Алиса находятся в абсолютно равном (симметричном) положении, каждый из них может как посылать шифрованные сообщения, так и получать и расшифровывать их. Для определенности на рис. 4.4 показана схема передачи сообщений со стороны Боба.

Боб зашифровывает свое сообщение — открытый текст X— функцией шифрования ***F*** с секретным ключом ***к*** и передает в открытый канал результат — шифрованный текст ***Y*** Алиса получает ***Y*** и передает его на вход функции дешифрования P, которая выполняет в обратном порядке все действия, которые выполнялись функцией ***F.*** Это может быть сделано, только, если на вход функции ***F*** будет подано то же самое значение параметра — значение ключа ***к.*** Алиса имеет секретный ключ и поэтому получает расшифрованное значение. При необходимости передавать шифрованные сообщения Бобу Алиса должна действовать анналогичным образом.

Алгоритм DES

До недавнего времени наиболее популярным стандартным симметричным алгоритмом шифрования данных был ***DES*** (Data Encryption Standard). Алгоритм разработан фирмой IBM и в 1976 году был рекомендован Национальным институтом стандартов и технологий США (NIST) к использованию в открытых секторах экономики. И хотя сейчас на смену DES пришел другой симметричный алгоритм AES, будет весьма поучительно рассмотреть некоторые приемы, использованные в алгоритме DES, поскольку они носят достаточно универсальный характер и используются в других симметричных алгоритмах.

Суть этого алгоритма заключается в следующем (рис. 4.5). Данные шифруются ***поблочно.*** Перед шифрованием любая форма представления данных преобразуется в числовую. Числа получают в результате применения любой открытой процедуры преобразования блока текста в число. Пусть требуется зашифровать текст RUSSIA2013. Исходный текст представим в числовой форме, например заменой каждой буквы ее шестнадцатиричным кодом ASCII:

R U S S I A 2 0
1 3

10100101010101 1010011 1010011 1001001 10000010110010 0110000 01100010110011

После слияния получаем следующую последовательность длиной 70 битов:

1010010101010110100111010011100100110000010110010011000001100010110011

Левая и правая части блока данных после перестановки

Левая L1 и правая R1 части блока данных после первого цикла преобразований

Рис. 4.5. Первый цикл преобразования Фейстеля

На вход шифрующей функции данные поступают блоками размером 64 бита, в этом примере мы ограничимся одним первым блоком, отбросив последние б битов:

1010010101010110100111010011100100110000010110010011000001100010

В алгоритме шифрования DES можно выделить несколько смысловых этапов.

1. Начальная перестановка

Перестановка битов в блоке данных выполняется в соответствии с правилом, заданным таблицей 4.1. То есть на место 1-го бита исходного текста ставится его 58-й бит, на место 2 бита — 50-й и так далее. (Подобные таблицы используются и на других этапах алгоритма DES, каждая из них содержит фиксированные для алгоритма DES значения. Эти таблицы являются неотъемлемыми компонентами алгоритма шифрования DES.)

В результате перестановки получаем первый блок в следующем виде:

1010001001111111000000111001011010000010111011001001011001000001O

Таблица 4.1

Начальная перестановочная таблица алгоритма DES

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

2. Циклические преобразования Фейстеля.

После выполнения перестановки блок делится на две равные части по 32 бита каждая. Обозначим левую часть L и правую часть R:

L= 10100010011111100000011100101101 R=

00000101110110010010110010000010

Алгоритм DES предполагает выполнение 16 циклов преобразований, называемых циклами Фейстеля, которые собственно и представляют собой процедуру шифрования. В каждом цикле выполняется преобразование данных, полученных в предыдущем цикле.

Рассмотрим первый цикл Фейстеля (рис. 4.5), для которого исходными данными являются левая L и правая R части блока данных после перестановки. Результатом первого цикла являются новые значения левой части L1 и правой части R1.

L1 получается простой заменой на R.

Получение правой части R1 оказывается сложнее. Для этого сначала вычисляется функция Фейстеля F, параметрами которой являются исходное значение правой части R и секретный ключ K.

3. Преобразование ключа

Исходное значение секретного ключа должно иметь 64 бита. Затем по определенной схеме из битовой последовательности убирается 8 битов, так чтобы каждый байт ключа содержал нечетное число единиц. Эти биты используются для контроля целостности 56-битового ключа при хранении и передаче. Затем для оставшихся 56 битов ключа делаются перестановки, которые задаются таблицей, подобной табл. 4.1, но имеющей другие значения в ячейках. Далее выполняется циклический сдвиг всей последовательности влево на 1 позицию. (Сдвиг на 1 позицию выполняется также в циклах Фейстеля с номерами 2, 9 и 16; для остальных циклов последовательность сдвигается на 2 позиции.) Из полученной в результате сдвига 56-битовой последовательности убираются 8 битов, позиции которых указаны в еще одной таблице. В результате получается 48-битовый ключ K.

4. Вычисление функции ***F***

Функция ***F*** вычисляется путем выполнения следующих действий:

- расширение правой части ***R,*** состоящей из 32 битов, до 48. Это делается дублированием некоторых битов, позиции которых определены в специальной таблице алгоритма DES;
- сложение по модулю 2 (операция XOR) ключа K и правой части R;
- преобразование 48-битовой суммы в 6-битовые блоки и замена их на 4-битовые по определенной в алгоритме схеме. В результате получается 32-битовая последовательность;
- перестановка битов в полученной последовательности.

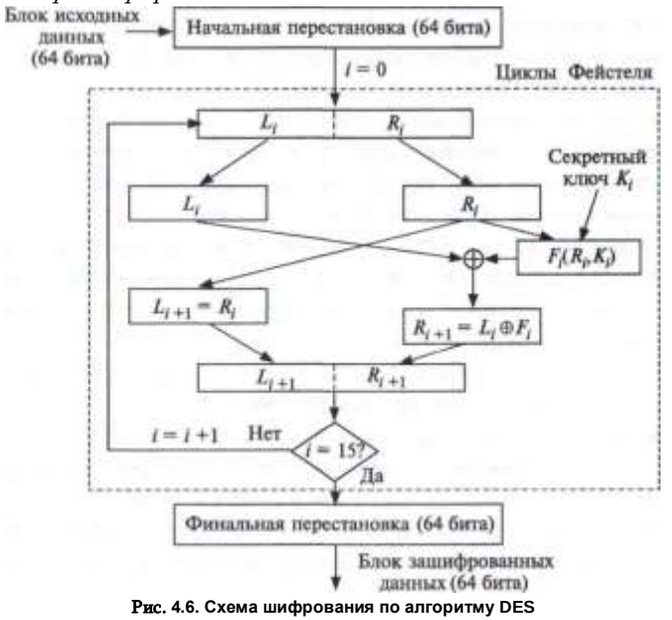


Рис. 4.6. Схема шифрования по алгоритму DES

Затем вычисляется новое значение правой части R1 как сумма по модулю 2 значений функции ***F*** и левой части L.

На этом заканчивается первый цикл и происходит переход к следующему, второму циклу (рис. 4.6).

6. Финальная перестановка

После того как будут выполнены все 16 циклов, делается финальная перестановка 64-битовой последовательности {L15, R15}. Эта перестановка является обратной по отношению к начальной, задаваемой таблицей 2.2. Так, например, на место 1-го бита последовательности {L15, R15} ставится ее 40-й бит, на место 2-го бита — 8-й бит и т. д.

Результат перестановки является зашифрованным значением исходного блока данных. Полный шифрованный текст получается слиянием результатов шифрования для всех блоков исходного текста.

Процедура дешифрования выполняется как обратный процесс: выполняется перестановка, обратная финальной, откручиваются назад 16 циклов Фейстеля, в каждом из которых активно используется секретный ключ, и в завершение выполняется перестановка, обратная начальной.

Поскольку собственно алгоритм DES не является секретом и широко доступен, в том числе доступны все таблицы, описывающие перестановки, то стойкость алгоритма (степень сложности дешифрования), определяется только сложностью подбора ключа. Чрезвычайно

сложно по значению функции Фейтеля ***F(R,K)*** вычислить значение ее аргумента ***B*** при неизвестном значении ключа ***K.*** Сложность подбора ключа прямо зависит от его длины.

Для того чтобы повысить криптостойкость алгоритма DES, был разработан его усиленный вариант, называемый тройным алгоритмом DES, который включает троекратное шифрование с использованием двух разных ключей. При этом можно считать, что длина ключа увеличивается с 56 до 112 битов, а значит, криптостойкость алгоритма существенно повышается. Но за это приходится платить производительностью — тройной алгоритм DES требует в три раза больше времени на реализацию, чем «обычный».

В 2001 году Национальное бюро стандартов США приняло новый стандарт симметричного шифрования, который получил название ***AES*** (Advanced Encryption Standard). Стандарт AES был разработан в результате проведения конкурса на разработку симметричного алгоритма шифрования, обладающего лучшим, чем у DES, сочетанием показателей безопасности и скорости работы. Победителем был признан алгоритм Rijndael, который и был положен в основу AES. В результате AES обеспечивает лучшую защиту, так как использует 128-битовые ключи (а также может работать со 192- и 25б-битовыми ключами) и имеет более высокую скорость работы, кодируя за один цикл 128-битовый блок в отличие от 64-битового блока DES. В настоящее время, кроме AES, достаточно распространенным симметричным алгоритмом шифрования является алгоритм Blowfish.

Криптостойкость всех симметричных алгоритмов зависит от качества ключа, это предъявляет повышенные требования к службе генерации ключей, а также к надежности канала обмена секретными ключами между участниками секретных переговоров.

Проблема распределения ключей

Симметричный подход к шифрованию изначально несет в себе очевидную проблему, называемую *распределением ключей* (key distribution). Проблема состоит в следующем.

Отправитель и получатель хотят обмениваться секретными сообщениями, но в их распоряжении имеется незащищенный канал. Поэтому они вынуждены использовать шифрование, но чтобы послать зашифрованное сообщение, нужно предварительно обменяться секретной информацией о значении ключа. Значит надо защищенным способом передать секретный ключ, а его нельзя передать по открытому каналу. Если ключ зашифровать другим ключом, то опять возникнет проблема передачи второго ключа. Получается замкнутый круг.

Единственным по-настоящему надежным решением этой проблемы является передача ключа при личной встрече абонентов. Однако при активном обмене требуется часто менять ключи, чтобы не дать

возможности криптоаналитику собрать большое количество шифрованного материала, — известно, что чем больше зашифрованных сообщений окажется в руках криптоаналитика, тем легче ему раскрыть криптосистему. Кроме того, если злоумышленник перехватывает и сохраняет сообщения, зашифрованные одним и тем же ключом, то при раскрытии данного ключа они ***все*** окажутся скомпрометированными. Следовательно, необходимы частые личные встречи абонентов для обмена ключами, что, во-первых, не всегда возможно, а во- вторых, вообще делает бессмысленным обмен данными по каналу связи — действительно, зачем шифровать данные, если их можно лично передать при встрече.

Другим, менее надежным способом распределения ключей является использование курьеров или других вариантов защищенной доставки ключей, это решение тоже имеет очевидные изъяны. Существуют и другие приемы, не решающие, но смягчающие проблему распределения ключей. Например, у абонента может быть несколько секретных ключей, которые он должен использовать по разному назначению. Один ключ выдается ему на долгий срок, этот ключ применяется только для шифрования (дешифрования) других, кратковременных ключей, каждый из которых действителен только на время одного сеанса связи. И хотя в этом случае все равно остается проблема доставки долговременного ключа, уже нет необходимости его частой смены, так как этот ключ используется относительно редко и шифрует небольшие порции данных — сеансовые ключи. Зато теперь появляется возможность защищенной доставки сеансового ключа в зашифрованном виде, его можно часто менять, а значит, передаваемые данные менее подвержены риску компрометации.

Несмотря на различные усовершенствования процедуры распределения ключей, они не могут полностью устранить коренной изъян симметричных методов — ***необходимость доставки секретного ключа по незащищенному каналу.***

Если проблема с ключами возникает в системе с двумя абонентами, то она многократно усугубляется в системе с большим числом абонентов (рис. 4.7). Пусть, например, ***п*** абонентов желают обмениваться секретными данными по принципу «каждый с каждым», в этом случае потребуется ***п(п*** — 1)/2 ключей, которые должны быть сгенерированы и распределены надежным образом. То есть количество требуемых ключей пропорционально квадрату количества абонентов, что при большом числе абонентов делает задачу чрезвычайно сложной. Но именно такая ситуация наблюдается во всех современных

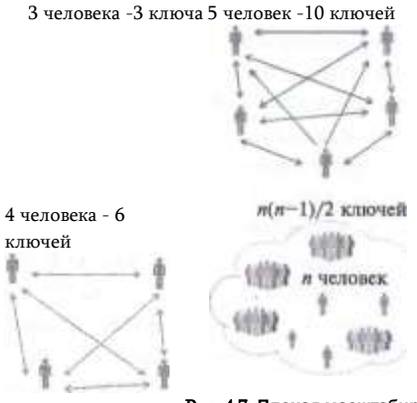


Рис. 4.7. Плохая масштабируемость симметричного алгоритма шифрования — количество ключей, требуемых для обмена секретными сообщениями, пропорционально квадрату участников обмена

Рис. 4.8. Уитфилд Диффи (Whitfield Diffie) и Мартин Хеллман (Martin Hell- man), открывшие в 1976 г. метод передачи секретного ключа без использования защищенного канала

сетях связи — телефонных, радио- и компьютерных сетях. Все это сделало чрезвычайно актуальной проблему распределения ключей.

В середине 70-х годов американские ученые Мартин Хеллман и Уилтфилд Диффи нашли способ, с помощью которого абоненты могли безопасно обмениваться секретными ключами без передачи их по каналу связи. Особенность этого открытия состоит в том, что оно противоречит всем интуитивным представлениям человека, делает возможным то, что кажется «очевидно» невозможным.

Метод Диффи-Хеллмана основан на использовании свойств односторонних функций.

Односторонняя функция (***one-way function***) — это функция ***y*** = ***F{x),*** которая легко вычисляется для любого входного значе-

ния ***x,*** но обратная задача — определение ***x*** по заданному значению функции ***y*** — решается очень трудно (рис. 4.9).

Рис. 4.9. Схема работы односторонней функции

Под односторонностью в данном случае понимается не то, что невозможно аналитически найти функцию ***F'*** такую, что ***x = F'(y).*** а то, что практически невозможно вычислить обратное значение, используя современные вычислительные средства, за обозримый интервал времени.

Рассмотрим несколько примеров односторонних функций.

• Простейшая функция двух аргументов ***F(p,q***) = ***pq.*** представляющая собой произведение двух простых чисел ***p*** и ***q,*** вычисляется сравнительно просто, даже если числа ***p*** и ***q*** очень большие. Но чрезвычайно сложно решить обратную задачу (называемую факторизацией): по произведению подобрать исходные два простых числа.

• Функция взятия модуля У(х) = xmodn, где операция xmodn («х по модулю п») дает в результате остаток от деления х на ***п.*** Эта функция является односторонней, так как, зная остаток от деления х на п, невозможно однозначно определить аргумент х.

• Функция У(х) = D^modP при некоторых ограничениях на параметры ***D*** и ***P*** является односторонней, т. е., зная У, а также параметры ***D*** и ***P,*** нельзя без экстраординарных вычислительных усилий найти аргумент х.

Пусть Алиса и Боб решили обмениваться шифрованными сообщениями, но в их распоряжении имеется только незащищенный, открытый канал связи, при этом никаких возможностей встретиться или передать секретный ключ через кого-нибудь другого у них нет.

В соответствии с алгоритмом Диффи-Хеллмана для успешного решения задачи Алиса и Боб должны выполнить следующие действия. Вначале они открыто договариваются о том, что будут использовать одностороннюю функцию У = D^xmodP. Затем они договариваются о значениях параметров ***D*** и ***P.*** Пусть, например, они договорились, что ***D*** = 7 и ***P*** = 13, т. е. функция имеет вид У = I^x mod 13. Еще раз подчеркнем, что в соответствии с алгоритмом Диффи-Хеллмана нся эта информация не является секретной, и даже, если переговоры будут подслушаны Евой, это не даст ей возможности прочитать юобщения Алисы и Боба. Дальнейшие действия участников обмена описываются в табл. 4.2.

В результате описанной процедуры на шаге 4 Алиса и Боб получили одно и то же число 3! Математические преобразования показы- мгот, что вычисления Алисы и Боба всегда будут давать одинаковые

Таблица 4.2

Действия Алисы и Боба в соответствии с алгоритмом Диффи-Хеллмана

Действия Алисы		Действия Боба	
1. Алиса секретным образом выбирает произвольное число *A* (закрытый ключ Алисы)	Пусть, например, *A* = 2	Боб также секретно выбирает произвольное число *B* (закрытый ключ Боба)	Пусть, например, *B* = 4
2. Алиса вычисляет значение *a* односторонней функции *Y*, используя в качестве аргумента свое секретное число *A*, т. е. $a = D^A$ modP (открытый ключ Алисы)	$a = 7^2 \bmod 13 = 10$	Боб также вычисляет значение *Ь* односторонней функции *Y*, используя в качестве аргумента свое секретное число *B*: Ъ = D^B mod *P* (открытый ключ Боба)	$Ь = 7^4 \bmod 13 = 2401 \bmod 13 = 9$
3. Алиса посылает Бобу свой открытый ключ *a*		Боб посылает Алисе свой открытый ключ *Ь*	
4. Алиса, получив от Боба число 6, вычисляет по формуле $K = b^A \bmod P$ (разделяемый секретный ключ)	$K = 9^2 \bmod 13 = 3$	Боб, получив от Алисы число в, вычисляет разделяемый секретный ключ по формуле $K = a^B \bmod P$ (разделяемый секретный ключ)	$K = 10^4 \bmod 13 = 3$
5. По правилам модульной арифметики $b^A \bmod P = (D^B \bmod P)^A \bmod P = D^{BA} \bmod P$		По правилам модульной арифметики $a^e \bmod P = (D^A \bmod P)^e \bmod P = D^{AB} \bmod P$	$K = 3$

результаты. Полученные в результате числа они могут использовать в качестве известного только им ключа для различных симметричных методов шифрования.

Посмотрим, может ли Ева подобрать разделяемый секретный ключ Алисы и Боба. Пусть на шаге 3, когда Алиса и Боб посылали друг другу свои открытые ключи *a* (10) и Ъ (9), Ева смогла перехватить эти числа (ведь канал является открытым) и теперь пытается вычислить разделяемый секретный ключ. Зная число ***a,*** которое Алиса послала Бобу, Ева хочет повторить действия Боба и вычислить разделяемый секретный ключ по формуле 10® mod 13. Для этого ей требуется закрытый ключ Боба *B,* который он, однако, хранит секретно от всех. Зато Ева знает, что Боб использовал свой закрытый ключ *B,* когда вычислял значение своего открытого ключа — ***Ь.*** То есть задача будет решена, если Ева сможет подобрать такое ***B,*** чтобы 7® mod 13 = 9. Но именно это практически не разрешимо, поскольку 7® mod 13 является односторонней функцией. Таким образом, Алиса и Боб действительно получили секретный ключ.

Но вот, что могла бы сделать Ева, так это с самого начала «вклиниться» в переговорный процесс и перехватывать все сообщения Алисы, заменяя их своими собственными. Так же, как и Алиса, Ева должна согласовать с Бобом значения параметров *D* и *P,* выбрать число ***A*** в

качестве своего закрытого ключа (понятно, что он будет отличаться от закрытого ключа Алисы), вычислить свой открытый ключ и послать его Бобу. Получив открытый ключ Боба, псевдо-Алиса может сгенерировать разделяемый секрет и тем самым получить возможность обмениваться секретными посланиями с ничего не подозревающим Бобом. Другими словами, *алгоритм Диффи-Хеллмана не обеспечивает взаимной аутентификации абонентов,* они не могут быть уверены в том, с кем они сгенерировали разделяемый секретный ключ.

Для того чтобы усложнить решение обратной задачи, т. е. восстановления закрытого ключа Алисы или Боба по открытому, на параметры алгоритма накладываются некоторые ограничения, в том числе следующие:

- все параметры ***D,P,A,B*** должны быть целыми положительными числами;
- ***A*** и *B* должны быть большими числами порядка $10^{шо}$;
- *P* должно быть большим простым числом порядка $_1 о^{300}$, причем желательно, чтобы (*P* — 1)/2 также было простым числом;
- число ***D*** не обязательно должно быть большим, обычно оно выбирается меньше десяти, *D* < *P*.

И хотя алгоритм Диффи-Хеллмана стал прорывом в области криптографии, в его исходном состоянии он представлял скорее теоретическую, нежели практическую ценность. Устранив препятствие в виде необходимости надежного закрытого канала для передачи ключа, этот метод не снял проблемы квадратичной зависимости числа ключей от числа абонентов. Кроме того, отсутствие взаимной аутентификации абонентов и необходимость переговорной процедуры для вычисления каждого разделяемого секрета также затуманивало перспективы практического применения данного метода распределения ключей. Решение пришло очень скоро — уже через год после появления алгоритма Диффи-Хеллмана была теоретически доказана возможность принципиально нового подхода к шифрованию — асимметричного шифрования, при использовании которого (помимо прочих преимуществ) кардинально упрощается задача распределения ключей.

Асимметричные алгоритмы шифрования

Исторические предпосылки

До сравнительно недавнего времени понятие «симметричное шифрование» не существовало, просто потому, что все методы, которые использовались человечеством на протяжении нескольких тысяч

Рис. 4.10. Сотрудники секретной правительственной лаборатории Великобритании Джеймс Эллис, Клиффорд Кокс и Малколм Уильямсон, первыми открывшими асимметричный подход к шифрованию

лет, по современной классификации могли быть отнесены к классу симметричных, и других не было. Более того, все эти тысячи лет существовала твердая убежденность, что других схем, кроме симметричной — «отправитель шифрует с помощью секретного ключа, получатель с помощью этого же ключа расшифровывает» — в принципе не может быть никогда!

Революция свершилась в конце 60-х — середине 70-х, когда с разницей в несколько лет две группы ученых, одна из которых — уже знакомые нам Диффи и Хеллман, а другая — сотрудники секретной правительственной лаборатории Великобритании12 Джеймс Эллис (James Ellis), Клиффорд Кокс (Clifford Cocks) и Малколм Уильямсон (Malcolm Williamson), независимо друг от друга изобрели принципиально новый подход к шифрованию, открывающий глобальные перспективы в области современных коммуникаций (рис. 4.10). Предельно упрощая, этот подход можно описать фразой: «отправитель шифрует сообщение с помощью своего секретного ключа, а получатель расшифровывает его с помощью другого, собственного ключа». Как видим, здесь на двух сторонах обменного канала используются разные ключи, т. е. присутствует асимметрия, соответственно все методы, основанные на таком подходе, стали называть «асимметричными».

Это удивительно, что за несколько тысяч лет не было ни одной известной науке попытки изобретения асимметричного метода шифрования, и вдруг, практически одновременно, две независимые группы ученых совершают это открытие! Возможно, причина кроется в том, что к концу 60-х годов произошло совпадение двух обстоятельств — во-первых, возникла острая потребность в новом типе шифрования и, во-вторых, появились технические возможности реализации этой идеи.

12 Известно, что исторически первыми были британские криптографы, которые открыли асимметричное шифрование на 6 лет раньше, чем Диффи и Хеллман, однако до 1997 года они не смогли обнародовать свои результаты, так как их работа имела гриф секретности.

Потребность была продиктована зрелостью таких видов массовых коммуникаций как телефон, радио, компьютерные сети, для которых, во-первых, особенно важна секретность в виду слабой защищенности публичных средств связи, а во-вторых, неприемлемы ограничения традиционных методов шифрации, выражающиеся в необходимости обмена секретным ключом для каждой пары абонентов.

К концу 60-х годов стали отчетливо вырисовываться перспективы использования Интернета как мировой сети связи, и одновременно с этим стало приходить осознание того, что глобальная публичная сеть может выполнить свою миссию только, если миллионам ее пользователей будет предоставлена возможность защищенного обмена сообщениями. Эти темы особенно волновали военных разных стран, которых очень привлекала возможность распределенного управления вооруженными силами, но пугала невозможность гарантировать секретность передаваемых директив. И если в недалеком прошлом проблема распределения секретных ключей, хотя и существовала, но была преодолимой, то в новых условиях она стала принципиальным препятствием.

К этому времени созрели технические возможности реализации вычислительно ёмких алгоритмов шифрования, к которым могут быть отнесены асимметричные алгоритмы шифрования. Массовое распространение получили компьютеры, обладающие такой вычислительной мощностью, которой до сих пор могли похвастаться только уникальные модели суперкомпьютеров. Это сделало шифрование обыденной операцией, которая может быть выполнена на обычном персональном компьютере.

Вот на таком историческом фоне была предложена концепция асимметричной криптосистемы, называемой также ***шифрованием с открытым ключом.***

Концепция шифрования с открытым ключом

На рис. 4.11 представлена модель асимметричной криптосистемы. Так же, как и в модели симметричного шифрования на рис. 4.4, здесь показаны три участника: отправитель (Боб), получатель (Алиса) и злоумышленник (Ева). В отличие от симметричной схемы шифрования, в которой наличие разделяемого секретного ключа автоматически означает возможность двустороннего защищенного обмена, здесь существует отдельная процедура для передачи зашифрованных

Рис. 4.11. Модель криптосхемы с открытым ключом D

сообщений в каждую из сторон. Рис. 4.11 иллюстрирует вариант, когда зашифрованные сообщения могут быть посланы только Бобом в сторону Алисы, но не наоборот.

1. Итак, Алиса пожелала, чтобы Боб посылал ей зашифрованные сообщения. Для этого она сгенерировала пару ключей: ***открытый ключ E*** (public key) и ***закрытый ключ D*** (private key). Для шифрования текста служит открытый ключ, но расшифровать этот текст можно только с помощью закрытого ключа. Алиса не хочет, чтобы кто-либо читал ее почту, поэтому она сохраняет закрытый ключ ***D*** (часто называемый также личным ключом) в секрете. Открытый же ключ ***E*** Алиса свободно передает всем, от кого хочет получать зашифрованные сообщения. Открытый ключ не представляет никакого секрета, Алиса может поместить его на своей странице в Фейсбуке или обнародовать в рекламе на телевидении. Все, кто хочет посылать Алисе зашифрованные сообщения, используют один и тот же ключ ***E,*** но при этом никто из них не может прочитать сообщения друг друга.

2. Алиса передает Бобу свой открытый ключ ***E*** по незащищенному каналу в незашифрованном виде.

3. Боб шифрует свое сообщение ***X*** открытым ключом Алисы ***E*** и посылает зашифрованный текст ***Y = FE{X***) по открытому каналу. Никто не может прочитать это сообщение. Даже сам Боб, если бы ему, вдруг, захотелось перечитать, что он там написал, не смог бы этого сделать, потому что для этого нужен закрытый ключ Алисы, которого у него нет.

4. Алиса получает шифрованное сообщение ***Y - FE(X***) и расшифровывает его своим закрытым ключом ***D. X = F'D(Y).***

Для того чтобы не только Боб, но и Алиса могла посылать защищенные сообщения, Боб должен аналогичным образом сгенерировать пару ключей и отправить свой открытый ключ Алисе. С его помощью Алиса сможет шифровать и отсылать сообщения Бобу, которые не сможет прочитать никто, кроме Боба.

Очевидно, что числа-ключи, одно из которых служит для шифрования текста, а другое — для дешифрования, не могут быть независимыми друг от друга, а значит, есть теоретическая возможность вычисления закрытого ключа по открытому. Однако это связано с огромным объемом вычислений, которые требуют соответственно огромного времени. Поясним принципиальную связь между закрытым и открытым ключами следующей аналогией.

ПРИМЕР-АНАЛОГИЯ

Пусть руководитель предприятия (на рис. 4.12 это пользователь 1) решает вести секретную переписку со своими сотрудниками. Рассмотрим вариант, когда требуется обеспечить конфиденциальность потока сообщений только в одну сторону — от сотрудников к руководителю. Для этого руководитель решает использовать какой-либо малоизвестный язык, например санскрит. С этой целью он обзаводится единственной копией санскритско-русского словаря, которую хранит в сейфе, и большим количеством широкодоступных русско-санскритских словарей, которые раздает всем своим сотрудникам.

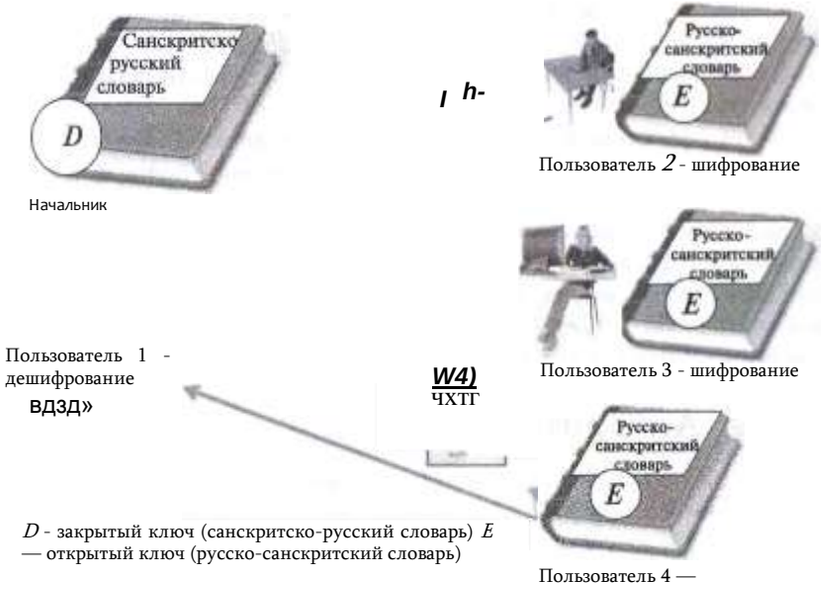


Рис. 4.12. Использование шифрования с открытым ключом для обеспечения конфиденциальности

Когда у сотрудников возникает необходимость написать секретное сообщение руководителю, они, пользуясь словарем, пишут сообщения на санскрите. Руководитель переводит сообщения на русский язык, пользуясь доступным только ему санскритско- русским словарем. Очевидно, что здесь роль открытого ключа *E* и закрытого ключа *D* руководителя играют русско-санскритский и санскритско-русский словари соответственно. Могут ли пользователи 2, 3 и 4 прочитать чужие сообщения 5г, S_3, S_4, которые посылает каждый из них руководителю? Вообще-то нет, так как для этого им нужен санскритско-русский словарь, обладателем которого является только пользователь 1. Так обеспечивается конфиденциальность потока сообщений в направлении руководителя.

Заметим, что у сотрудников имеется теоретическая возможность для разгадывания сообщений друг друга, так как затратив массу времени, можно прямым перебором составить санскритско-русский словарь по русско-санскритскому. Такая очень трудоемкая процедура, требующая больших затрат времени, отдаленно напоминает восстановление закрытого ключа по открытому.

Для того чтобы в сети все п абонентов имели возможность не только принимать зашифрованные сообщения, но и сами посылать таковые, каждый абонент должен обладать собственной парой ключей ***E*** и ***D.*** Всего в сети будет 2п ключей: ***п*** открытых ключей для шифрования и п секретных ключей для дешифрования. Таким образом решается проблема ***масштабируемости*** *— квадратичная зависимость количества ключей от числа абонентов в симметричных алгоритмах заменяется линейной зависимостью в асимметричных алгоритмах.* Решается и проблема доставки ключа, поскольку теперь он не является секретом, его можно без опаски передавать по открытому каналу. Злоумышленнику нет смысла стремиться завладеть открытым ключом, поскольку это не дает возможности расшифровывать текст или вычислить закрытый ключ.

Хотя информация об открытом ключе не является секретной, ее нужно защищать от подлогов, чтобы злоумышленник под именем легального пользователя не навязал свой открытый ключ, после чего с помощью своего закрытого ключа он сможет расшифровывать все сообщения, посылаемые легальному пользователю, и отправлять свои сообщения от его имени. Решение проблемы дает технология цифровых сертификатов13 — электронных документов, которые связывают конкретных пользователей с конкретными открытыми ключами.

Алгоритм RSA

Открыватели асимметричного подхода к шифрованию показали концептуальную возможность существования таких функций, на основе которых можно построить криптографическую систему, в кото-

13 См. раздел «Аутентификация на основе цифровых сертификатов» в главе 5 «Технологии аутентификации»

ключ, можно вычислить значение закрытого ключа. Однако необходимым промежуточным действием в этом преобразовании является нахождение

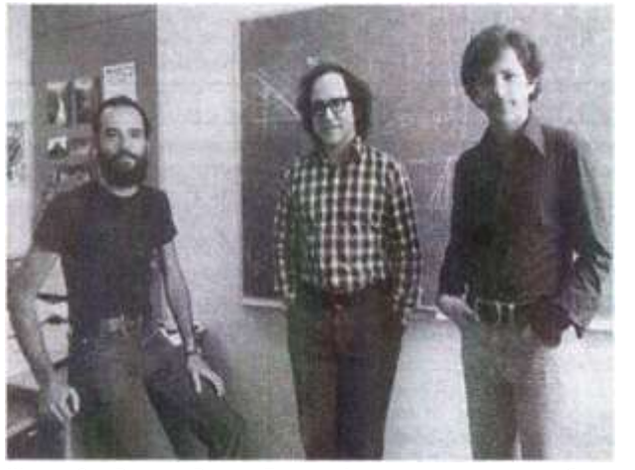


Рис. 4.13. Шамир, Эди Рональд Ривест, Леонард Адлеман — ученые, разработавшие RSA-алгоритм шифрования с открытым ключом

рой текст шифруется одним ключом, а расшифровывается — другим. Они также обрисовали те перспективы, которые открывает этот подход в деле решения проблемы распределения ключей. Ими были сформулированы два принципиальных требования, которым должны удовлетворять функции асимметричной криптосистемы:

- зашифрованное сообщение должно быть результатом вычислений односторонней функции, так чтобы никто не мог выполнить обратные преобразования и получить исходный текст;
- эта односторонняя функция должна быть сконструирована таким образом, чтобы у нее был некоторый секретный элемент, зная который, получатель шифровки, обладающий знанием этого секрета, мог легко выполнить обратное преобразование.

Функции, которые удовлетворяют данным требованиям назвали ***односторонними функциями с потайным входом*** (*trapdoor function).* Некоторое время ученым не удавалось найти ни одну функцию, удовлетворяющую критериям Диффи-Хеллмана, и идея асимметричного шифрования не находила практического применения. Наконец, в 1978 г. трое ученых американских ученых Ривест, Шамир и Адлеман (рис. 4.13) предложили долгожданный алгоритм асимметричного шифрования ***RSA,*** названный так по первым буквам их фамилий — Rivest, Shamir, Adleman. В

простых чисел ***P*** и ***Q,*** для чего нужно разложить на простые множители очень большое число ***N,*** а это является чрезвычайно трудоемкой процедурой. Таким образом, здесь мы имеем дело с од-

табл. 4.3 описываются основные шаги алгоритма RSA.

Еве для того, чтобы прочитать перехваченное сообщение ***C,*** требуется закрытый ключ Алисы (***D,N***). Но в ее распоряжении имеется только открытый ключ (***E,N***). Теоретически, зная открытый

Таблица 4.3

Последовательность действий участников обмена данными в соответствии с RSA-алгоритмом

Действия Алисы и Боба	Числовой пример
Алиса произвольно выбирает два случайных простых числа *P* и *Q*. Они должны быть очень большими — от этого зависит стойкость алгоритма шифрования.	В примере для простоты расчетов берутся очень маленькие числа. Пусть *P* = 7 и *Q* = 13
Алиса вычисляет два произведения *N* = *PQ*. M= (P - 1)(<3 - 1)	*N* = 91; *M* = б • 12 = 72
Алиса выбирает случайное целое число *E*, меньшее M и не имеющее с ним общих сомножителей	*E* = 5
Пара *(E,N)* — это открытый ключ Алисы, который она передает всем, от кого хочет получать шифрованные сообщения. Алиса посылает Бобу свой открытый ключ (*E,N*)	(5, 91)
Алиса находит *D* такое, что *DE* = 1 mod *M*. Пара *(D,N)* — это закрытый ключ Алисы, который она не показывает никому. С этого момента она готова получать зашифрованные сообщения от Боба	*D* ■ 5 = 1 mod 72; *D* = 29 (это число легко находится подбором, если учитывать признаки делимости на 5)
Боб получил открытый ключ Алисы и так же как все остальные, имеющие доступ к этому ключу, может посылать Алисе зашифрованные сообщения. Он представляет свое сообщение в любом цифровом формате и разбивает его на блоки *X* таким образом, чтобы O < *X* < *N*	Пусть секретный текст, посылаемый Бобом, состоит из одного символа R, который в коде ASCII имеет значение 1010010, или в десятичном коде 82
Боб шифрует сообщение *X* открытым ключом *(E,N): C* = X^E *mod N* и посылает Алисе зашифрованное сообщение С Алиса получает сообщение *C* и расшифровывает его своим закрытым ключом (Л, *N*): *X* = C^D mod *N*	*C* = 82^5 mod 91 = {82^3mod91 82^2 mod 91} mod 91 = 10
	X = 10^{29} mod 91 = {10^1 mod 91 ■ 10^4 mod 91-10® mod 91...} mod 91 (где внутри фигурной скобки последний сомножитель 10^6 mod 91 повторяется четыре раза) 10 mod 91 = 10; 10^4 mod 91 = 81; 10® mod 91 = 1;
	X = {10 ■ 81 ■ 1} mod 91 = 82 Результат

расшифровки *X* = 82 совпадает с исходным секретным сообщением

Примечание, Вычисление модуля от степени числа упрощается, если использовать следующее правило: У°+ь+с mod *p* - (у<* mod] *p* . у!> тос| р , ус тос] р) mod] *p*

посторонней функцией ***N = PQ.*** Но для Алисы это же действие — разложение большого числа на два простых множителя — не представляет никакого труда, потому, что она знает, как сконструировано это число ***N,*** она сама его вычислила, произвольно выбрав два сомножителя. Другими словами, Алисе известен «потайной вход» этой односторонней функции.

Продемонстрируем сложность разложения ***N*** на примере. Пусть ***N*** = 46843969. Это число получено умножением двух простых чисел. Эта операция, выполненная на простейшем калькуляторе, заняла доли секунды. Сделать обратное преобразование с помощью того же калькулятора несравнимо сложнее: для этого надо для каждого простого числа проверить, делится ли на него число 46843969 без остатка. Успех придет только после того, когда проверяющий, сделав примерно 700 циклов, дойдет до числа 5119 и получит в результате простое число 9151.

В случае, когда в качестве ***P*** и ***Q*** выбираются очень большие числа (сотни десятичных знаков), сложность подбора возрастает многократно. Именно с огромной вычислительной сложностью разложения большого числа ***N*** на простые множители ***P*** и ***Q*** связана высокая криптостойкость алгоритма RSA.

Атаки на криптосистемы

Для взламывания криптоалгоритмов используются два принципиально разных подхода:

- атаки на криптосистему как математический объект;
- атаки на систему реализации криптоалгоритма — так называемые атаки по побочным каналам.

Первый класс атак, уже не раз упоминавшийся в этой главе, заключается в выяснении секретного ключа, что в симметричных алгоритмах сводится к прямому перебору всех возможных чисел заданной разрядности, а в асимметричных алгоритмах — к разложению на сомножители произведения двух простых чисел. Поскольку вычислительные мощности год от года растут, то время, за которое атакующий может вычислить ключ простым перебором, сокращается. Кроме того, активно ведутся работы по совершенствованию методов разложения больших чисел, которые также сокращают время подбора ключа.

Известно, что длительность этих процедур подбора ключа зависит от их разрядности, поэтому главным средством противостояния атакам на криптоалгоритмы является увеличение длины используемых в этих алгоритмах ключей. Именно это мы и наблюдаем в последние годы, когда в ответ на очередное известие об успешной попытке разложения RSA-числа (так называют числа, полученные умножением двух простых чисел) изменяются стандартные требования к их

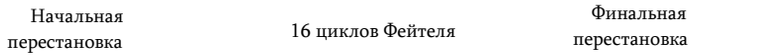


Рис. 4.14. Измерение уровня энергопотребления при выполнении алгоритма DES

длине. Еще недавно стандартом были 56 битов для алгоритма DES и 500 битов для алгоритма RSA, а сегодня, для того чтобы гарантировано обеспечить требуемую стойкость шифрования, стандарты подняты соответственно до 128 битов (AES) и 2048 битов (RSA). Эксперты утверждают, что вероятность взлома ключей такой длины в течение как минимум следующих 10 лет чрезвычайно мала. На момент написания этого текста наибольшее успешно раложенное RSA-число имело длину 768 битов (232 десятичных разряда).

Принципиально другим подходом к раскрытию криптосистемы являются *атаки по побочным каналам.* Этот сорт атак основывается на анализе параметров работы программно-аппаратного комплекса, реализующего криптоалгоритм. Для каждого конкретного сочетания программы и модели процессора некоторые параметры процесса вычислений — время получения результата, потребляемая мощность, электромагнитное излучение, даже громкость и характер звуков, издаваемых устройством — все они отражают специфику шифруемого текста и поэтому могут быть в той или иной степени использованы для раскрытия криптосистемы.

Так, в 1998 году была успешно проведена *атака по времени выполнения* (timing attack) против криптоалгоритма RSA. В ней использовалось предположение о различиях во времени выполнения различных операций алгоритма (умножения и возведение в степень). На основании анализа статистических данных о времени вычислений было определено значение секретного ключа.

Аналогичным образом проводятся *атаки по уровню энергопотребления* (power analysis attack) вычислительной системы во время выполнения криптоалгоритма. На рис. 4.14 показаны графики энергопотребления14 микросхемы смарт-карты при выполнении ею программы шифрования по алгоритму DES. На верхнем графике можно

14 См. раздел «Аутентификация информации. Цифровая подпись» в главе 5 «Технологии аутентификации».

ясно разглядеть фазу начальной перестановки, 16 циклов Фейтеля и фазу финальной перестановки. На нижнем графике приведены более детальные измерения энергопотребления на втором и третьем цикле. На основе анализа такого рода измерений можно успешно раскрыть секретный ключ.

Для противодействия атакам по побочным каналам используются экранирование, добавление шума, генерируемого как программно, так и аппаратно, уравнивание времени выполнения операций добавлением соответствующих задержек для каждого типа операций и множество других достаточно естественных приемов.

Сравнение симметричных и асимметричных методов шифрования

Алгоритмы шифрования с открытым ключом самим своим появлением обязаны остро проявившимся недостаткам симметричных методов шифрования, а именно плохой масштабируемости процедуры распределения ключей и потребности в надежном канале связи для передачи ключей. Эти проблемы были успешно решены асимметричными методами шифрования. Более того, асимметричный подход открыл новые возможности для аутентификации передаваемых сообщений, т. е. доказательства того, что передаваемое сообщение принадлежит именно тому, кто его подписал*. Однако всем этим дополнительным функциональным возможностям симметричных алгоритмов (в частности, RSA) симметричные алгоритмы (в частности, AES и DES) противопоставляют более высокую производительность.

При том, что в обоих подходах на вход функции шифрования поступает блок подлежащих шифрованию данных, представленных в числовом виде, характер операций, которые выполняются в каждом из этих двух случаев, существенно различается:

- симметричные функции шифрования AES и DES интерпретируют блок данных как ***последовательность битов,*** и шифрование представляет собой выполнение экономных в вычислительном плане операций над битами — перестановок, подстановок, циклических сдвигов, логических операций двоичной арифметики;
- в асимметричном алгоритме RSA блок исходных данных интерпретируется как одно очень большое ***целое число,*** над которым производятся вычислительно емкие арифметические операции — в основном возведение в степень и взятие модуля.

* «Introduction to Differential Power Analysis and Related Attacks», By Paul Kocher, Joshua Jaffe, and Benjamin Jun

Таблица 4.4

Сравнительные характеристики AES и RSA

Характеристика	Симметричные алгоритмы (AES, DES)	Асимметричные алгоритмы (RSA)
Характер функции шифрования	Операции над битами — перестановки, подстановки, логические операции двоичной арифметики	Операции над числами — возведение в степень, взятие модуля
Масштабируемость схемы распределения ключей	Низкая. Количество ключей квадратично зависит от числа абонентов	Высокая. Количество ключей линейно зависит от числа абонентов
Распределение ключей	Требуется защищенный канал для обмена ключами	Ключи могут быть переданы по открытому каналу
Длина ключа	128 битов	Не менее 1024 бита
Скорость шифрования	Существенно выше, чем у RSA	Существенно ниже, чем у AES
Время генерации ключа	Миллисекунды	Минуты
Наименее затратный криптоанализ	Перебор по всему ключевому пространству	Разложение числа на простые множители
Возможность аутентификации сообщений	Отсутствует	Имеется (механизм электронной подписи)

В связи с этим программная реализация криптоалгоритмов типа RSA существенно менее производительна, чем реализации классических симметричных криптоалгоритмов типа DES и AES.

С учетом достоинств и недостатков обоих подходов к шифрованию (табл. 4.4) их часто используют в комбинации: например, для шифрования небольших объемов информации, таких как сессионные секретные ключи, применяют RSA, а основную часть пересылаемой информации шифруют с помощью симметричного алгоритма AES.

Односторонние функции шифрования. Обеспечение целостности

В области информационной безопасности особое место занимает специальный класс односторонних функций, называемых хеш-функциями. ***Хеш-функцией*** *(hash function)* называют одностороннюю функцию, которая, будучи примененной к некоторым данным, дает в результате значение, состоящее из фиксированного сравнительно небольшого и не зависящего от длины исходных данных числа байтов. Результат работы хеш-функции называют ***хеш-кодом*** или ***дайджестом.***

Рассмотрим, например, функцию взятия модуля $Y(x)$ = xmodn, где операция xmodn ***(«x*** по модулю га») дает в результате остаток от деления x на n. Эта функция, во-первых, является односторонней,

так как, зная остаток от деления x на n, невозможно однозначно определить значение аргумента x, во-вторых, она относится к классу хеш-функций, поскольку ее результат не зависит от аргумента x и всегда находится в диапазоне от 0 до (п — 1).

Хеш-функции называют также ***односторонними функциям шифрования*** (ОФШ), где в качестве шифрованного представления исходных данных выступает дайджест. При этом знание дайджеста *не позволяет* и даже *не предполагает* восстановления исходных данных. Для чего же нужны односторонние функции шифрования?

Для ответа на этот вопрос рассмотрим несколько примеров. Пусть требуется обеспечить целостность сообщения, передаваемого по сети. Отправитель и получатель договорились, что они будут использовать одностороннюю функцию Я, в качестве параметра которой будет использоваться секретное число — ключ ***К.*** Прежде чем отправить сообщение ***X,*** отправитель вычисляет для него дайджест $M = H(X, K)$ и отправляет его вместе с сообщением X адресату (рис. 4.15,а). Адресат, получив данные X *и* M, применяет ту же самую ОФШ к переданному в открытом виде исходному сообщению X, используя известный ему секретный ключ K: $M' = H(X, K)$. Если значения вычисленного локально M' и полученного по сети M дайджестов совпадают, значит, содержимое сообщения не было изменено во время передачи.

Таким образом, хотя знание дайджеста не дает возможности восстановить исходное сообщение, оно позволяет *проверить целостность данных.* Можно ли считать в таком случае, что дайджест является своего рода *контрольной суммой* для исходного сообщения? И да, и нет. Контрольные суммы применяются тогда, когда нужно обнаружить ошибки, вызванные техническими неполадками, например помехами в линии связи. Однако это средство не распознает модификацию данных злоумышленником, который, подменив сообщение, может просто добавить к нему заново вычисленную контрольную сумму. В отличие от контрольной суммы дайджест вычисляется с использованием *параметра* — разделяемого секретного ключа. Поскольку значение секретного ключа для ОФШ известно только отправителю и получателю, любая модификация исходного сообщения будет немедленно обнаружена. Следует отметить, что использование здесь разделяемого секрета неизбежно приводит к известным проблемам рапределения ключей — необходимости надежного канала для передачи ключа и плохой масштабируемости.

На рис. 4.15Ф показан другой вариант использования односторонней функции шифрования для обеспечения целостности данных. Здесь односторонняя функция ***Н*** не имеет параметра-ключа, но зато

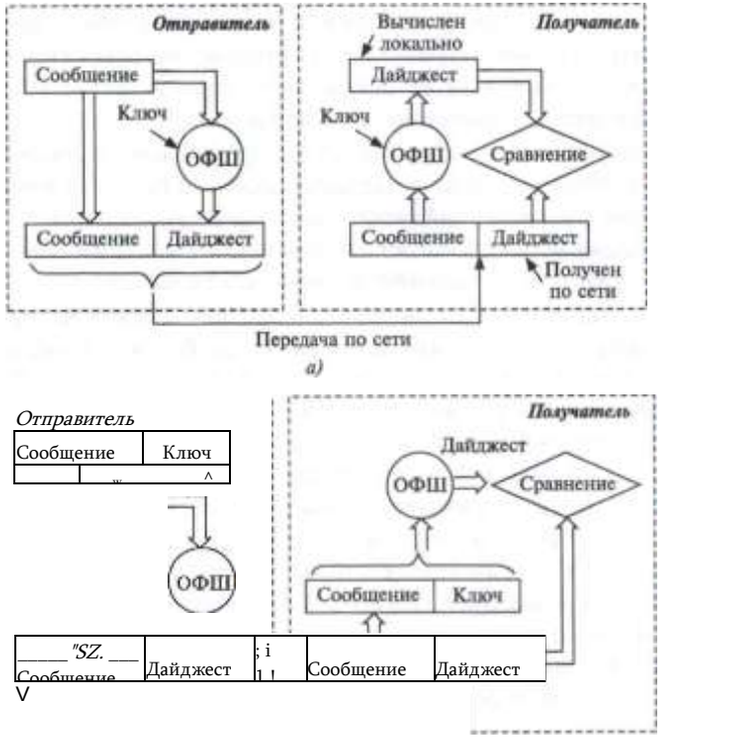


Рис. 4.15. **Использование односторонних функций шифрования для контроля целостности**

применяется не просто к сообщению, а к сообщению, дополненному секретным ключом. Получатель извлекает из полученных по сети данных исходное сообщение, потом дополняет его тем же известным ему секретным ключом и применяет к полученным данным одностороннюю функцию. Результат вычислений сравнивается с полученным по сети дайджестом.

Построение односторонних функций, пригодных для использования в криптографии, является нетривиальной задачей. Такого рода функции должны удовлетворять нескольким условиям:

- они должны быстро вычисляться;
- по дайджесту ***M,*** вычисленному с помощью функции Я, должно быть

невозможно за обзримое время вычислить исходное сообщение ***X***

- по дайджесту ***M,*** вычисленному с помощью данной функции Я, должно быть невозможно за обозримое время найти другое зна-

чение ***X'***, имеющее то же значение дайджеста ***H(X')*** = ***H(X)***\

- дайджесты ***H{X)*** и ***H(X'),*** вычисленные с помощью функции Я для очень близких значений ***X*** и ***X'***, должны значительно отличаться друг от друга.

Хеш-функции широко используются в сетевых протоколах, в алгоритмах электронно-цифровой подписи, в механизмах аутентификации на основе паролей. Наиболее популярными в системах безопасности в настоящее время является серия хеш-функций MD2, MD4, MD5. Все они генерируют дайджесты фиксированной длины 16 байтов. Адаптированным вариантом MD4 является американский стандарт SHA, длина дайджеста в котором составляет 20 байтов. Компания IBM поддерживает односторонние функции MDC2 и MDC4, основанные на алгоритме шифрования DES.

Вопросы к главе 4

1. Что из перечисленного может быть секретным ключом криптосистемы?

а) геометрическая фигура;

б) сборник детских сказок;

в) ключ от квартиры;

г) скитала;

д) число 5.

2. В каких из перечисленных ниже средствах обеспечения безопасности используется шифрование:

а) аутентификация;

б) авторизация;

в) антивирусные системы;

г) защищенный канал;

д) сетевой экран прикладного уровня;

е) фильтрующий маршрутизатор;

ж) цифровая подпись.

3. Сколько циклов перестановок и сдвигов выполняется в алгоритме DES?

а) 16;

б) 32;

в) 64;

г) 128;

Д) 192.

4. С ключами какой длины работает алгоритм AEF?

а) 64;

б) 128;

в) 192;

г) 256;

Д) 512.

5. Как можно передать секретный ключ?

а) при личной встрече;

б) с помощью надежного курьера;

в) по высокоскоростному открытому каналу;

г) по открытому каналу с помощью алгоритма Диффи-Хеллмана;

д) сеансовый ключ можно передать в зашифрованном виде по открытому каналу.

6. Сколько секретных ключей требуется для переписки 50 человек «каждый с каждым», использующих алгоритм DES?

а) 50;

б) 100;

в) 1225;

г) 2450;

д) 2500.

7. Какие из нижеперечисленных свойств определяют одностороннюю функцию У(х)?

а) функцию У (ж) чрезвычайно сложно вычислить для любого входного значения ж;

б) функция У (ж) легко вычисляется для любого входного значения ж;

в) аргумент *х* легко вычислить по известному У;

г) вычисление *х* по известному У чрезвычайно затруднено;

д) вычисление *х* по известному У абсолютно невозможно.

8. Какая функция из перечисленных является односторонней?

а) У (ж) = xmodn;

б) функция двух аргументов $F(p,q) = pq$. представляющая собой произведение двух простых чисел р и д;

в) У (ж) = $ж^3$;

9. Алгоритм Диффи-Хеллмана:

а) решает проблему передачи секретного ключа по открытому каналу;

б) это то же самое, что и алгоритм RSA;

в) решает проблему масштабируемости распределения ключей;

10. Метод асимметричного шифрования;

а) смягчает проблему масштабирования для процедуры распределения ключей;

б) снимает требование защищенного способа передачи ключа;

в) ускоряет процедуру шифрования;

г) используется для шифрования только текстовой информации.

11. Какие из утверждений правильные?

а) открытый ключ нужно защищать от подглядывания;

б) открытый ключ нужно защищать от подмены;

в) теоретически возможно зашифровать текст одним ключом, а расшифровать другим, который никак не связан с первым;

г) название алгоритма RSA раскрывается как «алгоритм Ривеста (Rivest) и Ша- мира (Shamir)»;

д) алгоритм RSA основан на использовании функции разложения произведения двух простых чисел на сомножители.

12. RSA-число это:

а) открытый ключ алгоритма RSA;

б) закрытый ключ алгоритма RSA;

в) число, полученное умножением двух простых чисел;

г) результат умножения открытого ключа на закрытый.

13. Какие атаки предпринимаются на криптосистемы?

а) атака по времени выполнения;

б) атака по энергопотреблению;

в) разложение числа на простые множители;

г) подбор ключа методом прямого перебора;

д) декомпрессия зашифрованного текста;

е) ни один из перечисленных видов атак.

14. Что отличает алгоритмы AES от RSA?

а) AES — симметричный алгоритм, а RSA — асимметричный;

б) в алгоритме AES выполняются операции над битами — перестановки, подстановки, а в алгоритме RSA — операции над числами — возведение в степень, взятие модуля;

в) в алгоритме RSA выполняются операции над битами — перестановки, подстановки, а в алгоритме AES — операции над числами — возведение в степень, взятие модуля;

г) требуемая длина ключа в алгоритме AES существенно ниже, чем в алгоритме RSA;

д) скорость шифрования по алгоритму AES существенно ниже, чем по алгоритму RSA;

е) масштабируемость RSA хорошая, а AES — плохая.

15. Какими свойствами должна обладать хеш-функция?

а) быстро вычисляться;

б) иметь результат фиксированной длины;

в) иметь результат небольшой длины;

г) не позволять по значению функции вычислять значение аргумента;

д) результаты вычисления функции от близких значений аргументов не должны значительно отличаться друг от друга.

16. В каких случаях предпочтительнее использовать симметричные алгоритмы шифрования, а в каких — алгоритмы шифрования с открытым ключом?

5 ТЕХНОЛОГИИ АУТЕНТИФИКАЦИИ

Как было определено в первой главе, аутентификация применительно к вычислительной системе — это доказательство подлинности различных элементов этой системы при их взаимодействии. ***Пользователь*** при входе в систему должен предъявить системе доказательства, что он именно тот пользователь, идентификатор которого он вводит. Таким доказательством может служить пароль. ***Документ,*** полученный пользователем по электронной почте, должен сопровождаться дополнительной информацией, которая бы подтверждала пользователю, что документ не был изменен при передаче и что автором этого документа является именно тот человек, от имени которого это письмо было послано. Здесь доказательством может служить электронная подпись. ***Устройства,*** взаимодействующие по сети, должны доказать друг другу, что ни одно из них не подменен злоумышленником с целью ответвления или прослушивания трафика. Для этого в протоколе взаимодействия этих устройств должна быть предусмотрена процедура взаимной аутентификации. Взаимная аутентификация требуется и для организации безопасного сеанса пользователя и серверного ***приложения.*** Аутентификация может проводится не только по отношению к отдельному пользователю, но и к группе пользователей. Методы аутентификации различаются в зависимости от того, что служит аутентификатором, а также от того, каким образом организован обмен аутентификационными данными между аутентифицируемым и аутентифицирующим элементами системы.

Факторы аутентификации человека

Абсолютно надежная аутентификация человека представляет собой теоретически неразрешимую задачу. Нет такого аутентификатора, который со 100%-ной надежностью доказывал бы аутентичность человека. Пароль можно перехватить, электронный ключ украсть, отпечаток пальца подделать, радужную оболочку глаза подменить качественным изображением-: Более того, не существует научного доказательства невозможности совпадения у разных людей отпечатков

н.1/1Ьцев или радужных оболочек глаза. Даже совпадение результатов .•М.1ЛИЗЭ ДНК при современном уровне развития техники не может с тужить абсолютным доказательством аутентичности человека.

Однако на практике при аутентификации пользователей в вычисли 1ельных системах ограничиваются некоторым не 100%-ным, хотя и достаточно высоким уровнем достоверности доказательства аутен- |ичности человека. Аутентификаторы, которые используются при чом разделяют на 3 класса, определяемые следующим образом:

- «что-то, что знаю» — к этому типу относятся многоразовые и одноразовые пароли, правила преобразования информации;
- «что-то, что имею» — различные миниатюрные устройства, называемые электронными ключами или токенами;
- «что-то чем являюсь» — различные биометрические показатели аутентифицируемого.

Класс аутентификаторов называют *фактором.* Если в процедура аутентификации предусматривается предъявление аутентифицируемый нескольких аутентификаторов, относящихся к разным классам, ю такую аутентификацию называют многофакторной. Наибольшее распространение в настоящее время получила *двухфакторная аутентификация,* при которой пользователь предъявляет многоразовый пароль («что-то, что знаю») и аппаратный ключ («что-то, что имею»).

(ледует заметить, что в некоторых случаях термин «многофакторная аутентификация» используется для обозначении процедур *многоступенчатой аутентификации,* построенных на использовании нескольких аутентификаторов, относящихся к одному и тому же классу. Примером такой процедуры является аутентификация владельца банковского счета при его звонке в банк: сначала его просят назвать несколько букв из его пароля, а затем задают несколько вопросов с заранее (огласованными и зафиксированными в базе данных аутентифицирующей организации ответами, например, о его памятном географическом пункте, девичьей фамилии матери и т. п.

Многоразовые пароли

Пароль — последовательность символов, выбранная пользователем либо сгенерированная программным или аппаратным средством либо назначенная администратором;

Пароли бывают одноразовыми и многоразовыми. В процедурах аутентификации, основанных на *одноразовых паролях,* аутентифицируемый должен каждый раз предъявлять новое значение пароля. Обычно для генерации одноразовых паролей используется специальная программа или аппаратное устройство, о которых будет сказано позже.

Многоразовые пароли, как это следует из их названия, могут использоваться для доказательства аутентичности многократно. Механизмы аутентификации на основе многоразовых паролей, обладая простотой и логической ясностью, традиционно являются самым популярным средством аутентификации. Однако им свойственны известные слабости. Это, во-первых, возможность раскрытия и разгадывания паролей, во-вторых, возможность «подслушивания» пароля при его передаче по сети путем анализа сетевого трафика. В третьих, обладатели паролей могут стать жертвами социального инжиниринга. Так, например, беглый экс-сотрудник Агентства национальной безопасности США Эдвард Сноуден, работая системным администратором на разведовательной базе США на Гавайях, использовал логины и пароли более 20 своих сослуживцев, чтобы получить доступ к секретным файлам. Он получал эти данные, объясняя, что они необходимы ему для работы.

Для снижения уровня угрозы раскрытия паролей администраторы сети, как правило, применяют встроенные программные средства, служащие для формирования *политики назначения и использования паролей:* задание максимального и минимального сроков действия пароля, хранение списка уже использованных паролей, управление поведением системы после нескольких неудачных попыток логического входа и т. п.

Многие пользователи пренебрегают угрозами, которые несут в себе легко угадываемые пароли. Так, червь Mumu, поразивший компьютерные сети в 2003 году, искал свои жертвы, подбирая пароли из очень короткого списка: password, passwd, admin, pass, 123, 1234, 12345, 123456 и пустая строка. Такая на удивление примитивная стратегия дала прекрасные (с точки зрения атакующей стороны) результаты — множество компьютеров было взломано.

В отчете, опубликованном в августе 2013 года компанией Google, приводится следующий рейтинг 10 наиболее популярных паролей, использованных пользователями Интернет при доступе к Web-серверам:

- имя домашнего животного;
- дата бракосочетания или другого важного семейного события;
- дата рождения близкого родственника;
- имя ребенка;
- имя другого члена семьи;
- место своего рождения;
- любимый праздник;
- что-либо, связанное с любимой футбольной командой;
- имя близкой подруги/друга;
- слово «password».

Как видно из приведенного списка, для заинтересованного человека не составит большого труда подобрать эти пароли, для этого ему достаточно разузнать, как зовут собаку или кошку, имена и даты рождения членов семьи, и получить другую такую же доступную информацию. Не рекомендуется выбирать в качестве пароля числа в формате даты, а также слова, которые могут содержаться в словарях разных языков, географические названия, имена собственные и т. п., так как алгоритм подбора паролей обычно включает примитивную проверку всех слов из популярных словарей.

Но даже при выборе менее предсказуемого пароля вы все же рискуете, что этот пароль будет разгадан простым перебором всех возможных символов, такой метод часто называют методом полного перебора, или ***брутфорс-атакой*** (***brute-force attack)*.*** Время подбора прямо зависит от разнообразия набора символов, из которого вы формируете свой пароль, и длины пароля. В табл. 5.1 приведены данные, характеризующие стойкость паролей, состоящих из 6 и 8 знаков, сформированных из разных наборов символов. Время определялось для специальной программы подбора паролей, выполняемой на компьютере со средними характеристиками (fast PC, Dual Processor PC)**. Обратите внимание, насколько сильно увеличивается время подбора пароля при увеличении его длины всего лишь на два знака. Так, при использовании только букв латинского алфавита (заглавных и прописных) время подбора пароля из 8 знаков в 3000 раз больше, чем из б знаков!

Таблица 5.1

Сравнение стойкости паролей

Множество символов	Количество комбинаций		Время подбора пароля	
	б знаков	8 знаков	б знаков	8 знаков
Цифры от 1 до 9	1 млн комбинаций	100 млн комбинаций	Практически мгновенно	10 с
26 только прописных или только заглавных букв латинского алфавита	309 млн комбинаций	200 млрд комбинаций	30 с	менее 6 ч (в 720 раз дольше, чем для б знаков)
Смесь 52 прописных и заглавных буквы латинского алфавита	19 млрд комбинаций	53 трлн комбинаций	Полчаса	Два месяца (почти в 3000 раз дольше, чем для б знаков)
Прописные, заглавные буквы, цифры и все символы (точка, двоеточие и т. п.)	782 млрд комбинаций	7,2 квадриллионов комбинаций	22 ч	57 лет(примерно в 22700 раз дольше, чем для б знаков)

Brute-force (англ.) — решать что-либо «в лоб», методом грубой силы.

Данные взяты из статьи http://www.lockdown.co.uk/?pg= combi

Одна из проблем выбора пароля заключается в трудности его запоминания, а очень важно именно помнить его, а не записывать на листочек и прикреплять к монитору. Эта проблема осложняется тем, что в целях безопасности пароль надо достаточно регулярно менять. В качестве одного из возможных решений рекомендуют использовать следующий прием: выбрать какое-нибудь легко запоминающееся предложение, например «Куда идем мы с Пятачком большой- большой секрет» и преобразовать его в пароль, скажем, взяв первые буквы каждого слова: kimsPb-bs.

Серьезной проблемой использования многоразовых паролей является их ***ручная синхронизация.*** В обычной жизни нам требуется не один, а несколько паролей: для входа в компьютерную сеть офиса, в котором мы работаем, для доступа к «личному кабинету» провайдера мобильной связи, для доступа к банковскому счету и еще для доступа к самым разным другим интернет-сайтам. Одни из этих Web-сайтов являются очень важными для нас — компрометация пароля, используемого для доступа к ним, может означать для нас большие материальные потери (например, доступ к банковскому счету), другие — менее важные, а третьи — вообще используются единожды, мы регистрируемся и больше мы к этим сайтам не возвращаемся. Однако часто случается, что во всех этих случаях используется один и тот же пароль (возможно с небольшими вариациями), потому что у нас нет времени придумывать и, главное, запоминать новый пароль для доступа к новому ресурсу. Такое явление называют ручной синхронизацией паролей. Выполнив регистрацию на сайте, незаслуживающем доверия, вы сообщаете его владельцам свой пароль, который может быть использован ими для доступа к другим вашим учетным данным, имеющим для вас критическое значение. Отсюда следует простое, но важное правило: никогда не используйте один и тот же пароль для доступа к разным ресурсам и услугам Интернета.

Слабостью паролей является также процедура реакции на неправильно введенный пароль. На первый взгляд естественным приемом, направленным на противодействование подбору паролей, может показаться блокирование учетной записи, с которой было проведено некоторое количество (обычно трех) неудачных попыток входа. Однако в некоторых ситуациях такой подход дает злоумышленнику прекрасную возможность быстро заблокировать работу предприятия. Действительно, идентификаторы пользователей являются менее защищенной информацией, чем пароли, к тому же они часто легко угадываемы (ADMIN, STUDENT, IVANOV, Natasha и т. п.) и их легче подсмотреть, так как они выводятся на экран. Поэтому злоумышленник может легко подобрать имена, выполнить по три неудачных попытки аутентификации с каждого аккаунта, вызвать их блокировку и привести таким образом систему в недоступное состояние. Снятие блокировок с учетных записей может стать серьезной проблемой, если таких записей окажется очень много.

Наряду с паролями существует другой вариант использования аутентификаторов из класса «что-то, что знаю». Пользователю заранее

безопасным образом администратор сообщает некоторое правило, например правило преобразования последовательности чисел в другие символы. Во время процедуры аутентификации система выводит на экран случайную последовательность чисел. Пользователь в соответствии с известным только ему и системе правилом преобразует их в другую последовательность символов, которую вводит в качестве пароля. Поскольку система также «знает» правило преобразования, она может проверить правильность введенного пароля. То есть изначально в данном случае в качестве разделяемого секрета выступает правило преобразования.

Электронные аутентификаторы

Электронные аутентификаторы относятся к разряду «что-то, что имею». Наиболее популярными из них являются (рис. 5.1):

- идентификаторы iButton (information button — информационная таблетка);
- USB-ключи или USB-токены (USB — порт компьютера, token — опознавательный жетон).
- смарт-карты (smart card — интеллектуальная карта);
- радиочастотные идентификаторы (radio-frequency identification) или RFID-идентификаторы;

Первые два вида аутентификаторов — идентификаторы iButton и USB-ключи — являются контактными, для считывания содержащейся в них информации они должны быть установлены в считывающее устройство или USB-порт компьютера, радиочастотные идентификаторы не требуют физического контакта со считывателями, а смарт-карты могут быть как контактными, так и бесконтактными.

Идентификаторы iButton представляют собой устройства, заключенные в круглый металлический корпус диаметром примерно 2 см, часто размещаемый на пластиковом держателе, подобном всем нам

Рис. 5.1. Электронные устройства-аутентификаторы: *а* — смарт-карта; *б* — RFID- идентификаторы; *в* — USB-ключ; *г* — iButton

знакомому ключу от подъездных дверей. Для считывания аутентификационных данных корпус устройства приводится в соприкосновение со считывателем. Во всех моделях идентификаторов iButton имеется ПЗУ, содержащее 64-битовый код (48-битовый уникальный идентификационный номер устройства, 8-битовый код типа изделия и 8 контрольных битов), однопроводный порт и схема, реализующая логику управления. В некоторых моделях помимо указанных обязательных элементов могут также содержаться криптографический микропроцессор, различные виды памяти (энергонезависимая SRAM, однократно программируемая память), литиевая батарейка, часы-календарь реального времени, термодатчик и др. При соприкосновении идентификатор iButton передает считывателю уникальный номер, считыватель проверяет его и инициирует сеанс обмена данными с идентификатором по определенному для этих двух устройств протоколу.

USB-ключи относятся к классу контактных электронных аутентификаторов, они подключаются непосредственно к USB-порту компьютера, исключая необходимость использования дорогостоящих считывающих устройств. Конструктивно USB-ключи выполняются подобно переносным устройствам памяти memory stick. Каждый USB-ключ имеет уникальный номер, присвоенный производителем.

Радиочастотные идентификаторы (RFID) относятся к бесконтактным аутентификаторам. Они изготавливаются в виде пластикового брелока, не имеющего никаких внешних информационных выходов/входов, а также никаких дисплеев и других видимых средств отображения информации. Внутри пластикового корпуса находится интегральная схема, которая хранит и обрабатывает информацию, а также миниатюрная антенна, предназначенная для передачи и приема радиосигналов. Система радиочастотной аутентификации помимо RFID- идентификатора включает ***считыватель радиочастотных сигналов,*** который встраивается в электронные замки, вычислительные устройства и др. Обладатель RFID-идентификатора использует его как пропуск в помещения предприятия, поднося его к считывателю на стене перед дверью с электронным замком. Считыватель постоянно излучает радиочастотный сигнал, который при поднесении идентификатора на определенное расстояние (зависящее от типа устройства) принимается его антенной и передается в виде питания интегральной схеме. Эта энергия используется микросхемой для излучения аутентификационных данных в направлении считывателя. Этот же идентификатор может использоваться для доступа к принтеру и другим устройствам вычислительной системы. Кроме того, RFID-идентификатор позволяет службе безопасности отслеживать перемещение его владельца по всем помещениям, оснащенными соответствующими датчиками.

Смарт-карты могут быть как контактными, так и бесконтактными Контактные смарт-карты имеют на одной из сторон контактные площадки, через которые при соприкосновении с контактами считы- п.Ігеля происходит передача электропитания и аутентификационной информации. Считыватели имеют самое разнообразное конструктивное

выполнение, например они могут быть встроены в клавиатуру или п корпус компьютера. В бесконтактных смарт-картах предусматривается радиочастотный блок со встроенной антенной, проложенной по периметру карты. Антенна служит как для передачи аутентификационных данных, так и для извлечения энергии из электромагнитного поля, излучаемого считывателем.

Смарт-карта содержит процессор, ПЗУ, в которой хранится криптграфическая программа, ОЗУ, используемое как рабочая память, I также EEPROM (электрически стираемое программируемое ПЗУ), '■держащее изменяемые данные владельца карты. Аутентификация пл«(дельца карты осуществляется по уникальному серийному номеру карты, который присваивается ей на предприятии-изготовителе, .1 также персональным данным владельца, хранящимся в памяти в ышифрованном виде.

Смарт-карту следует отличать от *карты памяти* ***(memory card),*** которая, представляя собой магнитный носитель, способна только «ранить данные, но не имеет средств для их обработки. Для реа- линации некоторых схем аутентификации этого вполне достаточно.

Аппаратные аутентификаторы, имеющие одинаковое конструк-ивное выполнение, могут реализовывать разные алгоритмы аутен-ификации. Е1апример, смарт-карта может использоваться как гене-I1.1гор одноразовых паролей, как аутентификатор на основе закрытого ключа или в схеме аутентификации со словом-вызовом (все эти и дру- I иг методы аутентификации рассматриваются далее в этой главе).

Общим недостатком всех аппаратных аутентификаторов являет- I и ю, что они могут быть потеряны или, что значительно хуже, укра- дны. Любой человек, завладевший аутентификатором, теоретически получает в свое распоряжение все полномочия законного владельца.

Биометрические аутентификаторы

Биометрические аутентификаторы относятся к разряду «что-то, ним являюсь» и представляют собой анатомические и поведенческие особенности человека. В отличие от паролей, которые можно шбыть, или аппаратных ключей, которые можно потерять, биометрические характеристики всегда при аутентифицируемом (исключая ..... страшные случаи, когда для предъявления отпечатков пальцев человека подвергают насильственнной ампутации). Еще одним преимуществом аутентификаторов из разряда «что-то, чем являюсь» по

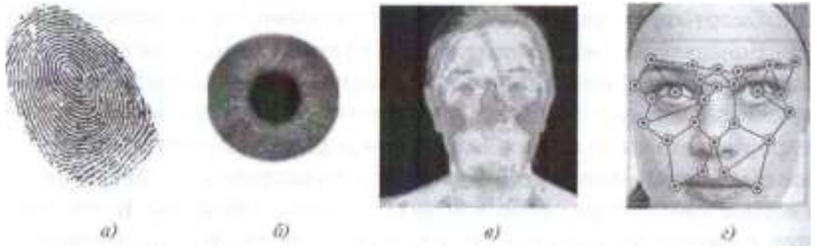


Рис. 5.2. Статические анатомические характеристики: о — отпечатки пальцев; 6— рисунок радужной оболочки глаза; *в* — термограмма лица; г — черты лица

сравнению с аутентификаторами, относящимися к классам «что-то, что знаю» и «что-то, что имею», является то, что их нельзя передать другому человеку. Хотя остается криминальная возможность предъявлять носителя биометрических характеристик против его воли для получения доступа к защищенным системам.

Наиболее часто для распознавания человека используются следующие *статические* анатомические характеристики:

- рисунки папиллярных сосудов на пальцах (отпечатки пальцев);
- овал лица, относительное расположение глаз, носа, рта. Для определения уникального шаблона, соответствующего определенному человеку, требуется от 12 до 40 характерных элементов. Шаблон должен учитывать множество вариаций изображения на случаи поворота лица, наклона, изменения освещённости, изменения выражения;
- термограмма лица;
- рисунок радужной оболочки или сетчатки глаза;
- геометрические параметры ладони, рисунок линий ладони.

К числу наиболее «говорящих» *динамическихх*арактеристик (поведенческих черт) человека относятся:

- характеристики речи (рис. 5.3);
- особенности письма вообще и личной подписи в частности;
- особенности походки;
- особенности набора текста на клавиатуре (скорость ввода парольной фразы, характерные ошибки, временные интервалы между нажатиями разных клавиш и др.).

Алгоритм биометрической аутентификации аналогично всем остальным способам аутентификации включает *процедуру регистрации.* во время которой в базу данных системы заносятся характеристики пользователей, необходимые для установления их аутентичности. Процедура регистрации выполняется с привлечением специальных устройств для взятия *биометрических образцов.* Эти устройства отражают специфику используемых для аутентификации анатомических

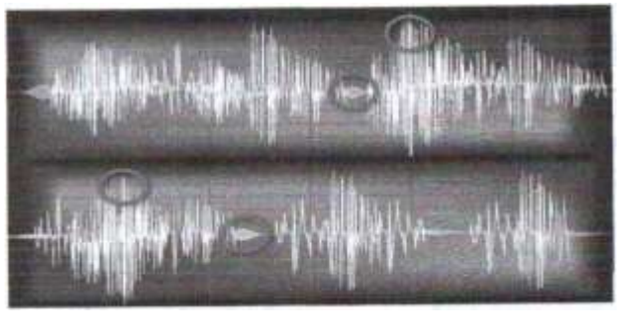


Рис. 5.3. Анализ особенностей речи с целью аутентификации

признаков: для снятия отпечатков пальцев используются различные виды сканеров, для получения изображений лица и радужной оболочки глаза — фотокамеры, для получения термограммы лица камеры инфракрасного диапазона, для взятия образцов голоса — звукозаписывающие устройства (рис. 5.4). Затем из биометрического образца с помощью специальной программы извлекают ***шаблон*** — данные, уникально характеризующие конкретного человека. Полученный шаблон, представляющий собой компактный код (в некоторых случаях до нескольких байтов), помещается в базу данных системы.

Собственно ***процедура аутентификации*** включает те же действия: взятие биометрического образца и извлечения шаблона — экстракта

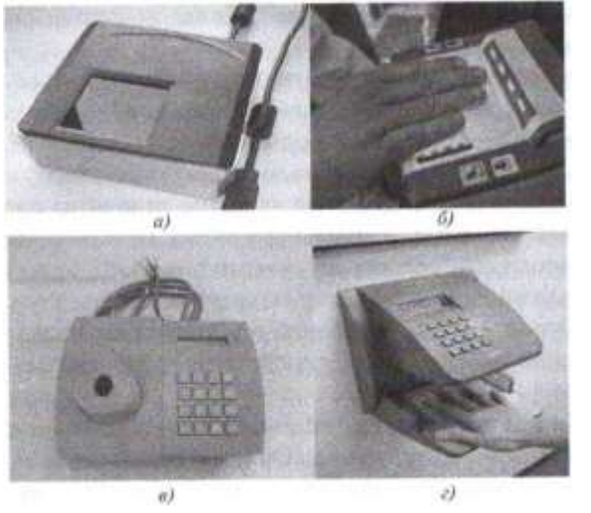


Рис. 5.4. Примеры конструктивного выполнения сканеров: о — прокатный сканер отпечатков одного пальца; *б*— оптический сканер отпечатков пальцев; *в* — сканер сетчатки глаза; *г* — устройство для снятия параметров ладони

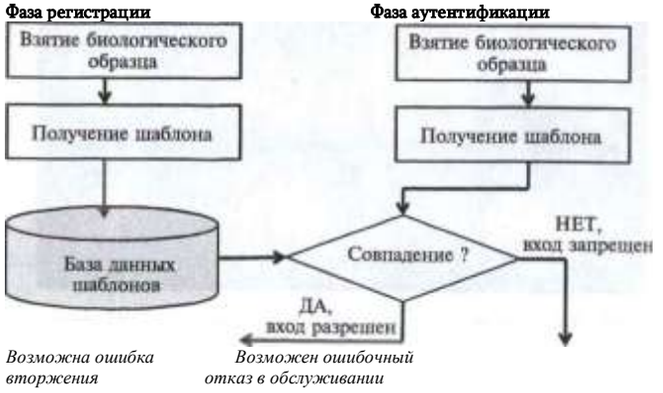


Рис. 5.5. **Алгоритм биометрической аутентификации**

индивидуальных черт объекта аутентификации. Актуальный шаблон сравнивается с шаблоном из базы данных системы. Если произошло совпадение, то аутентификация произошла успешно (рис. 5.5).

Остановимся на некоторых «подводных камнях» этой простой схемы, которые могут прояснить причины того, что биометрическая аутентификация все еще не получила того широкого распространения, которое казалось бы она заслуживает. В отличии от паролей и способов аутентификации, использующих электронные ключи, биометрическое распознавание основано не на точном совпадении, а на степени соответствия, поэтому здесь более вероятны ошибки, причем ошибки двух родов.

Во-первых, система может *отказать в обслуживании легальному пользователю,* если его отличительные биометрические черты изменились. А вероятность этого отнюдь не нулевая. Например, вы можете порезать палец и быть отлученным от доступа к своему компьютеру в течение недели, а то и двух. На качество отпечатка пальца может повлиять наличие крема, чернильного пятна, мозоли и проч. и проч. Если вы выполняете голосовую аутентификацию, то на нее может повлиять болезнь, внешняя температура, влажность, шум — музыка, работа транспорта и др. Особенно эти проблемы проявляются при аутентификации в мобильных устройствах — планшетах, телефонах, ноутбуках.

Во-вторых, система аутентификации может *позволить злоумышленнику осуществить вторжение.* Ошибки такого рода могут возникать как из-за того, что индивидуальные особенности аутентифицируемого были неправильно оценены системой как совпадающие с характеристиками другого человека, так и в результате атак, предпринятых на систему аутентификации. Помимо нарушений безопасности

15 См закон о защите персональных данных в Приложении 1.

Таблица 5.2

Достоинства и недостатки биометрических систем аутентификации

Достоинства	Недостатки
Пользователю удобно использовать аутентифи-	Вероятностный характер распозна- катор,
который всегда «с собой»	вания приводит к ложным сраба тываниям
Аутентификатор не может быть передан другому человеку	Биометрические параметры могут со временем изменятся по разным причинам: болезнь, возраст, изменение состояния внешней среды
Аутентифицируемый, пройдя аутентификацию, не сможет в дальнейшем отречься от своих действий Многие современные электронные устройства (компьютеры, телефоны и др) готовы для внедрения биометрической аутентификации, поскольку изначально оснащены средствами снятия биометрических образцов: фотокамерами, звукозаписывающими системами Биометрические параметры достаточно просто могут быть использованы в двухфакторной аутентификации, например в комбинации с паролем	Возможна подмена, насильственное изъятие и фабрикация биометрического материала Единожды скомпрометированный биометрический аутентификатор нельзя сменить на другой
	Сложности обеспечения безопасности персональных данных человека

универсального характера (инсайдеры, DoS, трояны, вирусы и др.) биометрические системы могут быть подвергнуты атакам, эксплуатирующим специфические уязвимости таких систем. В частности, существует угроза использования подделки в ходе взятия биометрического образца, например в сканер или другое устройство распознавания может быть представлены модель пальца или реальный мертвый палец, искусственное воссоздание лица, обманным путем полученные образцы голоса и др. Для противодействия этому виду атак разрабатываются различные методы распознавания неживого состояния объекта, у которого берутся пробы.

Другим видом атак на систему биометрической аутентификации являются несанкционированный доступ и копирование шаблонов из базы данных. Обладание шаблоном в некоторых случаях дает возможность злоумышленнику решить обратную задачу — генерации биометрического образца, который, будучи представлен во время аутентификации, даст значение шаблона, совпадающее с похищенным из базы данных.

Обеспечение безопасности биометрических аутентификаторов имеет еще один аспект — защита персональных данных. Закон строго регламентирует их использование15, поскольку украденные биометри

ческие данные могут использоваться для слежки за человеком или его компроментации, Поэтому в базах данных современных биометрических систем шаблоны создаются таким образом, чтобы из них нельзя было восстановить исходные биометрические характеристики, а следовательно, использовать в криминальных целях. То есть процедура получения шаблона из исходного биометрического образца должна работать подобно односторонней хеш-фунции — быстрое получение компактного уникального дайджеста (шаблона) и отсутствие возможности обратного вычисления по дайджесту исходного значения функции (биометрического образца).

В табл. 5.2 представлены некоторые достоинства и недостатки биометрических систем аутентификации.

Строгая аутентификация на основе многоразового пароля

Аутентификация пользователей сети средствами ОС

В современных операционных системах предусматриваются централизованные службы аутентификации. Такая служба поддерживается одним из серверов сети и использует для своей работы базу данных, в которой хранятся ***учетные данные*** о всех пользователях сети. Учетные данные содержат наряду с другой информацией идентификаторы и пароли пользователей. Упрощенно схема аутентификации в сети выглядит следующим образом. Когда пользователь осуществляет логический вход в сеть, он набирает на клавиатуре своего компьютера свои идентификатор и пароль. Эти данные используются службой аутентификации — в централизованной базе данных, хранящейся на сервере, по идентификатору пользователя находится соответствующая запись, из нее извлекается пароль и сравнивается с тем, который ввел пользователь. Если они совпадают, то аутентификация считается успешной, пользователь получает легальный статус и те права, которые определены для него системой авторизации.

Однако такая упрощенная схема имеет большой изъян. А именно — при передаче пароля с клиентского компьютера на сервер, выполняющий процедуру аутентификации, этот пароль может быть перехвачен злоумышленником. Поэтому в разных операционных системах применяются разные приемы, чтобы избежать передачи пароля по сети в незащищенном виде.

Аутентификация, в процессе которой используются методы шифрования, а аутентификатор не передается по сети, называется ***строгой аутентификацией.***

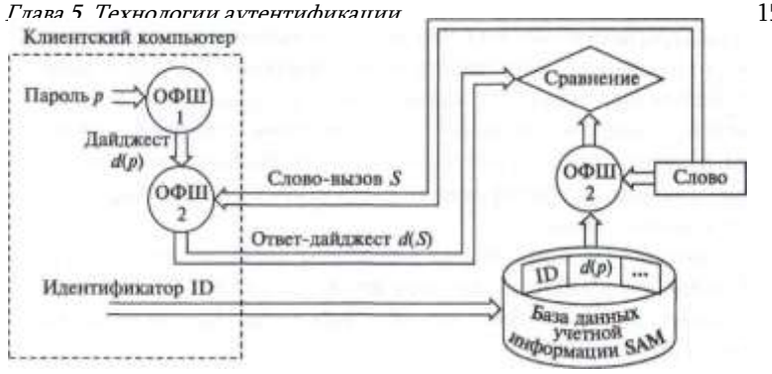


Рис. 5.6. Схема сетевой аутентификации на основе многоразового пароля

Рассмотрим пример строгой аутентификации пользователей в компьютерной сети.

Аутентификация пользователей сети выполняется на основе их паролей, хранящихся в зашифрованном виде в централизованной базе SAM (Security Accounts Manager) (рис. 5.6). Пароли зашифровываются с помощью односторонней функции шифрования при занесении их в базу данных во время процедуры создания учетной записи для нового пользователя. Введем обозначение для этой односторонней функции — ОФШ1. Таким образом, пароль ***P*** хранится в базе данных SAM в виде дайджеста ***d(P).*** (Напомним, что знание дайджеста не позволяет восстановить исходное сообщение.)

При логическом входе пользователь локально вводит в свой компьютер имя-идентификатор (Ю) и пароль ***P.*** Клиентская часть подсистемы аутентификации, получив эти данные, передает запрос по сети на сервер, хранящий базу SAM. В этом запросе в открытом виде содержится идентификатор пользователя ID, но пароль *не передается в сеть* ни в каком виде.

К паролю на клиентской станции применяется та же односторонняя функция ОФШ1, которая была использована при записи пароля в базу данных SAM, т. е. динамически вычисляется дайджест пароля ***d(P).***

В ответ на поступивший запрос серверная часть службы аутентификации генерирует случайное число ***S*** случайной длины, называемое ***словом-вызовом*** (challenge). Это слово передается по сети с сервера на клиентскую станцию пользователя. К слову-вызову на клиентской стороне применяется односторонняя функция шифрования ОФШ2. В отличие от функции ОФШ1, функция ОФШ2 является параметрической и получает в качестве параметра дайджест пароля ***d{P).*** Полученный в результате ответ ***d(S)*** передается по сети на сервер базы SAM.

Параллельно этому на сервере слово-вызов ***S*** аналогично шифруется с помощью той же односторонней функции ОФШ2 и дайджеста пароля пользователя ***d{P),*** извлеченного из базы SAM, а затем сравнивается с ответом, переданным клиентской станцией. При совпадении результатов считается, что аутентификация прошла успешно. Таким образом, аутентификация была выполнена без передачи пароля по каналам связи.

Заметим, что при каждом запросе на аутентификацию генерируется новое слово-вызов, так что перехват ответа ***d(S)*** клиентского компьютера не может быть использован в ходе другой процедуры аутентификации.

Аутентификация по протоколу CHAP

Протокол ***аутентификации по квитированию вызова*** (Challenge Handshake Authentication Protocol, CHAP) является одним из протоколов семейства протоколов PPP (Point-to-Point Protocol — протокол двухточечной связи). Протокол PPP предусматривает два режима аутентификации: аутентификация по протоколу PAP, когда пароль передается по линии связи в открытом виде, и аутентификация по протоколу CHAP, при которой пароль по линии связи не передается и, следовательно, обеспечивается более высокий уровень безопасности. В протоколе CHAP предусмотрено 4 типа сообщений: ***Success*** (успех), ***Challenge*** (вызов), ***Response*** (ответ), ***Failure*** (ошибка).

Этот протокол используется, например, при аутентификации удаленных пользователей, подключенных к Интернету по коммутируемому каналу. Здесь аутентификатором является сервер провайдера, а аутентифицируемым — клиентский компьютер (рис. 5.7). При заключении договора клиент получает от провайдера пароль (пусть, например, это будет слово parol). Этот пароль хранится в базе данных провайдера в виде дайджеста ***Z*** = rf(parol), полученного при применении к паролю односторонней хеш-функции MD5.

Аутентификация выполняется в следующей последовательности:

1. Пользователь-клиент активизирует программу (например, программу дозвона) удаленного доступа к серверу провайдера, вводя имя и назначенный ему пароль. Имя (на рисунке это «Moscow») передается по сети провайдеру в составе запроса на соединение, но пароль не передается в сеть ни в каком виде. То есть здесь мы имеем дело со строгой аутентификацией.

2. Сервер провайдера, получив запрос от клиента, генерирует псевдослучайное слово-вызов (пусть это будет слово «challenge») и передает его клиенту вместе со значением, идентифицирующем сообщение в рамках данного сеанса (ID), и собственным именем (здесь «Paris»). Это сообщение типа ***Challenge.*** (Для защиты от перехвата

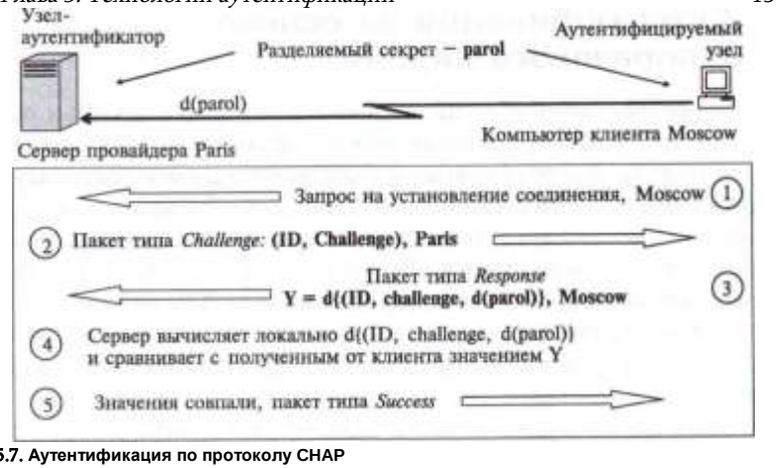


Рис. 5.7. Аутентификация по протоколу CHAP

ответа аутентификатор должен использовать разные значения слова-вызова при каждой процедуре аутентификации.)

3. Программа клиента, получив этот пакет, извлекает из него слово-вызов, добавляет к нему идентификатор и вычисленный локально дайджест ***Z =*** d(parol), а затем вычисляет с помощью все той же функции MD5 дайджест У = d{(ID, challenge, d(parol)} от всех этих трех значений. Результат клиент посылает серверу провайдера в пакете ***Response.***

4. Сервер провайдера сравнивает палученный по сети дайджест У с тем значением, которое он получил, локально применив ту же хеш-функцию к набору аналогичных компонентов, хранящихся в его памяти.

5. Если результаты совпадают, то аутентификация считается успешной и аутентификатор посылает партнеру пакет ***Success.***

Способ аутентификации, при котором многоразовые пароли пользователей хранятся в базе данных сервера в виде дайджестов, кажется вполне безопасным, ведь даже если злоумышленник сможет получить к ним доступ, он даже теоретически не сможет восстановить исходное значение паролей по дайджесту. Однако создатель первого червя Роберт Моррис решил эту проблему. Он разработал довольно простую программу, которая генерировала возможные варианты паролей, как используя слова из словаря, так и последовательным перебором символов. Для каждого сгенерированного слова вычислялся дайджест и сравнивался с дайджестами из файла паролей. Удивительно, но такая стратегия оказалась весьма эффективной, и хакеру удалось завладеть несколькими паролями.

Аутентификация на основе одноразового пароля

Алгоритмы аутентификации, основанные на многоразовых паролях, не очень надежны. Пароли можно подсмотреть, разгадать или просто украсть. Более надежными оказываются схемы с *одноразовыми паролями.* Одноразовые пароли также намного дешевле и проще биометрических систем аутентификации, таких как сканеры сетчатки глаза или отпечатков пальцев. Все это делает системы, основанные на одноразовых паролях, очень перспективными. Следует иметь в виду, что, как правило, системы аутентификации на основе одноразовых паролей рассчитаны на проверку удаленных, а не локальных пользователей.

Генерация одноразовых паролей может выполняться либо программно, либо аппаратно. Аппаратные реализации систем доступа на основе одноразовых паролей называют *аппаратными ключами.* Они представляют собой миниатюрные устройства со встроенным микропроцессором, похожие либо на обычные пластиковые карточки, используемые для доступа к банкоматам, либо на карманные калькуляторы, имеющие клавиатуру и маленькое дисплейное окно (рис. 5.8). Аппаратные ключи могут быть также реализованы в виде присоединяемого к разъему компьютера устройства.

Существуют и программные реализации средств аутентификации на основе одноразовых паролей — *программные ключи.* Программные ключи размещаются на сменном магнитном носителе в виде обычной программы, важной частью которой является генератор одноразовых паролей.

Независимо от того, какую реализацию системы аутентификации на основе одноразовых паролей выбирает пользователь, он, как и в системах аутентификации с применением многоразовых паролей, сообщает системе свой идентификатор, однако вместо того, чтобы

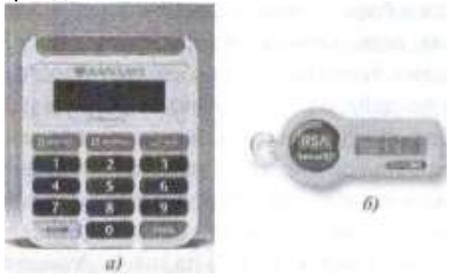


Рис. 5.8. Аппаратный ключ, который используют клиенты банка Barklays для доступа к своим счетам (о); аппаратный ключ SecurID (б)

вводить каждый раз один и тот же пароль, он указывает последовательность цифр, сообщаемую ему аппаратным или программным ключом. Через определенный небольшой период времени ключ генерирует другую последовательность — новый пароль. Сервер аутентификации проверяет введенную последовательность и разрешает пользователю осуществить логический вход. Сервер аутентификации может представлять собой отдельное устройство, выделенный компьютер или программу, выполняемую на обычном сервере.

Схема с использованием синхронизации

Рассмотрим схему использования аппаратных ключей, в основе которой лежит *синхронизация по времени.* Этот популярный алгоритм аутентификации был разработан компанией Security Dynamics.

Идея метода состоит в том, что аппаратный ключ и аутентифицирующий сервер вычисляют некоторое значение по одному и тому же алгоритму. Алгоритм имеет два параметра:

- ***разделяемый секретный ключ,*** представляющий собой 64-разрядное число, уникально назначаемое каждому пользователю и хранящееся как в аппаратном ключе, так и в базе данных сервера аутентификации;
- ***значение текущего времени.***

Если вычисленные значения совпадают, то аутентификация считается успешной.

Итак, пусть удаленный пользователь пытается совершить логический вход в систему с персонального компьютера (рис. 5.9). Аутентифицирующая программа предлагает ему ввести его личный ***персональный номер*** (PIN), состоящий из 4 десятичных цифр (на рисунке 2360), а также 6 цифр случайного числа, отображаемого в тот момент на дисплее ***аппаратного ключа*** (на рисунке — 112511). На основе PIN- кода сервер извлекает из базы данных информацию о пользователе, а именно — его секретный ключ. Затем сервер выполняет вычисления по тому же алгоритму, которой заложен в аппаратном ключе, используя в качестве параметров секретный ключ и значение текущего времени, проверяя, совпадает ли сгенерированное число с числом, которое ввел пользователь. Если они совпадают, то пользователю разрешается логический вход.

Потенциальной проблемой этой схемы является временная синхронизация сервера и аппаратного ключа. Ясно, что вопрос согласования часовых поясов решается просто. Гораздо сложнее обстоит дело с постепенным рассогласованием внутренних часов сервера и аппаратного ключа, тем более что потенциально аппаратный ключ может работать несколько лет. Компания Security Dynamics решает эту

Рис. 5.9. Аутентификация, основанная на временной синхронизации

проблему двумя способами. Во-первых, при производстве аппаратного ключа измеряется отклонение частоты его таймера от номинала. Далее эта величина учитывается в виде параметра алгоритма сервера. Во-вторых, сервер отслеживает коды, генерируемые конкретным аппаратным ключом, и если таймер данного ключа постоянно спешит или отстает, то сервер динамически подстраивается под него.

Существует еще одна проблема, связанная со схемой временной синхронизации. Одноразовый пароль, генерируемый аппаратным ключом, действителен в течение некоторого интервала времени (от нескольких десятков секунд до нескольких десятков минут), т. е. в течение этого времени одноразовый пароль в сущности является многоразовым. Поэтому теоретически возможно, что очень проворный хакер сможет перехватить PIN-код и одноразовый пароль с тем, чтобы также получить доступ в сеть в течение этого интервала.

Схема временной синхронизации не требует наличия компьютера на стороне аутентифицируемого, для этих целей можно ограничиться простым терминалом или факсом. Пользователи могут даже вводить свой пароль с телефонной клавиатуры, когда звонят в сеть для получения голосовой почты.

Схема с использованием слова-вызова

Другая схема применения аппаратных ключей, называемая часто ***«запрос-ответ»,*** основана на идее, очень сходной с идеей сетевой аутентификации, рассмотренной в предыдущем разделе. И в том и в другом случаях используется слово-вызов. Когда пользователь пытается осуществить логический вход, аутентификационный сервер передает ему запрос в виде случайного числа (рис. 5.10). Аппаратный ключ пользователя зашифровывает это случайное число, исполь

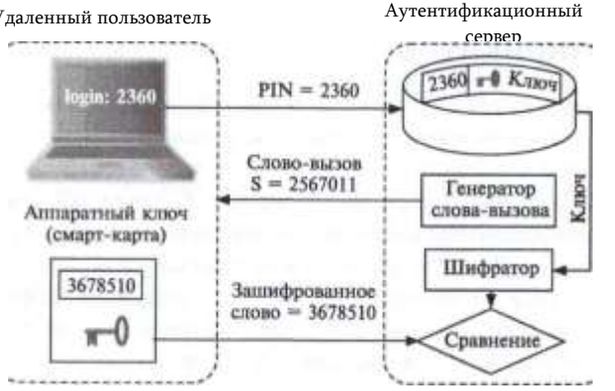


Рис. 5.10. Аутентификация по схеме «запрос-ответ»

зуя алгоритм DES и секретный ключ пользователя. Секретный ключ пользователя хранится в базе данных сервера и в памяти аппаратного ключа. В зашифрованном виде слово-вызов возвращается на сервер. Сервер в свою очередь также зашифровывает сгенерированное им самим случайное число с помощью алгоритма DES и того же секретного ключа пользователя, а затем сравнивает результат с числом, полученным от аппаратного ключа. Как и в методе временной синхронизации, в случае совпадения этих двух чисел пользователю разрешается вход в сеть.

Механизм с использованием слова-вызова имеет свои ограничения — он обычно требует наличия компьютера на каждом конце соединения, так как аппаратный ключ должен иметь возможность как получать, так и отправлять информацию. А схема временной синхронизации позволяет использовать простой терминал или факс. В этом случае пользователи могут даже вводить свой пароль с телефонной клавиатуры, когда они звонят в сеть для получения голосовой почты.

Схема «запрос-ответ» уступает схеме временной синхронизации по простоте использования. Для логического входа с использованием схемы временной синхронизации пользователю достаточно набрать 10 цифр. Схемы же «запрос-ответ» могут потребовать от пользователя выполнения большего числа ручных действий. В некоторых схемах «запрос-ответ» пользователь должен сам ввести секретный ключ, а затем набрать на клавиатуре компьютера полученное с помощью аппаратного ключа зашифрованное слово-вызов. В некоторых случаях пользователь должен вторично совершить логический вход в коммуникационный сервер после того, как осуществилась его аутентификация.

Аутентификация на основе сертификатов

Аутентификация с применением цифровых сертификатов является альтернативой применению паролей и представляется естественным решением в условиях, когда число пользователей сети (пусть и потенциальных) измеряется миллионами. В таких обстоятельствах процедура предварительной регистрации пользователей, связанная с назначением и хранением их паролей, становится крайне обременительной, опасной, а иногда и просто нереализуемой. При наличии сертификатов сеть, которая дает пользователю доступ к своим ресурсам, не хранит никакой информации о своих пользователях — они ее предоставляют сами в своих запросах в виде сертификатов, удостоверяющих личность пользователей. Сертификаты выдаются специальными уполномоченными организациями — ***центрами сертификации (Certificate Authority, CA***). Поэтому задача хранения секретной информации (закрытых ключей) возлагается на самих пользователей, что делает это решение гораздо более масштабируемым, чем вариант с централизованной базой паролей.

Схема использования сертификатов

Аутентификация личности на основе сертификатов происходит примерно так же, как на проходной большого предприятия. Вахтер пропускает людей на территорию на основании пропуска, который содержит фотографию и подпись сотрудника, удостоверенных печатью предприятия и подписью лица, выдавшего пропуск. Сертификат является аналогом пропуска и выдается по запросам специальными сертифицирующими центрами при выполнении определенных условий.

Сертификат представляет собой электронную форму, в которой содержится следующая информация:

- открытый ключ владельца данного сертификата;
- сведения о владельце сертификата, такие, например, как имя, адрес электронной почты, наименование организации, в которой он работает, и т. п.;
- наименование сертифицирующей организации, выдавшей данный сертификат;
- электронная подпись сертифицирующей организации, т. е. зашифрованные закрытым ключом этой организации данные, содержащиеся в сертификате.

Использование сертификатов основано на предположении, что сертифицирующих организаций немного и их открытые ключи широко доступны, например из публикаций в журналах.

Рис. 5.11. Формы представления цифрового сертификата

Сертификаты могут быть представлены в следующих трех формах (рис. 5.11):

- в открытой форме — сертификат содержит всю информацию в незашифрованном виде;
- в форме, состоящей из двух частей — открытой части, содержащей всю информацию в незашифрованном виде, а также закрытой части, представляющей собой открытую часть, зашифрованную закрытым ключом пользователя;
- в форме, состоящей из 3 частей — открытой части, открытой части, зашифрованной закрытым ключом пользователя, а также третьей части, представляющей собой первые две части, зашифрованные закрытым ключом сертифицирующей организации. Когда пользователь хочет подтвердить свою личность, он предъявляет свой сертификат в двух формах: открытой (т. е. такой, в которой он получил его в сертифицирующей организации) и зашифрованной с применением своего закрытого ключа (рис. 5.12). Сторона, проводящая аутентификацию, берет из незашифрованного сертификата открытый ключ пользователя и расшифровывает с его помощью зашифрованный сертификат. Совпадение результата с открытым сертификатом подтверждает, что предъявитель действительно является владельцем закрытого ключа, соответствующего указанному открытому.

Затем с помощью известного открытого ключа указанной в сертификате организации проводится расшифровка подписи этой организации в сертификате. Если в результате получается тот же сертификат с тем же именем пользователя и его открытым ключом, значит,

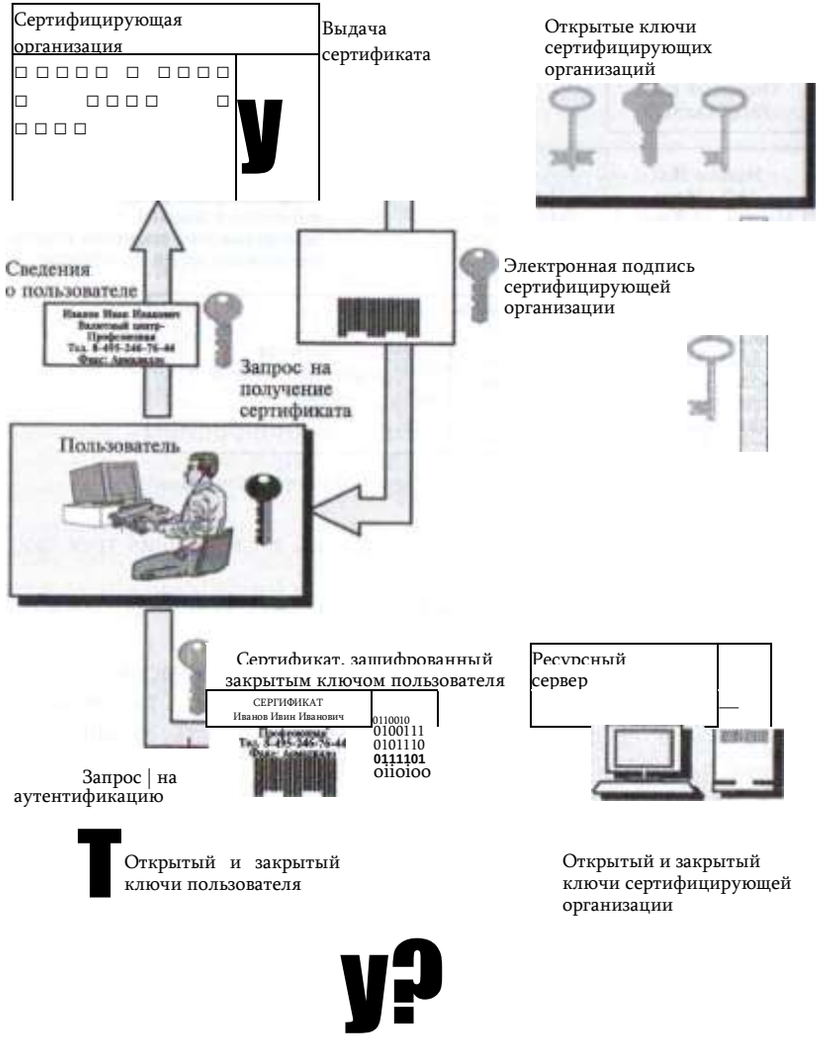


Рис. 5.12. Аутентификация пользователей на основе сертификатов

он действительно прошел регистрацию в сертификационном центре, является тем, за кого себя выдает, и указанный в сертификате открытый ключ действительно принадлежит ему.

Сертификаты можно использовать не только для аутентификации, но и для предоставления избирательных прав доступа. Для этого в сертификат могут вводиться дополнительные поля, в которых указывается принадлежность его владельцев к той или иной категории пользователей. Эта категория назначается сертифицирующей организацией в зависимости от условий, на которых выдается сертификат. Например, организация, поставляющая через Интернет на коммерческой основе информацию, может

выдавать сертификаты определенной

категории пользователям, оплатившим годовую подписку на некоторый бюллетень, тогда веб-сервер будет предоставлять доступ к страницам бюллетеня только пользователям, предъявившим сертификат данной категории.

Подчеркнем тесную связь открытых ключей с сертификатами. Сертификат является удостоверением не только личности, но и принадлежности открытого ключа. *Цифровой сертификат устанавливает и гарантирует соответствие между открытым ключом и его владельцем.* Это предотвращает угрозу подмены открытого ключа. Если некоторый абонент ***А*** получает по сети сертификат от абонента Б, то он может быть уверен, что открытый ключ, содержащийся в сертификате, гарантированно принадлежит абоненту Б, адрес и другие сведения о котором содержатся в этом сертификате. Это значит, что абонент ***А*** может без опасений использовать открытый ключ абонента Б для секретных посланий в адрес последнего.

При использовании сертификатов отпадает необходимость хранить на серверах корпораций списки пользователей с их паролями, вместо этого достаточно иметь на сервере список имен и открытых ключей сертифицирующих организаций. Может также понадобиться некоторый механизм отображений категорий владельцев сертификатов на традиционные группы пользователей для того, чтобы можно было в неизменном виде задействовать механизмы управления избирательным доступом большинства операционных систем или приложений.

Сертифицирующие центры

Сертификат является средством аутентификации пользователя при его обращении к сетевым ресурсам, роль аутентифицирующей стороны играют при этом информационные серверы корпоративной сети или Интернета. В то же время и сама процедура получения сертификата включает этап аутентификации, когда аутентификатором выступает сертифицирующая организация. Для получения сертификата клиент должен сообщить сертифицирующей организации свой открытый ключ и те или иные сведения, удостоверяющие его личность. Все эти данные клиент может отправить по электронной почте или принести на съемном носителе лично. Перечень необходимых данных зависит от типа получаемого сертификата. Сертифицирующая организация проверяет доказательства подлинности, помещает свою цифровую подпись в файл, содержащий открытый ключ, и посылает сертификат обратно, подтверждая факт принадлежности данного конкретного ключа конкретному лицу. После этого сертификат может быть встроен в любой запрос на использование информационных ресурсов сети.

Практически важным является вопрос о том, кто имеет право выполнять функции сертифицирующей организации. Во-первых, задачу обеспечения своих сотрудников сертификатами может взять на себя само предприятие. В этом случае упрощается процедура первичной аутентификации при выдаче сертификата. Предприятия достаточно осведомлены о своих сотрудниках, чтобы брать на себя задачу подтверждения их личности. Для автоматизации процесса генерации, выдачи и обслуживания сертификатов предприятия могут использовать готовые программные продукты, например компания Netscape Communications выпустила сервер сертификатов, который организации могут у себя устанавливать для выпуска своих сертификатов.

Во-вторых, эти функции могут выполнять независимые центры по выдаче сертификатов, работающие на коммерческой основе, например сертифицирующий центр компании Verisign. Сертификаты компании Verisign выполнены в соответствии с международным стандартом X.509 и используются во многих продуктах защиты данных, в том числе в популярном протоколе защищенного канала SSL. Любой желающий может обратиться с запросом на получение сертификата на веб-сервер этой компании. Сервер Verisign предлагает несколько типов сертификатов, отличающихся уровнем полномочий, которые получает владелец сертификата.

- *Сертификаты класса 1* предоставляют пользователю самый низкий уровень полномочий. Они могут применяться при отправке и получении шифрованной электронной почты через Интернет. Чтобы получить сертификат этого класса, пользователь должен сообщить серверу Verisign свой адрес электронной почты или свое уникальное имя.
- *Сертификаты класса 2* дают возможность его владельцу пользоваться внутрикорпоративной электронной почтой и принимать участие в подписных интерактивных службах. Чтобы получить сертификат этого более высокого уровня, пользователь должен организовать подтверждение своей личности сторонним лицом, например своим работодателем. Такой сертификат с информацией от работодателя может эффективно применяться при деловой переписке.
- *Сертификаты класса 3* предоставляют владельцу все те возможности, которые имеет обладатель сертификата класса 2, плюс возможность участия в электронных банковских операциях, электронных сделках по покупке товаров и некоторые другие возможности. Для доказательства своей аутентичности соискатель сертификата должен явиться лично и предоставить подтверждающие документы.
- *Сертификаты класса 4* используются при выполнении крупных финансовых операций. Поскольку такой сертификат наделяет владельца самым высоким уровнем доверия, сертифицирующий центр Verisign проводит тщательное изучение частного лица или организации, запрашивающей сертификат.

Механизм получения пользователем сертификата хорошо автоматизируется в сети в модели клиент-сервер, когда браузер исполняет роль клиента, а в сертифицирующей организации установлен специальный

сервер выдачи сертификатов. Браузер генерирует для пользователя пару ключей, оставляет закрытый ключ у себя и передает частично заполненную форму сертификата серверу. Для того чтобы неподписанный еще сертификат нельзя было подменить при передаче по сети, браузер ставит свою электронную подпись, зашифровывая сертификат выработанным закрытым ключом. Сервер сертификатов подписывает полученный сертификат, фиксирует его в своей базе данных и возвращает его каким-либо способом владельцу. Очевидно, что при этом может выполняться еще и неформальная процедура подтверждения пользователем своей личности и права на получение сертификата, требующая участия оператора сервера сертификатов. Это могут быть доказательства оплаты услуги, доказательства принадлежности к той или иной организации — все случаи жизни предусмотреть и автоматизировать нельзя. После получения сертификата браузер сохраняет его вместе с закрытым ключом и использует при аутентификации на тех серверах, которые поддерживают такой процесс. Примеры документов с цифровой подписью можно посмотреть в главе 21.

В настоящее время существует большое количество протоколов и продуктов, использующих сертификаты. В частности, практически все браузеры и операционные системы реализуют поддержку сертификатов.

Инфраструктура с открытыми ключами

Несмотря на активное использование технологии цифровых сертификатов во многих системах безопасности, эта технология еще не решила целый ряд серьезных проблем. Это, прежде всего, поддержание базы данных о выпущенных сертификатах. Сертификат выдается не навсегда, а на некоторый вполне определенный срок. По истечении срока годности сертификат должен либо обновляться, либо аннулироваться. Кроме того, необходимо предусмотреть возможность досрочного прекращения полномочий сертификата. Все заинтересованные участники информационного процесса должны быть вовремя оповещены о том, что некоторый сертификат уже недействителен. Для этого сертифицирующая организация должна оперативно поддерживать список аннулированных сертификатов.

Имеется также ряд проблем, связанных с тем, что сертифицирующие организации существуют не в единственном числе. Все они выпускают сертификаты, но даже если эти сертификаты соответствуют единому стандарту (сейчас это, как правило, стандарт X.509), все равно остаются нерешенными многие вопросы. Все ли сертифицирующие центры заслуживают доверия? Каким образом можно проверить полномочия того или иного сертифицирующего центра? Можно ли создать иерархию сертифицирующих центров, когда сертифицирующий центр, стоящий выше, мог бы сертифицировать центры, расположенные ниже в иерархии? Как организовать совместное использование сертификатов, выпущенных разными сертифицирующими организациями?

Для решения этих и многих других проблем, возникающих в системах,

использующих технологии шифрования с открытыми ключами, оказывается необходимым комплекс программных средств и методик, называемый *инфраструктурой с открытыми ключами* ***(Public Key Infrastructure, PKI)***. Информационные системы больших предприятий нуждаются в специальных средствах администрирования и управления цифровыми сертификатами, парами открытых/закрытых ключей, а также приложениями, функционирующими в среде с открытыми ключами,

В настоящее время любой пользователь имеет возможность, загрузив широко доступное программное обеспечение, абсолютно бесконтрольно сгенерировать себе пару открытый/закрытый ключ. Затем он может также совершенно независимо от администрации вести шифрованную переписку со своими внешними абонентами. Такая «свобода» пользователя часто не соответствует принятой на предприятии политике безопасности. Для более надежной защиты корпоративной информации желательно реализовать централизованную службу генерации и распределения ключей. Для администрации предприятия важно иметь возможность получить копии закрытых ключей каждого пользователя сети, чтобы в случае увольнения пользователя или потери пользователем его закрытого ключа сохранить доступ к зашифрованным данным этого пользователя. В противном случае резко ухудшается одна из трех характеристик безопасной системы — доступность данных.

Процедура, позволяющая получать копии закрытых ключей, называется *восстановлением ключей.* Вопрос, включать ли в продукты безопасности средства восстановления ключей, в последние годы приобрел политический оттенок. В США прошли бурные дебаты, тему которых можно примерно сформулировать так: обладает ли правительство правом доступа к любой частной информации при условии, что на это есть постановление суда?

И хотя в такой широкой постановке проблема восстановления ключей все еще не решена, необходимость включения средств восстановления в корпоративные продукты ни у кого сомнений не вызывает. Принцип доступности данных не должен нарушаться из-за волюнтаризма сотрудников, монопольно владеющих своими закрытыми ключами. Ключ может быть восстановлен при выполнении некоторых условий, которые должны быть четко определены в *политике* безопасности предприятия.

Как только принимается решение о включении в систему безопасности средств восстановления, возникает вопрос, как же быть с надежностью защиты данных, как убедить пользователя в том, что его закрытый ключ не употребляется с каким-либо другими целями, не имеющими отношения к резервированию? Некоторую уверенность в секретности хранения закрытых ключей может дать технология *депонирования ключей.* Депонирование ключей — это предоставление закрытых ключей на хранение третьей стороне, надежность которой не вызывает сомнений. Этой третьей стороной может быть правительственная организация или группа уполномоченных на это сотрудников предприятия, которым оказывается полное доверие.

Технология единого логического входа

Традиционный способ аутентификации с помощью многоразовых паролей отлично подходит для случая, когда пользователь все время работает с единственным компьютером, используя только его ресурсы и ресурсы Интернета, не требующие аутентификации. Такому пользователю нужно запоминать и периодически менять только один пароль. К сожалению, такая ситуация редко встречается в жизни. Более типичным является случай, когда пользователь должен использовать различные географически рассредоточенные компьютеры — рабочий, домашний, личный планшет, гостевой компьютер предприятия-партнера — и при этом получать доступ к различным серверам, например северам своего предприятия, серверам предприятия-партнера, а также к защищенным web-сайтам Интернета.

В том случае, когда каждый компьютер и каждый сервер требуют отдельной аутентификации с помощью многоразового пароля, пользователю приходится помнить и обновлять довольно много паролей, и с этой задаче многие пользователи справляются не очень успешно. Согласно исследованию, проведенному Network Applications

Рис. 5.13. Схема единого логического входа

Consortium, около 70 % звонков пользователей в службу ИТ-поддерж- ки связаны с просьбой восстановления забытого пароля, а в среднем пользователь крупной корпоративной сети тратит на процедуры логического входа 44 часа в год.

Не удивительно, что большие усилия затрачиваются на разработку процедур *единого логического входа* (*Single Sign On,* ***SSO).*** Целью процедуры единого логического входа является создание такого порядка аутентификации, при котором пользователь выполняет вход в сеть только один раз, доказывая свою аутентичность с помощью любого способа аутентификации, а затем результат этой аутентификации прозрачным для пользователя способом используется каждый раз, когда ему нужно доказывать свою аутентичность какому-либо серверу или приложению.

В настоящее время не существует программной системы аутентификации, реализующей концепцию единого логического входа, которая бы работала со всеми типами операционных систем, приложений и при этом учитывала бы разнообразные отношения между организациями, к которым принадлежат пользователи и информационные ресурсы. Однако имеются системы, которые позволяют организовать единый логический вход для однородной в каком-то отношении информационной системы, например для сети, использующей только одну определённую ОС или один определённый протокол аутентификации, либо для группы организаций, доверяющих друг другу при аутентификации своих пользователей.

Обобщённая схема, иллюстрирующая идею систем единого логического входа представлена на рис. 5.13. В этой схеме имеются три элемента:

- *Пользователь,* который располагает некоторой информацией, достаточной для его аутентификации. Это может быть информация любого типа из описанных выше — многоразовый пароль, одноразовый пароль, цифровой сертификат, биометрические данные и т. п. На рисунке в качестве примера показан вариант аутентификации на основе многоразового пароля.
- *Провайдер идентичности* (Identity Provider, IdP) — система, которая может аутентифицировать пользователя на основе базы данных учетных записей пользователей. Этот элемент может иметь и другие названия, например *сервер аутентификации.*

- *Сервис-провайдер* (Service Provider, SP) — система, предоставляющая сервисы пользователям. Такими сервисами могут быть файловый сервис, почтовый сервис, веб-сервис, сервис баз данных и т. п. Предполагается, что сервис предоставляется только аутентифицированным пользователям. Другим названием этого элемента является *ресурсный сервер.*

Особенностью этой схемы является то, что провайдер сервисов не поддерживает базу учётных данных пользователей. База учётных данных имеется только у провайдера идентичности, а провайдер сервисов доверяет результатам аутентификации пользователей, выполненной провайдером идентичности. Говорят, что в таком случае существуют *доверительные отношения* (*trust relationships)* между провайдером идентичности и провайдером сервисов.

Пользователь выполняет логический вход в сеть, обращаясь к провайдеру идентичности. Если пользователь смог подтвердить свою аутентичность, то провайдер идентичности выдает пользователю *токен доступа,* который пользователь хранит в своей базе данных.

При необходимости получения доступа к некоторому сервису пользователь предъявляет токен доступа ресурсному серверу. Токен доступа защищён криптографически таким образом, что ресурсный сервер имеет возможность проверить тот факт, что токен был выдан пользователю сервером аутентификации, которому ресурсный сервер доверяет аутентифицировать пользователей. Говорят, что в этом случае происходит *вторичная аутентификация* пользователя, но для самого пользователя она прозрачна, так как предъявлением токена доступа занимается программное обеспечение его компьютера.

Токен доступа обычно имеет ограниченное время действия, например сутки, поэтому пользователь должен его возобновлять, повторяя процедуру с сервером аутентификации.

Описанная схема допускает различные реализации в зависимости от применяемых технологий и протоколов аутентификации, типов операционных систем и сервисов, а также организационной принадлежности ее элементов.

В зависимости от применяемых криптографических технологий аутентификации системы единого логического входа делятся на два класса:

- *системы на основе разделяемого секрета:* многоразовых и одноразовых паролей, биометрических данных и другой информации, которая имеется как у пользователя, так и в базе учетных данных провайдера идентичности;
- *системы на основе технологии открытых и закрытых ключей,* использующей цифровые сертификаты и публичную (PKI) или корпоративную инфраструктуру верификации открытых ключей. В этом варианте провайдером идентичности является центр сертификации (Certification Authority, CA), выдавший цифровой сертификат пользователю, а сам сертификат представляет собой токен доступа. Сервис провайдер проверяет подлинность открытого ключа пользователя в сертификате с помощью иерархии публичных центров сертификации в случае использования PKI или же доверяя корпоративному центру сертификации.

В свою очередь системы на основе разделяемого секрета могут далее подразделяться в зависимости от применяемого протокола аутентификации. Наиболее популярными протоколами этого типа являются протоколы Kerberos16, RADIUS, TACACS.

В зависимости от типа сервиса, к которому пользователь получает доступ, системы единого логического входа делятся на:

- *системы входа на основе веб-сервисов.* Эти системы рассчитаны на широкий класс веб-сервисов, к которым доступ осуществляется с помощью веб-браузера. Эти системы используют специфику веб-сервисов — протокол HTTPS, языки XML, SAML. Примером SSO этого типа является система *Shibboleth,* разработанная сообществом Internet2;
- *системы входа на основе корпоративных сервисов.* Здесь под корпоративными сервисами понимаются все сервисы, не использующие веб-браузер в качестве пользовательского интерфейса, например сервисы мейнфреймов, баз данных и других корпоративных приложений. В частности к этому типу систем единого логического входа относится система Kerberos.

В зависимости от организационной принадлежности системы единого логического входа делятся на корпоративные и федеративные:

- в *корпоративной системе* все элементы — пользователи, провайдеры идентичности и сервис провайдеры — принадлежат одной организации, возможно — разным ее структурным отделениям;
- *федеративную систему* составляют различные организации, которые договорились доверять друг другу в отношении аутентификации своих пользователей. В общем случае каждая организация выполняет функции как провайдера идентичности, так и сервис провайдера. Каждая организация поддерживает базу учетных записей своих пользователей, которая не реплицируется в другие организации. При необходимости доступа пользователя одной организации к сервису другой организации выполняется вторичная аутентификация с помощью определенного протокола, например RADIUS или Shibboleth. Федеративная аутентификация достаточно популярна в академической среде, примером федерации такого типа является федерация Eduroam, членами которой являются многие университеты и исследовательские центры Европы и Америки. В Eduroam используется иерархия серверов RADIUS, а доступ ограничивается только доступом пользователей к Интернет через беспроводные сети.

16 Подробно о протоколе и системе Kerberos читайте в главе 16.

Однородность операционных систем, используемых в системах единого логического входа, упрощает их организацию. Поэтому не удивительно, что существуют продукты, позволяющие построить систему единого логического входа только в среде одной операционной системы. К таким продуктам относятся распределенные справочные службы Novell NDS и Microsoft Active Directory. Они позволяют достаточно просто строить крупные иерархические системы единого логического входа, но только в том случае, когда все операционные системы являются системами Novell Netware (этот пример носит исторический характер) или Microsoft Windows. Наличие единых стандартов в области аутентификации создает хорошую базу и для неоднородных систем единого логического входа, но успехи здесь пока скромнее.

Существуют программные системы, которые иногда также относят к системам единого логического входа, хотя некоторые специалисты считают такую классификацию спорной. Это системы, которые работают в сетях с традиционной инфраструктурой большого количества локальных баз учетных записей, т. е. в сетях, где каждый компьютер имеет свою базу идентификаторов и паролей пользователей. Такие программные системы помогают пользователю оперировать с большим количеством различных паролей, а иногда и работают от имени пользователя, выполняя интерактивный вход в удаленный сервер и делая процедуру аутентификации прозрачной. Существует три типа таких систем:

- системы кэширования паролей на стороне клиента:
- системы кэширования паролей на стороне сервера;
- системы синхронизации паролей между серверами.

Система кэширования паролей на стороне клиента хранит в надежном зашифрованном виде все пароли, которые пользователь использовал при входе в тот или иной сервер и может при повторном обращении к этому серверу выполнить логический вход от имени пользователя. Вы, конечно, сталкивались с такой функцией, встроенной во многие браузеры, когда браузер предлагает вам запомнить пароль. Считается, что кэширование паролей на стороне клиента является весьма опасной практикой, так как злоумышленник в случае получения доступа к вашему компьютеру, сразу получает доступ к всем серверам и сервисам, которыми вы пользуетесь, в том числе и вашему банковскому счету.

Кэширование на стороне сервера означает, что все пароли пользователя хранятся централизованно, например на выделенном сервере предприятия. При логическом входе пользователя в корпоративную сеть кэш паролей временно загружается в его клиентский компьютер, а после окончания сессии работы с ним кэш уничтожается. Считается, что это более защищенный способ по сравнению с постоянным хранением кэша паролей на клиентском компьютере.

Программы синхронизации паролей между серверами помогают уменьшить разнообразие паролей пользователей в корпоративной сети и свести их в крайнем случае к одному, действительному для всех серверов сети.

Все три разновидности программ манипулирования паролями не ликвидируют главную причину необходимости выполнения многочисленных логических входов в различные системы — наличие большого количества локальных баз учетной информации пользователей. Из-за этого обстоятельства такие системы и не относят к системам единого логического входа специалисты-пуристы.

Аутентификация информации. Электронная подпись

Аутентификация данных включает:

- подтверждение ***целостности*** хранящихся и переданных по сети данных и программ, т. е. установление факта того, что они не подвергались модификации17;
- доказательство ***авторства*** сообщения (документа, программы), в том числе и для недопущения отказа от авторства;
- доказательство ***легальности*** приобретения программного обеспечения.

Все эти задачи в той или иной мере могут быть решены с использованием электронной подписи.

Электронная подпись

Согласно терминологии, утвержденной Международной организацией по стандартизации (ISO), под термином ***«электронная (цифровая) подпись»*** понимаются методы, позволяющие устанавливать подлинность автора сообщения (документа) при возникновении спора относительно авторства. Основная область применения цифровой подписи — это финансовые документы, сопровождающие электронные сделки, документы, фиксирующие международные договоренности, и т. п. Подчеркнем, что электронная подпись не ставит задачи обеспечения конфиденциальности сообщений.

Хотя для получения подписи могут использоваться и симметричные алгоритмы, более распространенными являются алгоритмы на основе открытого и закрытого ключей. Как уже отмечалось (см. раздел «Алгоритм

17 См. рис. 4.1; в предыдущем разделе, посвященном криптографии, был рассмотрен метод подтверждения целостности с использованием односторонних функций шифрования.

RSA»), в основе этого алгоритма лежит концепция Диффи-Хеллмана. Она заключается в том, что каждый пользователь сети имеет свой закрытый ключ, необходимый для формирования подписи, а соответствующий этому секретному ключу открытый ключ, предназначенный для проверки подписи, известен всем другим пользователям сети.

ПРИМЕР-АНАЛОГИЯ

Вернемся к гипотетическому примеру организации, в которой для *безопасного обмена* данными в качестве открытых и закрытых ключей использовались русско- санскритские и санскритско-русские словари. Сейчас мы используем этот же пример (рис. 5.14) для пояснения концепции использования открытого и закрытого ключей с целью *подтверждения авторства* (аутентификации) посылаемого сообщения. Пусть подтверждение авторства требуется для посланий руководителя своим сотрудникам. Для этого руководитель обзаводится единственным экземпляром русско-санскритского словаря (он играет роль закрытого ключа *D*). который он хранит в защищенном от всех остальных месте, и множеством санскритско-русских словарей (открытых ключей *E*). которые раздаются всем сотрудником и лежат в общедоступных местах. Теперь те письма, аутентичность которых должна быть подтверждена, руководитель пишет на санскрите (т. е. шифрует их закрытым ключом *D*). Сотрудник, получивший послание, пытается перевести зашифрованный текст письма, пользуясь санскритско-русским словарем (открытым ключом *E*). Если ему это удается, то это доказывает, что текст был зашифрован закрытым ключом, парным открытому ключу *E* руководителя, и значит, именно он является автором этого сообщения.

Заметим, что в этом случае сообщения руководителя S_{12}. S13, S14. адресованные пользователям 2, 3 и 4, не являются секретными, так как все адресаты — обладают одним и тем же открытым ключом, с помощью которого они могут расшифровывать все сообщения, поступающие от пользователя 1, но задача обеспечения секретности в данном случае и не ставилась.

На рис. 5.15 показана схема формирования цифровой подписи по алгоритму RSA. Прежде всего отправитель должен сгенерировать пару ключей (***D,n***) и ***(E, п)*** и отправить получателю открытый ключ (***E,n***). Затем он должен сформировать сообщение. Подписанное сообщение состоит из двух частей: незашифрованной части, в которой

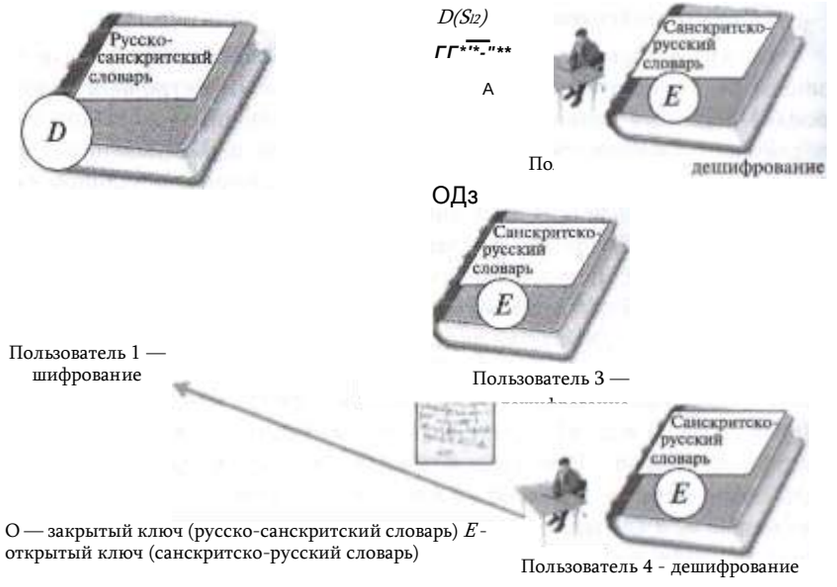


Рис. 5.14. Использование шифрования закрытым ключом для подтверждения авторства

содержится исходный текст ***T,*** и зашифрованной части, представляющей собой цифровую подпись. Цифровая подпись ***S*** вычисляется с использованием закрытого ключа (***D***, ***п)*** по формуле $S = T^D \bmod n$.

Рис. 5.15. Схема формирования цифровой подписи по алгоритму RSA

Сообщение посылается в виде пары (Г, ***S).*** Каждый пользователь, имеющий соответствующий открытый ключ (Я, гг), получив сообщение, отделяет открытую часть Т, расшифровывает цифровую подпись ***S*** и проверяет равенство $T = S^E \bmod n$.

Если результат расшифровки цифровой подписи совпадает с открытой частью сообщения, считается, что документ подлинный, не претерпел никаких изменений в процессе передачи, а автором его является именно тот человек, который передал свой открытый ключ получателю. Если сообщение снабжено цифровой подписью, то получатель может быть уверен, что оно не было изменено или подделано по пути.

К недостаткам данного алгоритма можно отнести то, что длина

Рис. 5.16. Обеспечение конфиденциальности документа с цифровой подписью

подписи в этом случае равна длине сообщения, что не всегда удобно. Для уменьшения «длины» электронной подписи вместо $S = T^{L>} \bmod n$ используются формулу $S = (H(T))^D \bmod n$, где ***H{T)*** — хеш-функция, отображающая исходное сообщение в короткий дайджест. В этом случае получатель сообщения (Г, ***S***) должен сначала применить к открытому тексту ***T*** хеш-функцию Я и получить дайджест Я(Т). А затем приступить к расшифровке подписи ***S*** откры- |ым ключом. Если расшифрованная подпись совпадает с дайджестом, то авторство сообщения доказано. Использование хеш-функций дает выигрыш не только в объеме сообщения, но и во времени получения электронной подписи.

Электронную подпись называют ***квалифицированной,*** если открытый ключ передается адресату в виде цифрового сертификата, надежно связывающего отправителя с его открытым ключом. Отсут- I гвие такой связи такой связи делает возможной подделку сообщений. Например, злоумышленник может, выдав себя за некоего легального автора документов, навязать получателю свой собственный открытый ключ, а затем посылать документы, снабженные электронной подпи- | |»ю на основе своего же закрытого ключа. Получатель выполнит проверку, убедится в «аутентичности» подметного письма и решит, что <>но написано легальным автором. Цифровой сертификат исключает мкую ситуацию, так как в нем имеется верифицированная информация о том, кто является владельцем присланного открытого ключа.

Если помимо проверки аутентичности документа, обеспечиваемой цифровой подписью, надо обеспечить его ***конфиденциальность,*** то после применения к тексту цифровой подписи перед передачей его по каналу связи выполняют совместное шифрование исходного текста и цифровой подписи (рис. 5.16).

Аутентификация программных кодов

Электронная подпись может быть использована и для доказательства аутентичности (подлинности) программ. Пользователю важно быть уверенным, что программа, которую он загрузил с какого-либо сервера Интернета, действительно содержит коды, разработанные определенной компанией. Протоколы защищенного канала (см. далее в главе 7) типа SSL помочь здесь не могут, так как позволяют удостоверить только аутентичность сервера. Компания Microsoft предложила использовать для этих целей технологию *аутентикода* (***authenticode***).

Организация, желающая подтвердить свое авторство на программу, должна встроить в распространяемый код так называемый подписывающий блок — аутентикод (рис. 5.17). Этот блок состоит из двух частей. Первая часть — сертификат организации-разработчика данной программы, полученный обычным образом от какого-либо сертифицирующего центра. Вторую часть образует зашифрованный дайджест, полученный в результате применения хеш-функции к распространяемому коду. Шифрование дайджеста выполняется с помощью закрытого ключа организации.

Компания-разработчик может потребовать от пользователя программы доказательство легальности ее приобретения. Для этого компания может запросить регистрационный номер программы (Product ID или Serial Number), называемый также ***лицензионным ключом активации.*** Обычно этот номер пишется на отдельном бланке, прила-

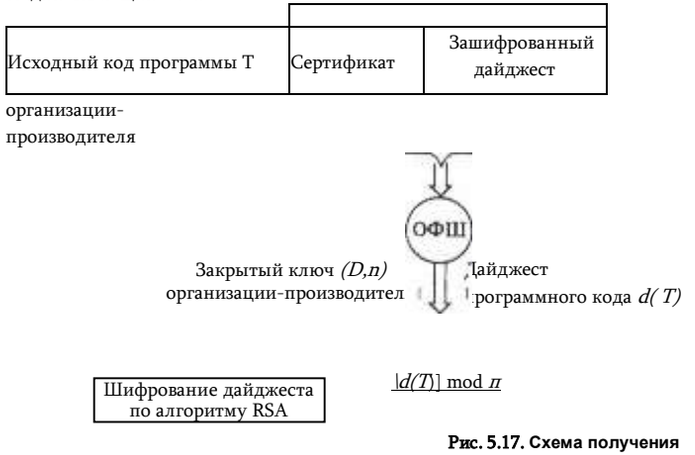


Рис. 5.17. Схема получения аутентикода

юемом к поставляемой программе, или наносится на упаковку, или высылается по электронной почте при покупке программы через Ин- юрнет.

Другим способом доказательства легальности приобретения и законности использования программных продуктов являются миниатюрные электронные устройства — *электронные замки,* подобные уже рассмотренным нами аппаратным аутентификаторам. Эти устройства поставляются вместе с защищаемыми от нелегального использования программами. Перед запуском программы электронный замок должен быть подключен к компьютеру, например через USB-порт. Инициирующий блок программы обращается к данному устройству с запросом и, получив «правильный» ответ, начинает работать, если же ожидаемый ответ не поступает, то выполнение программы блокируется. Таким образом электронный замок действует как специфический аутентификатор пользователя, доказывающий то, что он является законным владельцем программы.

Вопросы к главе 5

1. В чем состоит ручная синхронизация паролей:

а) администратор синхронизирует значения баз данных паролей на разных серверах сети;

б) ручная процедура приведения всех паролей одного пользователя к единому значению;

в) использование одного и того же пароля для доступа к разным по степени защищенности сервисам.

2. Аутентификаторы из разряда «что-то знаю»:

а) многоразовый пароль;

б) одноразовый пароль, сгенерированный устройством;

в) преобразование в соответствии с некоторым правилом символов, выводимых системой на экран, и использование их в качестве пароля;

г) информация о режиме работы учреждения;

д) информация о месте и времени встречи.

3. Отметьте слабости аутентификации с использованием многоразовых паролей:

а) возможность разгадывания пароля, если он слишком короткий;

б) необходимость передачи пароля по сети в открытом виде;

в) возможность подслушивания, подглядывания, «выманивания» пароля;

г) ручная синхронизация;

д) трудность запоминания надежных паролей;

е) сложность реализации.

4. Какие поведенческие характеристики человека используются для аутентификации:

а) характеристики речи;

б) особенности проведения свободного времени, специфические интересы;

в) особенности письма;

г) особенности личной подписи;

д) особенности походки;

е) особенности набора текста на клавиатуре.

5. Какие особенности биометрической аутентификации являются ее достоинствами:

а) вероятностный характер распознавания;

б) биометрический аутентификатор всегда «с собой»;

в) биометрический аутентификатор нельзя передать другому;

г) биометрический аутентификатор может изменяться со временем;

д) биометрический аутентификатор может использоваться помимо аутентификации и в других, самых разных целях.

б- При аутентификации пользователей сети шифрование пароля пользователя может выполняться:

а) для безопасного локального хранения пароля на клиентском компьютере;

б) для безопасного хранения паролей пользователей на сервере;

в) для безопасной передачи пароля по сети;

г) для обеспечения целостности пароля.

7. Какие из следующих утверждений о системах аутентификации ошибочны?

а) схемы, основанные на одноразовых паролях, дешевле биометрических;

б) схемы, основанные на одноразовых паролях, в основном используются для аутентифкации удаленных пользователей;

в) генерация одноразовых паролей выполняется только аппаратными устройствами;

г) после нескольких неуспешных попыток пользователя ввести пароль надо всегда блокировать его аккаунт.

8. Что из перечисленного содержится в сертификате?

а) информация о владельце сертификата;

б) открытый ключ владельца сертификата;

в) закрытый ключ владельца сертификата;

г) открытый ключ сертифицирующей организации;

д) закрытый ключ сертифицирующей организации;

е) информация о сертифицирующем центре, выпустившим данный сертификат.

9. Вставьте пропущенные слова в следующее предложение: «Сертификат может быть представлен в форме, состоящей из 3 частей: во-первых, ... части; во- вторых, ... части, зашифрованной ... ключом пользователя, в-третьих, части, представляющей собой первые две части, зашифрованные ... ключом сертифицирующей организации».

а) открытый;

б) закрытый;

10. Основная идея процедуры единого логического входа состоит в том, что:

а) пользователь выполняет логический вход в сеть только один раз при поступлении на работу, все остальное время система выполняет только его авторизацию;

б) все пользователи выполняют процедуру логического входа с одного и того же компьютера;

в) пользователь выполняет логический вход в сеть только один раз, затем результат этой аутентификации используется другими серверами или приложениями.

11. Какие из ниже перечисленных определений могут быть применимы к системам единого логического входа:

а) система на основе технологии открытых и закрытых ключей;

б) система входа на основе веб-сервисов;

в) система на основе почтового сервиса;

г) локальная система входа;

д) федеративная система.

12. Цели использования электронной подписи:

а) доказательство целостности сообщения;

б) обеспечение конфиденциальности сообщения;

в) доказательство авторства сообщения;

г) обеспечение неотказуемости.

13. Если T — исходный текст, а (D, п) и $(E,п)$ — закрытый и открытый ключи отправителя соответственно, то:

а) цифровая подпись S отправителя вычисляется по формуле $S = T^D$ modn;

б) получатель расшифровывает цифровую подпись S, выполняя вычисления по формуле $T = S^B$modn;

в) цифровая подпись S отправителя вычисляется по формуле $T = S^E$ modn;

г) получатель расшифровывает цифровую подпись S. выполняя вычисления по формуле $S = T^D$ mod п;

д) цифровая подпись S отправителя вычисляется по формуле $S = TE°$ modn;

е) получатель расшифровывает цифровую подпись S. выполняя вычисления по формуле $S = TD^E$ mod n.

14. С какой целью в электронной подписи используется хеш-функция?

а) для дополнительной гарантии целостности;

б) для уменьшения времени получения электронной подписи;

в) для обеспечения конфиденциальности;

г) для уменьшения длины сообщения.

15. Для доказательства легальности приобретения программы может использоваться:

а) регистрационный номер программы;

б) электронный замок;

в) сертификат организации-разработчика.

16. Как убедиться в подлинности сертификата?

а) его надо расшифровать с помощью открытого ключа владельца сертификата и сравнить с открытой информацией сообщения;

б) его надо расшифровать с помощью закрытого ключа владельца сертификата и сравнить с открытой информацией сообщения;

в) его надо расшифровать с помощью открытого ключа сертифицирующей организации;

г) его надо послать для проверки в сертифицирующий центр.

17. Нужна ли клавиатура набора цифр на карточке — аппаратном ключе, применяемом при аутентификации на основе одноразовых паролей по схеме «запрос-ответ»? А по схеме, использующей синхронизацию по времени?

18. Опишите процедуру получения сертификата. На основании каких сведений о пользователе выдается сертификат?

19. В чем заключается масштабируемость метода аутентификации на основе сертификатов?

6 ТЕХНОЛОГИИ АВТОРИЗАЦИИ И УПРАВЛЕНИЯ ДОСТУПОМ

Основная цель системы безопасности информационной системы — обеспечивать защиту данных от несанкционированного просмотра, изменения, уничтожения. Поставленная цель достигается самыми разными средствами, среди которых важная роль принадлежит аутентификации и авторизации. После того как пользователь, пройдя аутентификацию, доказал свою легальность, ему предоставляется некоторый набор прав по отношению к защищаемым системой ресурсам.

Наделение легальных пользователей правами доступа к ресурсам — называется *авторизацией.*

Процедура приведения авторизации в действие называется *управлением доступом* (Access Control).

Управление доступом реализуется применением различных *механизмов управления доступом.* Если, например, субъект пытается использовать ресурс с запрещенным для него типом доступа, то механизм управления доступом должен отклонить эту попытку и, возможно, уведомить систему об этом инциденте с целью генерации сигнала тревоги.

При решении задачи управления доступом необходимо руководствоваться *принципом минимальных привилегий.* В соответствии с ним каждому субъекту в системе должен быть назначен минимально возможный набор прав, достаточный для выполнения ровно тех задач, на которые он уполномочен. Применение этого принципа ограничивает те возможные потери, которые могли бы быть нанесены в результате неумышленных ошибок или неавторизованных действий.

Формы представления ограничений доступа

Решение о наделении пользователей правами (или что одно и то же — об ограничении их прав по доступу к ресурсам) основывается на политике безопасности предприятия и может формулироваться в разных формах.

Правила

Ограничение доступа может задаваться в форме *правил.* На основании правила система управления доступом в любой момент времени динамически решает вопрос о предоставлении или непредостав- лении доступа. Правило может строится с учетом различных факторов, в том числе длительности сеанса связи (ограничение доступа по времени использования

ресурса), возраста человека (ограничение для детей на доступ к некоторым сайтам), времени суток (разрешение использования ресурсов и сервисов Интернета только в рабочие часы). Популярной мерой ограничения доступа в Интернет является *капча* (***capcher***) — субъекту, обратившемуся с запросом к ресурсу, предлагается ввести символы, выведенные на экран в таком искаженном виде, в котором их сможет распознать только человек; таким образом исключается доступ к ресурсам искусственных субъектов — программных систем. Другим широко используемым правилом является правило, которое носит специальное название — *«необходимо знать»* («need- to-know»). В соответствии с этим правилом каждый сотрудник имеет право доступа только к тем информационным ресурсам, которые ему необходимы для выполнения его служебных обязанностей.

Для ограничения доступа используются также *контентно-* и *контекстно-зависимые правила.* Например, в компании может быть принято правило, что некоторым категориям пользователей запрещается доступ к документам, содержащим те или иные ключевые слова или фразы, такие как «для ограниченного использования», «секретно» или кодовое название проекта. Ограничения могут быть наложены на доступ к ресурсам, содержащим текст на иностранном языке. Это были примеры контентно-зависимых правил. Контекстно-зависимые правила принимают во внимание некоторые факторы, характеризующие текущее состояние среды и/или предысторию (контекст) запроса. Простейшим правилом такого рода является отказ в доступе пользователю, который сделал подряд три безуспешных попытки аутентификации. Или доступ к некоторому сетевому ресурсу предприятия может быть запрещен, если к моменту текущего обращения пользователь выполнил несколько обращений к внешнему сайту, содержимое которого не связано прямо с его профессиональными интересами.

Эффективным средством ограничения доступа является *конфигурирование пользовательского интерфейса.* Таким путем пользователь может быть лишен возможности не только обращаться к тем или иным каталогам и файлам, но и возможности видеть на своем экране часть структуры файловой системы, доступ к которой ему запрещен. Администратор может настроить систему меню пользовательского интерфейса так, что некоторые пункты этих меню не будут

выводится на экран, что исключит принципиальную возможность запуска пользователем некоторых функций.

Матрица прав доступа

Матрица прав доступа — это наиболее общий способ задания разрешений, которые получают пользователи в ходе авторизации. Многие правила и даже конфигурационные ограничения могут быть представлены в виде матрицы.

Матрица прав доступа является универсальной и наиболее гранулированной формой представления политики контроля доступа, она прямо, «в лоб» описывает для каждого пользователя набор ***конкретных*** операций, которые ему разрешается выполнять по отношению к ***каждому*** объекту (табл. 6.1).

Обозначим элемент матрицы ту, где ***i*** — номер строки, а ***j*** — номер столбца, Каждый столбец матрицы соответствует одному субъекту (пользователю), а строка — объекту (ресурсу). Каждый элемент матрицы ***rriij*** описывает, какие операции по отношению к ***i***-му ресурсу может выполнять ***j***-й пользователь. В нашем примере в матрице отображены взаимоотношения 158 субъектов и 2503 объектов.

Матричный способ задания прав доступа теоретически дает возможность отразить все многообразие отношений субъектов и объектов системы для всех возможных сочетаний (субъект, объект, назначенные права}. Однако этот универсальный способ представления, как правило, очень сложно реализовать на практике из-за его громоздкости, учитывая огромное число элементов — как субъектов, так и объектов, присутствующих в вычислительной системе.

Таблица 6.1

Матрица прав доступа доступа; R1-R2503 — ресурсы; User5-Userl58 — пользователи

	userl	user2		userl2	userl3		userl56	user 157	userl58
R1	7771,1	0		0	0		0	0	0
R2	0	7772,		0	777150		7772,1	7772,1	0
	o o	o o	o o	o o			o o	0...0	0...0
R1500	0	0		0	777150		7771500	7771500	0
R1501	0	777 1	o o	0	0		0	0	0
R1502	0	0	o o	777150	0		0	0	0
	o o	o o			o o	o o		0...0	
R2500	777250	0		0	777250	0...0		0	0
R2501	0	777250	0	0	0	0...0	0	0	7772501,
R2502	0	0	0	777250	0		0	0	0
R2503	0	777250	0	777250	■0	0...0	*772503,	0	0

Списки доступа

Особенностью матрицы прав доступа является не только ее большая размерность, но и наличие большого числа нулевых элементов. Такой вид матриц в математике называют ***разряженными.*** Нулевое значение элемента матрицы здесь говорит о том, что для данного сочетания (субъект, объект} права доступа не определены, а именно такие сочетания составляют большинство в реальных системах. В табл. 6.1 «0...0» означает множество нулевых значений во всем интервале, а «...» множество элементов матрицы, среди которых есть нулевые и ненулевые значения.

Свойство разрежённости матрицы может быть использовано для более компактного представления правил доступа. С каждым объектом можно связать *список управления доступом* ***(Access Control List, ACL),*** в котором указаны только те субъекты (пользователи), которые имеют разрешения на доступ к данному объекту (ресурсу). Ясно, что количество субъектов в данном списке будет значительно меньше общего числа субъектов, определенных в матрице.

Например, для объекта R2503 список включает только три субъекта — *user2, userl2* и *userl56,* для которых определены права доступа к этому файлу 7712503,2* ^2503,12 и Ш2503Д56 соответственно. Такие списки должны быть созданы для всех ресурсов. Набор списков настолько же универсален и гибок, как и матрица, и вместе с тем имеет более компактный вид, так как он не включает пустые элементы матрицы (рис. 6.1).

Список ACL состоит из *элементов управления доступом (Access Control Element, ACE),* каждый из которых описывает права доступа определенного пользователя к данному ресурсу. На рис. 6.1 список ACL состоит из трех элементов ACE.

Права доступа могут быть определены, как по отношению к ресурсам, так и по отношению к пользователям. В последнем случае его называют *списком разрешений* (capability). На рис. 6.2 показан список

Рис. 6.1. **Список управления доступом к объекту R2503**

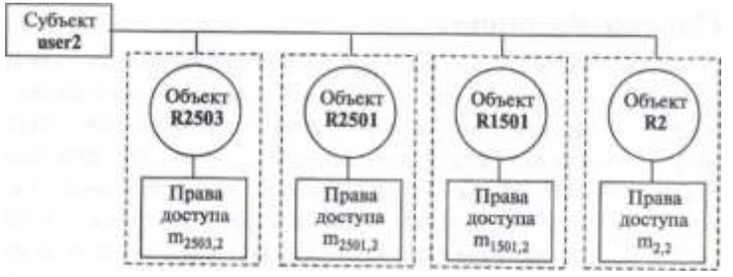


Рис. 6.2. **Список разрешений пользователя user2**

разрешений, которые имеет пользователь user2: ему разрешен доступ к ресурсам R2, R1501, R2501 и R2503.

Очевидно, что совокупность списков управления доступом ко всем ресурсам системы несет ту же самую информацию, что и совокупность списков разрешений для всех пользователей, так как и те и другие являются разными проекциями одной и той же матрицы. В одних реализациях систем управления доступом (например, в большинстве операционных систем) используются ограничения, заданные для объекта — ACL, а в других (например, в некоторых расширениях системы Kerberos, включающих авторизацию) — ограничения для субъекта — списки разрешений.

Группы

Другим способом «сжатия» матрицы является определение прав доступа для групп субъектов по отношению к группам объектов (табл. 6.2). Такое представление возможно, когда многие элементы матрицы имеют одинаковое значение. Пусть, например, все множество из 2503 информационных ресурсов оказалось возможным разделить на три группы:

R1-R1500 — файлы, содержащие открытую информацию;

R1501-R2499 — файлы, содержащие данные для служебного пользования;

R2500-R2503 — файлы, содержащие секретные данные.

При этом многим пользователям в данной организации предоставлены одинаковые права доступа к этим видам ресурсов, в частности группа пользователей — Userl-Userll, в которую входят руководители подразделений, имеют право читать, записывать и уничтожать файлы, содержащие открытую информацию, читать и записывать в файлы сданными для служебного пользования и только читать секретную информацию. Наряду с группой «руководители» созданы еще две группы: «рядовые сотрудники» (Userl2-Userl56) и «гости» (Userl57 и Userl58).

Таблица 6.2

Матрица с большим числом одинаковых элементов

	userl user2	userl2 userl3 ... userl56	userl57 userl58
R1 R2 R1500	Mu = {rtiij} читать, писать, удалять 1 < *i* ^ 1500; 1 < j < И	Mn = {rriij} читать, писать, 1 * sg 1500; 12 < *j* 156	M13 = {rriij} читать 1 < *i* sg 1500; 157 < *j* sg 158
R1501 R1502 R2499	м21 = {rriij} читать, писать, 1501 ^ г ^ 2499; 1 < *j* < 11	M2 2 = {rriij} читать, 1501 < *i* sj 2499; 12 ^ *j* s; 156	M23 = {rriij} = 0 (все виды доступа запрещены), 1501 < г sg 2499; 157 < *j* sg 158
R2500 R2501 R2502 R2503	M31 = {rriij} читать, 2500 ^ г ^ 2503; 1 < *j* < и	M2 3 = {rriij} = 0 (все виды доступа запрещены), 2500 «5 *i* < 2503; 12 sj *j* sg 156	M33 = {m12} = 0 (все виды доступа запрещены), 2500 sg г $ 2503; 157 < *j* sg 158

Все пользователи, входящие в одну группу, имеют одинаковые права. Это дает возможность компактно описать права доступа с помощью матрицы меньшей размерности (табл. 6.3). Для нашего очень условного примера число элементов в полной матрице 158х2503 = = 395474, а в матрице, основанной на групповом представлении, — всего 9.

В некоторых случаях, если существует простое ***правило*** определения прав доступа, хранение матрицы вообще не требуется, значения элементов матрицы могут вычисляться системой управления доступом динамически. Например, пусть все объекты и субъекты системы изначально снабжены метками из одного и того же множества. Кроме

Таблица 6.3

Компактное представление матрицы объединением в группы объектов и субъектов

Группа объектов	Г руппа субьектов		
	Г руппа «Руководители» (userl-userll)	Группа «рядовые сотрудники» (userl2-userl56)	Группа «Гости» (userl57 и userl58)
Открытые данные (R1-R1500)	Мц (читать, писать, уничтожать)	Mi2 (читать, писать)	*M13* (читать)
Данные для служебного пользования (R1501-R2499)	M2i (читать, писать)	M22 (читать)	Л723 (все виды доступа запрещены)
Секретные данные (R2500-R2503)	M31 (читать)	*M32* (все виды доступа запрещены)	M33 (все виды доступа запрещены)

того, предположим для простоты изложения, что для всех объектов определен только один вид операции доступа. И пусть существует правило: доступ к объекту разрешен, если метки субъекта и объекта совпадают, и не разрешен, если не совпадают. Имея такое правило, нет смысла заранее создавать и хранить матрицу — проще вычислять соответствующий элемент при каждой попытке доступа.

Способы назначения прав

Выше мы рассматривали различные подходы к хранению и представлению информации о правах доступа, не придавая значение тому, каким образом они были назначены. Однако способ назначения прав (авторизации) существенно влияет на способ управления доступом. Существует два основных подхода к авторизации:

- для авторизации выделяется особый *полномочный орган* (authority), который принимает все решения о наделении пользователей правами относительно всех объектов;
- функции принятия решений по авторизации *делегируются* некоторым субъектам.

Как видим, управление доступом может быть реализовано множеством различных способов, отражающих разные подходы к заданию и приведению в исполнение ограничений, однако большинство реализуемых на практике способов может быть отнесено к одной из следующих категорий:

- ***дискреционный метод доступа*** (DAC, Discretionary* Access Control), называемый также избирательным или произвольным методом доступа;
- ***мандатный метод доступа*** (MAC, Mandatory18 19 Access Control), называемый также принудительным;
- ***ролевой доступ*** (RBAC — Role-based Access Control), называемый также недискреционным управлением доступом (Nondiscretionary Access Control).

Помимо этих методов, взятых «в чистом виде», система управления доступом может базироваться на их комбинации.

Дискреционный метод управления доступом

Одно из первых систематических изложений принципов DAC было предпринято в 1987 году в документе NCSC-TG-003-87 «Руководство по дискретному управлению доступом». В то время DAC-модель была самой распространенной схемой управления доступом, таковой она остается и по сегодняшний день — большинство универсальных ОС реализуют дискреционную модель. В документе NCSC дается следующее определение DAC-метода:

Дискреционный метод представляет собой средство ограничения доступа к объектам, базирующееся на уникальных идентификаторах субъекта и/или групп, к которым этот субъект относится. Управление

Discretionary — действующий по своему усмотрению.

доступом DAC является дискреционным, или произвольным в том смысле, что субъект, обладающий некоторыми разрешениями на доступ к объектам, может по своему усмотрению передать часть своих полномочий (иногда прямо, а иногда — опосредовано) другим субъектам.

Таким образом, можно сформулировать следующие две главных особенности дискреционного метода:

1. Права доступа в методе DAC описываются в виде *списков* ***ACL,*** которые дают возможность гибкого и гранулированного (т. е. тонко дифференцированного) определения набора разрешенных операций для каждого отдельного пользователя по отношению к каждому отдельному ресурсу; и пользователи и ресурсы задаются уникальными идентификаторами.

2. В методе DAC право назначать права на доступ к объектам *делегируются* отдельным пользователям-владельцам объектов. То есть им разрешается действовать «по своему усмотрению» и назначать другим пользователям права доступа к тем объектам, владельцами которых они являются. Таким образом процедура авторизации является распределенной между множеством пользователей-владельцев. Владельцами считаются пользователи, создавшие объект, или пользователи, которые были назначены владельцами другими уполномоченными на то пользователями или системными процессами. Владелец имеет полный контроль над созданным им объектом и несет всю полноту ответственности за управление доступом к нему. Вместе с тем, он может назначать права доступа к своим объектам, руководствуясь некоторым, принятом на предприятии правилом.

Основным достоинством метода DAC является его *гибкость,* обусловленная свободой пользователей наделять правами или аннулировать права других пользователей по доступу к своим ресурсам, а также возможностями тонкой настройки набора разрешенных операций. Однако это достоинство имеет свою оборотную сторону. Как и всякая распределенная система, система управления доступом по методу DAC страдает от невозможности гарантированно проводить общую политику, осуществлять надежный контроль действий пользователей. Любая политика безопасности, принятая на предприятии,

19 Mandatory — обязательный/ принудительный.

может быть нарушена в результате ошибочных или вредительских действий пользователей.

Другой недостаток дискреционного метода связан с тем, что здесь права на доступ определяются по отношению к объекту, а не его содержимому. Это означает, что любой пользователь, имеющий доступ к файлу с некоторым ACL1, может скопировать его содержимое в другой файл, характеризуемый другим списком ACL2. Остановимся на этом подробнее.

Пользователь осуществляет доступ к объектам операционной системы не непосредственно, а с помощью прикладных процессов, которые запускаются от его имени. Вход пользователя в систему порождает ***процесс-оболочку,*** который поддерживает диалог с пользователем и запускает для него другие процессы. Процесс-оболочка получает от пользователя символьное имя и пароль и находит по ним числовые идентификаторы пользователя и его групп. Эти идентификаторы связываются с каждым процессом, запущенным оболочкой для данного пользователя. Говорят, что ***процесс выступает от имени*** данного пользователя или данных групп пользователей. В наиболее типичном случае любой порождаемый процесс наследует идентификаторы пользователя и групп от процесса-родителя.

Такой способ наследования прав является врожденным ***изъяном*** DAC-метода, делающим систему уязвимой для атаки типа «Троянский конь». Рассмотрим пример. Если один пользователь обладает правами на доступ к файлу TopSecret с секретными данными, а другой — нет, то последний может поступить следующим образом. Он может разработать для этих целей специальную программу, имеющую самое невинное назначение, например системную утилиту. Однако, кроме своей явной функции, эта утилита делает тайную работу по копированию файла TopSecret в файл, принадлежащий злоумышленнику. Если эту программу запустит злоумышленник, то у него ничего не выйдет, так как эта утилита унаследует его права и система DAC сможет защитить файл TopSecret от несанкционированного доступа. Совсем другое дело, если злоумышленник сможет под благовидным предлогом устроить так, чтобы эту программу запустил пользователь с высоким уровнем прав. Тогда, получив необходимые привилегии, утилита с встроенным в нее «троянским конем» скопирует содержимое секретного файла в файл злоумышленника.

Этот пример показывает, что системы с контролем доступа DAC не могут быть использованы там, где требуется очень высокий уровень защиты информации. Действительно, выполняемый на компьютере программный комплекс может включать программы, поставляемые как надежными производителями, так и менее надежными разработчиками, которым нельзя доверять на 100 %. Если код программы не был подвергнут тщательной проверке на предмет наличия в ней «троянских коней», то тогда, какие бы строгие ограничения прав доступа не были введены, выполнение этой программы несет риск нарушения безопасности данных.

Рассмотрим нескоторые положения дискреционного управления на примере доступа к файлам. Подчеркнем, что в современных операционных системах для контроля доступа используются одни и те же механизмы к

объектам любого типа, отличия заключаются лишь в наборе операций, характерных для того или иного класса объектов. Например, для файлов это операции чтения, записи, удаления, выполнения, а для принтера — перезапуск, очистка очереди документов, приостановка печати документа и т. д. Во всех этих случаях действует общая схема контроля доступа к ресурсу: пользователи пытаются выполнить с разделяемым ресурсом определенные операции, а ОС решает, имеют ли пользователи на это право.

В DAC-методе важную роль играет понятие «группа пользователей». Определение групп обычно выполняется специальным пользователем, имеющим на это право, таким как системный администратор. Каждый пользователь и каждая группа пользователей имеет символьное имя, а также уникальный числовой идентификатор. В общем случае один и тот же пользователь может входить в несколько разных групп, получая в результате «суммарные» права доступа. При выполнении процедуры логического входа в систему пользователь сообщает свое символьное имя и пароль, а операционная система определяет соответствующие числовые идентификаторы пользователя и групп, в которые он входит. Все идентификационные данные, в том числе имена и идентификаторы пользователей и групп, пароли пользователей, а также сведения о вхождении пользователя в группы хранятся в специальном файле (файл /etc/passwd в Unix) или специальной базе данных (в Windows NT).

Система контроля доступа ОС должна предоставлять средства для задания прав пользователей по отношению к объектам дифференцированно по операциям, например пользователю может быть разрешена операция чтения и выполнения файла, а операция удаления — запрещена. В разных операционных системах для одних и тех же типов ресурсов может быть определен свой список дифференцируемых операций доступа. Для файловых объектов этот список может включать следующие операции:

- создание файла;
- уничтожение файла;
- открытие файла;
- закрытие файла;
- чтение файла;

- запись в файл;
- дополнение файла;
- поиск в файле;
- получение атрибутов файла;
- установка новых значений атрибутов;
- переименование;
- выполнение файла;
- чтение каталога;
- смена владельца;
- изменение прав доступа.

Набор файловых операций ОС может состоять из большого количества элементарных операций, а может включать всего несколько укрупненных операций. Приведенный выше список является примером первого подхода, который позволяет весьма тонко управлять правами доступа пользователей, но создает значительную нагрузку на администратора. Пример укрупненного подхода демонстрируют операционные системы семейства Unix, в которых существует всего три операции с файлами и каталогами: читать (read, r), писать (write, w) и выполнить (execute, x). Хотя в Unix для операций используется всего три названия, в действительности им соответствует гораздо больше операций. Например, содержание операции выполнить зависит от того, к какому объекту она применяется. Если операция «выполнить файл» интуитивно понятна, то операция «выполнить каталог» интерпретируется как поиск в каталоге определенной записи. Поэтому администратор Unix, по сути, располагает большим списком операций, чем это кажется на первый взгляд.

В ОС Windows NT разработчики применили гибкий подход — они реализовали возможность работы с операциями над файлами на двух уровнях: по умолчанию администратор работает на укрупненном уровне (уровень стандартных операций), а при желании может перейти на элементарный уровень (уровень индивидуальных операций).

Практически во всех операционных системах, реализующих дискреционный метод управления доступом, права на выполнение операций пользователей и групп пользователей по отношению к этому файлу или каталогу описываются в виде списков управления доступом (ACL). Во многих случаях система использует назначение прав по умолчанию (by default). Например, при создании некоторым пользователем файла к нему приписывается ACL, в котором уже присутствует элемент ACE, описывающий права этого пользователя как владельца файла.

Список управления доступа к некоторому файлу или каталогу является частью характеристик данного файла или каталога и хранится

на диске в соответствующей области. Не все файловые системы поддерживают списки управления доступом, например его не поддерживает файловая система FAT, так как она разрабатывалась для однопользовательской однопрограммной операционной системы MS-DOS, для которой задача защиты от несанкционированного доступа была не актуальна.

В приведенном на рис. б.3 примере процесс, который выступает от имени пользователя с идентификатором 3 и групп с идентификаторами 14 и 72, пытается выполнить операцию записи (W) в файл.

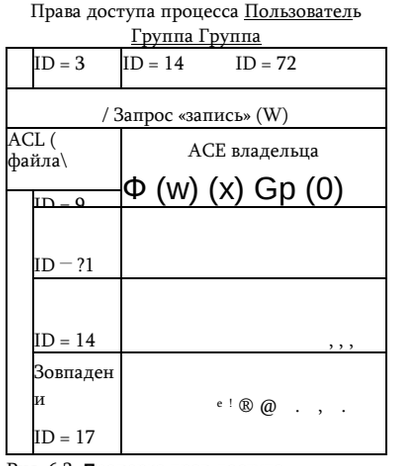


Рис. 6.3. **Проверка прав доступа**

Файлом владеет пользователь с идентификатором 17. Операционная система, получив запрос на запись, находит характеристики безопасности файла (на диске или в буферной системной области) и последовательно сравнивает все идентификаторы процесса с идентификатором владельца файла и идентификаторами пользователей и групп в элементах ACE. В данном примере один из идентификаторов группы, от имени которой выступает процесс, а именно 14, совпадает с идентификатором одного из элементов ACE. Так как пользователю с идентификатором 14 разрешена операция чтения (признак W имеется в наборе операций этого элемента), то ОС разрешает процессу выполнение операции.

В разных операционных системах реализации дискреционного метода управления доступом имеют свои особенности. В части 4 эти особенности будут рассмотрены на примере операционных систем Unix и Windows NT.

Мандатный метод управления доступом

Мандатный доступ позволяет реализовать системы, отвечающие самым строгим требованиям безопасности, как правило, они используются в правительственных и военных учреждениях или в других организациях, для которых чрезвычайно важен высокий уровень защиты данных.

К основным чертам мандатного метода управления доступом можно отнести следующие:

Таблица 6.4

Правило мандатного доступа, представленное в виде матрицы

Уровень секретности объектов	Уровень допуска субъектов		
	Уровень от «совершенно секретно» и ниже	Уровень от «секретно» и ниже	Уровень данных для служебного пользования
«Совершенно секретно»	Доступ разрешен	0	0
«Секретно»	Доступ разрешен	Доступ разрешен	0
Данные для служебного пользования	Доступ разрешен	Доступ разрешен	Доступ разрешен

- при мандатном доступе авторизацию и управление доступом осуществляет центральный *полномочный орган,* отвечающий за безопасность (обычно в роли такого органа выступает операционная система);
- в MAC-системах решение о предоставлении права доступа принимается операционной системой динамически *на основе правила.* Правило разрабатывается уполномоченными на то лицами на основе политики безопасности. Правило доступа выбирается достаточно простым, таким, чтобы решения могли приниматься операционной системой автоматически, без участия человека. Простота правил достигается тем, что как субъекты так и объекты разбиваются на небольшое число групп. Каждой группе объектов присваивается *уровень (гриф) секретности,* а группам субъектов — *уровни допуска* к объектам того или иного уровня секретности. В разных системах могут быть приняты разные правила, но все они базируются на сравнении уровня секретности объекта и уровня допуска субъекта. Например, правило может быть следующим: субъекту разрешается доступ к объекту, если уровень его допуска равен или выше уровня секретности объекта. В табл. 6.4 показано представление этого правила в виде матрицы.

Пользователи должны принимать решение системы как данность, они лишены возможности управлять доступом к своим ресурсам или передавать свои права другим пользователям. В отличие от DAC- систем, мандатный доступ имеет централизованный характер и позволяет жестко проводить принятую политику безопасности.

Элементы, описывающие уровни секретности объектов или уровни допуска субъектов, называют ***метками безопасности (****security labels).* Мандатный метод управления доступом предусматривает назначение меток безопасности всем без исключения субъектам и объектам системы, чтобы в дальнейшем они использовались системой для принятия решения о допуске. Отсюда следует критичность обеспечения целостности значений меток безопасности для эффективности работы MAC-метода.

В большинстве случаев для адекватного отражения политики безопасности невозможно сформулировать правило, основанное на учете только уровней секретности и допусков. К одному и тому же уровню секретности могут быть отнесены самые различные материалы, а в соответствии с *принципом минимальных привилегий* пользователь должен получать доступ только к той информации, которую ему необходимо знать.

Для того чтобы сделать возможным более специфическое задание прав доступа, в метки безопасности объекта и субъекта добавляется информация о конкретном виде данных, к которому относится данный объект или к

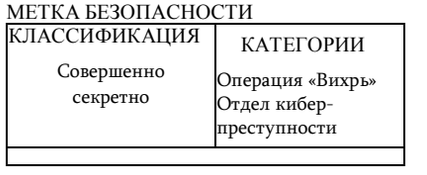


Рис. 6.4. **Структура метки безопасности объекта/субъекта**

которому разрешен доступ данному субъекту соответственно.

Таким образом каждая метка безопасности состоит из двух частей (рис. 6.4):

- часть, отражающая уровень секретности/допуска, называемая ***классификацией,***
- часть, характеризующая вид информации, называемая ***категорией.***

Категория относит данные к определенному виду информации. Например, разные категории могут быть присвоены материалам, относящимся к разным проектам, разным подразделениям, разным профессиональным группам. Разделение документов по категориям отражает специфику организации. Одному и тому же объекту/субъекту может быть присвоено несколько категорий. Например, отчет о завершении этапа некоторой антитеррористической операции может быть отнесен не только к категории материалов, касающихся данной операции, но и дополнительно отнесен к категории материалов подразделения, занимающегося этой работой. Объекты одной категории могут быть классифицированы по-разному, например часть отнесена к более высокому уровню секретности, а часть — к более низкому.

Уровни секретности/допуска, которых обычно немного, образуют иерархию от наивысшего до самого низкого уровня. Субъект, имеющий допуск к некоторому уровню, получает его и по отношению ко всем нижележащим уровням.

Правило, определяющее право доступа, строится на анализе обоих частей меток безопасности объекта и субъекта. Доступ разрешается, если выполняются два условия:

- классификация субъекта равна или выше классификации объекта (в таком случае говорят, что субъект *доминирует)',*

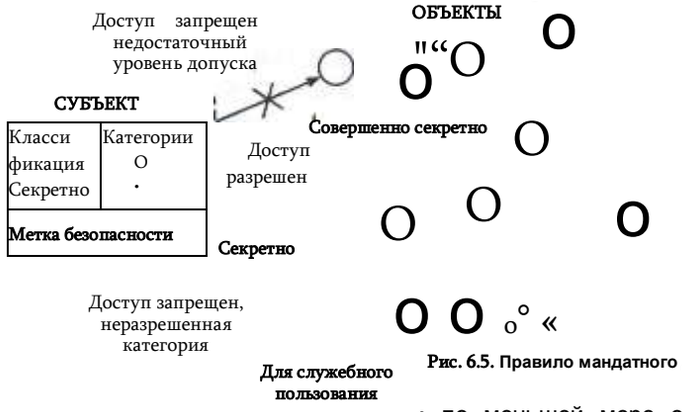


Рис. 6.5. Правило мандатного доступа

• по меньшей мере одна из категорий объекта, к которому пытается получить доступ субъект, совпадает хотя бы с одной из категорий данного субъекта.

Множество категорий является плоским, неструктурированным. Рисунок 6.5 иллюстрирует соотношение между классификацией и категорией. Здесь разная закраска кружков используется для обозначения разных категорий объектов. На рисунке показано три уровня классификации: «совершенно секретно», «секретно» и «для служебного пользования». Объекты одной категории могут принадлежать разным уровням классификации. В метке безопасности субъекта указана классификация— «секретно», и перечислены две категории, к которым ему разрешен доступ. На рисунке стрелками показаны три попытки доступа. Попытка обращения к уровню «совершенно секретно» была заблокирована системой из-за недостаточно высокого уровня допуска субъекта. Обращение к объекту уровня «секретно» была разрешена, так как классификация субъекта равна классификации объекта, а категория объекта совпала с одной из категорий, указанных в метке безопасности субъекта.

Попытка доступа к объекту уровня «для служебного пользования» была пресечена, хотя субъект и имеет более высокий уровень допуска («секретно»), В данном случае ограничением служит категория объекта, которая не совпадает ни с одной категорией субъекта.

В некоторых реализациях мандатного управления доступом используются элементы дискреционного доступа. Мандатный алгоритм доступа ограничивает группы пользователей возможностью получения доступа к группам ресурсов исходя из соотношения их уровней допуска и секретности соответственно. А дискреционный алгоритм применяется для предоставления пользователю возможности опреде

лять другим пользователям конкретный тип операции по отношению к ресурсу, находящемуся под его контролем. При этом права, предоставляемые пользователем, не должны выходить за пределы ограничений, накладываемых алгоритмом MAC. То есть для доступа к ресурсу пользователь должен быть наделен правом доступа ко всем ресурсам данного уровня секретности по алгоритму MAC, а сверх этого у него должно быть право выполнить операции над конкретным ресурсом, заданные алгоритмом DAC.

Мандатный доступ, как уже было сказано, является более безопасным, чем дискреционный, но для его эффективной реализации требуется большой объем подготовительной работы, а после запуска системы необходимо поддерживать в актуальном состоянии метки безопасности существующих объектов, а также назначать метки новым ресурсам и пользователям.

Ролевое управление доступом

Метод управления доступом RBAC, основанный на ролях, по сравнению с методами DAC и MAC является более приближенным к реальной жизни. Как видно из названия, основным его свойством является использование «ролей». Понятие «роль» в данном контексте ближе всего к понятию «должность» или «круг должностных обязанностей». Поскольку одну и ту же должность могут занимать несколько людей, то и одна и та же роль может быть приписана разным пользователям.

Роли устанавливаются для целей авторизации. Набор ролей в RBAC-системе должен в некоторым образом (не однозначно) соответствовать перечню различных должностей, существующих на предприятии, к которому эта система относится. Система RBAC лучше всего работает в организациях, в которых существует четкое распределение должностных обязанностей.

Разрешения приписываются ролям, а не отдельным пользователям или группам пользователей (рис. 6.6). А уже затем те или иные роли приписываются пользователю. Например, в системе управления доступом, развернутой в банке, всем юристам будет приписана роль «юрист», трейдерам — роль «трейдер», менеджерам — роль «менеджер» и т. д. Процесс определения ролей должен включать тщательный анализ того, как функционирует организация, какой набор функций должен выполнять работник, имеющий ту или иную должность. Каждой из ролей назначаются права доступа, необходимые и достаточные пользователям для выполнения служебных обязанностей, обусловленных приписыванием к данной роли.

Каждому пользователю может быть приписано несколько ролей (с некоторыми ограничениями, о которых будет сказано далее). Во

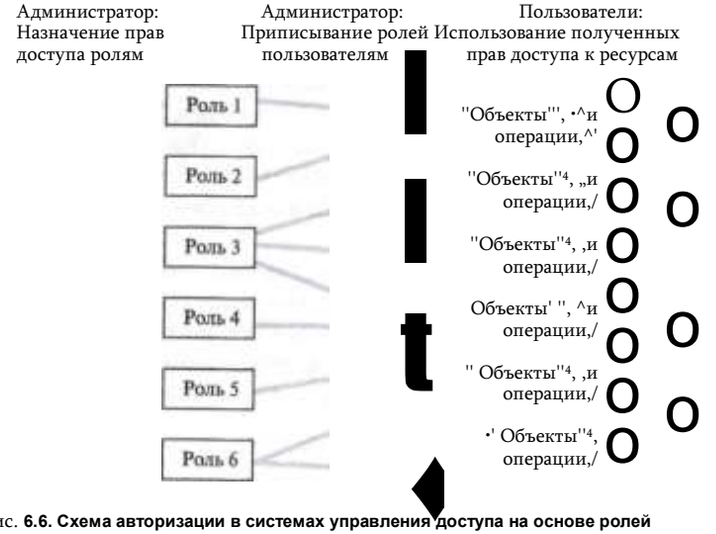


Рис. **6.6. Схема авторизации в системах управления доступа на основе ролей**

время сеанса работы пользователя все роли, которые ему назначены, становятся *активными* и он получает права доступа, являющиеся результатом объединения прав доступа всех этих групп.

Все пользователи, играющие одну и ту же роль, имеют идентичные права. Изменение производственной ситуации — расширение бизнеса, внедрение новых технологий, продвижение сотрудника по служебной лестнице или перевод в другое подразделение и др. — все это может вызвать аннулирование одной роли пользователя и приписывание ему другой роли. Такой подход упрощает администрирование прав доступа: вместо необходимого в DAC- и MAC-методах отслеживания и обновления прав каждого отдельного пользователя, в методе RBAC достаточно изменить роль или заменить одну роль на другую.

Таким образом, в системе RBAC имеются удобные механизмы для соблюдения принципа минимальных привилегий. И хотя теоретически метод DAC позволяет проводить еще более тонкую настройку прав пользователя, практически невозможно проконтролировать этот процесс так, чтобы добиться реализации этого принципа. В системе, где механизм назначения прав распределен между всеми пользователями, очень сложно отследить ситуацию, когда набор прав пользователя становится неадекватным решаемым им задачам.

Иерархия ролей

Согласно природе производственных отношений должностные обязанности сотрудников, занимающих разные позиции, могут частично перекрываться. Некоторые самые общие функции, такие, на-

Рис. 6.7. **Отношения наследования ролей: *а* — независимая роль; *б*— роль R1 является наследницей роли R2**

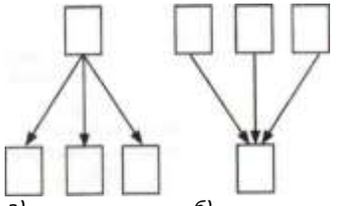


Рис. 6.8. **Виды наследования ролей; *а* — одна предшественница и несколько наследниц; *б* — несколько предшественниц и одна наследница**

пример, как ознакомление с инструкциями по соблюдения режима работы предприятия, резервирование отпусков, фиксирование на сайте интранет индивидуального рабочего графика и др., могут быть обязательными для всех сотрудников. Применительно к ролям это означает, что администратор должен выполнять много рутинной работы по приписыванию одних и тех же прав доступа разным ролям, в том числе вновь создаваемым. Решением этой проблемы является иерархическая организация ролей, когда одна роль может включать другую роль, тем самым расширяя свой набор прав за счет добавления прав, ассоциированных с инкапсулированной ролью.

Иерархия ролей создается определением для них отношений, называемых ***наследованием:*** в соответствии с этим определением, если роль R2 является наследницей R1, то все права роли R1 приписываются к правам роли R2, а все пользователи роли R2 приписываются к пользователям роли R1 (рис. 6.7). Таким образом, установление отношений наследования является еще одним способом наделения пользователя правами, наряду с явным назначением пользователю некоторой роли.

Отношения наследования относятся к типу «многие ко многим», т. е. у одной роли может быть несколько наследниц и одна роль может быть наследницей нескольких ролей (рис. 6.8).

Иерархия ролей обычно в той или иной степени отражает структуру реального предприятия. На рис. 6.9 показан фрагмент организационной структуры предприятия.

Как было сказано выше, должность «сотрудник», без уточняющих определений, ассоциируется с некоторым обязательным набором прав. Должность сотрудника отдела по работе с клиентами (ОРК) требует дополнительных полномочий, например допуска к базе данных клиентов. Должность менеджера по работе с клиентам добавляет к должностным обязанностям рядового сотрудника отдела ОРК

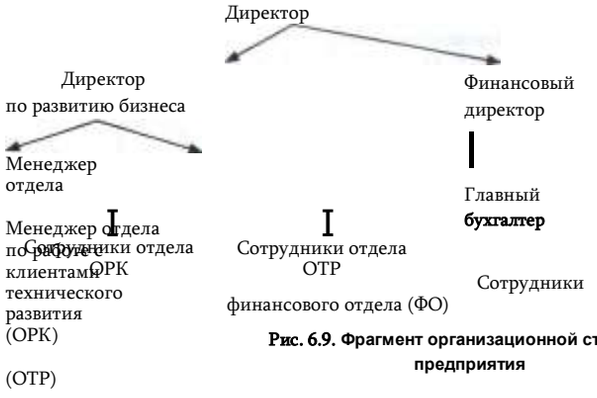


Рис. 6.9. Фрагмент организационной структуры предприятия

еще ряд функций. Например, менеджер обязан разрабатывать план увеличения клиентской базы, что требует доступ к некоторым финансовым документам. Находящийся с ним на одном уровне менеджер отдела развития (ОР) также нуждается в расширении прав доступа к информационным ресурсам по отношению к рядовым сотрудникам отдела ОР. Директор по развитию бизнеса обладает полномочиями менеджеров ОРК и ОР с добавленными к ним дополнительными правами. И наконец, на высшем организационном уровне находится директор, полномочия которого не обязательно являются объединением полномочий всех должностей, но он определенно должен иметь более широкие права, чем тот минимальный набор прав, которым обладает каждый сотрудник предприятия.

Такой *организационной структуре* может быть поставлена в соответствие *ролевая структура,* полученная в результате установления между ролями отношений наследования (рис. 6.10). Роль «сотрудник» представляет собой те общие, наличествующие у всех сотрудников организации права. После установления отношений наследования с ролями «сотрудник ОРК», «сотрудник ОТР», «сотрудник ФО» и «директор», все общие права сотрудников оказались неявным образом добавлены к этим ролям-наследницам, а их пользователи соответственно переместились наверх. Наследники следующей ступени —

Рис. 6.10. **Структура ролей, образованная отношениями наследования**

роли «менеджер ОРК» и «менеджер ОТР» сами являются предшественниками для роли «директор по развитию», которая таким образом аккумулировала права этих двух ролей.

Разделение обязанностей

Важным положением безопасности является *принцип разделения обязанностей,* в соответствии с которым некоторые должностные функции не должны поручаться одному и тому же человеку. К примеру, сотрудник, которому назначена роль «инженер», побывав в командировке, должен после возвращения составить финансовый отчет о своих тратах. Затем этот отчет должен быть проверен и представлен к оплате. Эти функции возлагаются на сотрудника, отнесенного к роли «сотрудник финансового отдела». Понятно, что такое совмещение функций, т. е. одновременная принадлежность одного пользователя к роли «инженер» и роли «сотрудник финансового отдела» является нежелательным. Чтобы избежать таких ситуаций, в методе RBAC предусмотрен специальный механизм, накладывающий ограничения на приписывание ролей пользователям. Этот механизм действует следующим образом. Совокупность ролей, относительно совмещения которых нужно устанавливать ограничения, объединяется в устойчивую группу, которой приписывается число-ограничитель. В нашем случае это группа ролей {«инженер», «сотрудник финансового отдела»}, которой должен быть приписан ограничитель 1. Если теперь администратором будет сделана попытка приписать пользователю обе эти роли, то система заблокирует эти действия. Этот механизм может работать в более «мягком» варианте — когда ограничение касается только ролей, находящихся в активном состоянии. При установлении ограничений, связанных с соблюдением принципа разделения обязанностей, необходимо учитывать принадлежность пользователя к тем ролям, которые были ему приписаны в результате наследования.

В заключение приведем некоторые характерные особенности ролевого управления доступом:

1. RBAC *сочетает в себе черты мандатного и дискреционного* способа управления доступом.

2. Ролевую систему управления доступом *легче администрировать и контролировать,* чем дискреционную. В DAC права назначаются пользователю «мелкими порциями» что позволяет ему выполнять ту или иную отдельную операцию над отдельным объектом (запись в определенный файл, чтение другого файла, запускать некоторую программу). Такой способ помогает с ювелирной точностью создавать и индивидуально настраивать комплекс прав доступа пользователя, однако он является очень трудоемким и, в следствие этого возрастает возможность ошибок. В RBAC права доступа выдаются в

виде «глыбы» — интегрированного набора разрешений, рассчитанных на возможность выполнения некоторых относительно сложных операций: заполнение кредитного документа, генерация отчетов и др.

3, RBAC является ***централизованным*** методом управления доступом — так же как и в мандатном методе пользователь лишен возможности управлять назначением прав. Назначение пользователю роли можно считать некоторым аналогом приписывания пользователю мандатной системы уровня допуска. Однако RBAC является более гибким способом, чем MAC, по возможностям настройки прав доступа он находится ближе к DAC.

4. Метод RBAC нельзя отнести к хорошо масштабируемым. Он эффективно работает в пределах единой системы или приложения, таких, например, как FreeBSD, Solaris, СУБД Oracle, MS Active Directory, но на больших предприятиях, имеющих тысячи сотрудников, поддержание множества ролей с их отношениями наследования становится сложной и запутанной задачей. Занимая промежуточное положение между мандатным и дискреционным методами, ролевое управление доступом уступает им обоим в масштабируемости. В мандатном методе управления доступом централизованный характер принятия решений (который не способствует масштабируемости), компенсируется простотой выполняемого алгоритма назначения прав. В дискреционном же методе, напротив, сложность механизма наделения правами компенсируется распределенным характером процедуры принятия решений.

Формальные модели безопасности управления доступом

При построении систем с жесткими требованиями к защите данных очень важны ***гарантии*** безопасности, иногда они оказываются даже более важными, чем функциональная полнота или производительность. Недостаточно сказать, что да, система обеспечивает высокий уровень безопасности. В некоторых случаях нужна 100%-ная гарантия. Как ее обеспечить? Никакое самое тщательное тестирование не может доказать, что система ***всегда*** будет находится в безопасном состоянии. Заказчикам требуются гарантии. В такой общей постановке задача вряд ли является разрешимой, но для некоторых частных случаев решение было найдено. А именно, были разработаны математические модели, которые с математической точностью гарантировали безопасность нескольких политик управления доступом.

Суть модели — это абстрактное представление политики. Неформальные понятия «пользователи», «программы», «файлы», «устройства», «запустить программу», «скопировать файл» отображаются в виде абстрактно определенных элементов модели «объект», «субъект», «операция читать». Затем для этих абстрактных объектов формулируются математически определенные правила, которые отражают те концепции и

принципы функционирования системы, высказанные в политике безопасности. Имея формальную модель политики управления доступом, для которой математически доказана безопасность, можно на ее основе строить реальную техническую систему, начиная с выработки технических требований и разработки архитектуры, заканчивая написанием кода программ.

Модели на основе конечного автомата

Модели на основе автоматов ***(state machine models***) являются концептуально простым и универсальным способом отображения работы реальных систем разного типа. Автомат — это абстрактный математический объект, который определяется начальным состоянием Q_0, множеством состояний {Q}, множеством входных воздействий {x} и функцией переходов ***F.*** Конечный автомат имеет конечное множество состояний. Автомат является детерминированным, если у него детерминированная (а не вероятностная) функция переходов. Функция переходов вычисляет новое состояние автомата Q_n+\ по заданному текущему состоянию Q_n и входному воздействию x_n:

$$Qn+i \sim F(Q_n, Xn)-$$

Пусть мы хотим описать в виде автомата систему управления доступом, концепции которой определены в некоторой политике безопасности.

Для отображения неформальных понятий, которыми оперирует политика безопасности, в модели используются абстрактные элементы — субъекты и объекты. Их состояния характеризуются набором атрибутов, принимающих значения из некоторого конечного множества, определенного для этой модели (например, значений разрешений на доступ).

В каждый момент времени автомат находится в одном из своих возможных состояний, каждое из которых определяется конкретным сочетанием состояний всех составляющих его элементов. Поскольку число элементов и их атрибутов конечно, то конечно и число возможных комбинаций их состояний, а значит, конечно число состояний автомата. В общем случае среди множества возможных состояний есть как безопасные, так и небезопасные. Состояние считается бе- юпасным, если оно безопасно с точки зрения используемой в данной модели политики безопасности. То есть, если в политике безопасности определено понятие несанкционированного доступа, то состояние системы, при котором возможен несанкционированный доступ, является небезопасным.

Переход из одного состояния в другое происходит в соответствии с функцией переходов в зависимости от текущего состояния и значения входного воздействия. Входными воздействиями в модели управления доступом могут являться операции изменения атрибутов объектов и субъектов (например, в результате выполнение некоторой операции субъекта над объектом). Учитывая конечность состояний автомата и конечность множества значений входных воздействий, теоретически можно провести тестирование автомата, заключающееся в выполнении всех возможных переходов.

Если начальное состояние является безопасным (всем атрибутам присвоены «правильные» начальные значения) и функция переходов автомата сконструирована так, что при поступлении любых возможных входных воздействий осуществляется переход только в безопасное состояние, то такой автомат моделирует безопасную систему. Такой вывод может быть сделан, например, в результате программного моделирования соответствующим образом определенного автомата.

Модель Белла-ЛаПадулы

Модель Белла-ЛаПадулы (Bell-LaPadula model), известная также как модель «по *read up, no write down»*,* была разработана в 1975 году Дэвидом Беллом и Леонардом ЛаПадулой в ответ на запросы военных и правительственных организаций США в предоставлении им систем, гарантированно обеспечивающих высокий уровень безопасности систем для хранения классифицированных данных. Эта модель отражает политику безопасности, основанную на концепции мандатного доступа, когда разрешение доступа определяется соотношением уровня допуска пользователя и уровня конфиденциальности документа.

Для математического доказательства безопасности модель Белла-ЛаПадулы использует концепцию конечных автоматов.

Модель Белла-ЛаПадулы включает в себя субъекты и объекты. На основе текущей политики безопасности каждому субъекту и каждому объекту назначаются собственные уровни секретности. Уровни секретности образуют иерархию от самого высокого до самого низкого. Для предоставления доступа к объекту уровень секретности субъекта сравнивается с уровнем секретности объекта.

В модели Белла-ЛаПадулы под безопасностью понимается такое состояние системы, при котором обеспечивается *конфиденциальность* информации, т. е. такое состояние системы (субъектов и объектов), при котором исключается несанкционированный доступ.

В качестве входных воздействий на систему выступают операции доступа субъектов «читать» и «записывать». В результате этих воздействий система может переходить как в безопасные состояния, так

Нельзя читать выше, нельзя записывать ниже».

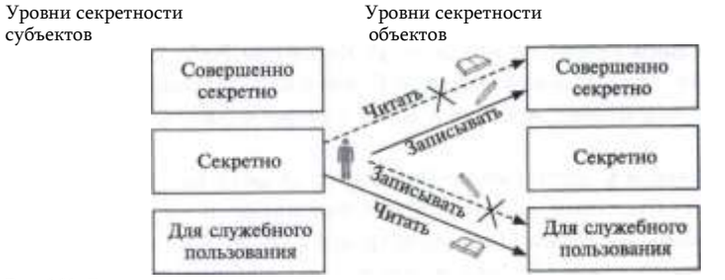


Рис. 6.11. Правила модели Белла-ЛаПадулы

и в небезопасные. Например, если в результате выполнения какой- либо операции данные становятся доступными для субъектов с более низким допуском (стал возможен несанкционированный доступ), то это означает, что система перешла в небезопасное состояние.

Ставится задача предложить такие правила управления доступом, при которых система *всегда* переходила бы только в безопасные состояния. Такие правила формулируются в рамках модели в виде двух свойств (рис. 6.11).

Простое свойство безопасности (*The Simple Security Property)* — субъекту данного уровня секретности запрещено выполнять операцию «читать» по отношению к объектам более высокого уровня секретности (правило «по read up»).

Это свойство является интуитивно понятным, действительно, если у вас допуск «секретно», то вам не позволено читать документы с грифом «совершенно секретно».

♦-Свойство (*The *-property)* — субъекту данного уровня секретности запрещено выполнять операцию «записывать» по отношению к объектам более низкого уровня секретности (правило «по write down»).

Если пользователь системы, обладающий высоким уровнем допуска, запишет некоторые данные (возможно имеющие уровень секретности, равный его собственному) в объект с более низким уровнем секретности, то они могут стать доступными субъекту с более низким, чем разрешено политикой безопасности, уровнем допуска. Следование этому свойству исключает возникновение ситуаций, подобных ситуации с «Троянским конем», которая обсуждалась в разделе «Дискреционный метод управления доступом».

С другой стороны, субъект может безопасно записать свои данные «наверх», туда, где к этой информации гарантированно получат доступ только субъекты с более высоким, чем у него уровнем секретности.

Модель Биба

Модель Биба (***Biba***), предложенная Кеннетом Биба в 1977 году, также является формальной моделью безопасного управления доступом, однако

под безопасностью в этом случае понимается ***целостность.***

Так же, как и модель Белла-ЛаПадулы, модель Биба оперирует субъектами и объектами, которые также разбиваются на иерархически организованные уровни. Но вместо уровней секретности здесь вводятся уровни целостности. Чем выше уровень целостности объекта (например, документа), тем более он заслуживает доверия, тем выше вероятность, что он содержит точные данные, тем строже правила, допускающие субъекты к работе с этими данными. Чем выше уровень, к которому отнесен субъект, тем больше ему доверяют, в том числе по возможностям модификации информации, содержащейся в объектах.

Субъекты выполняют над объектами операции «читать» и «записывать».

Модель Биба определяет два правила (рис. 6.12), при соблюдении которых система гарантированно будет находится в безопасном состоянии.

Простая аксиома целостности *(The Simple Integrity Axiom)* — субъекту данного уровня целостности запрещено выполнять операцию «читать» по отношению к объектам более низкого уровня целостности (правило «по read down»). Субъект, читая данные из объекта, характеризуемого более низким уровнем целостности, рискует «испортить» данные своего уровня, сделать их менее достоверными, поэтому такие операции должны быть запрещены. Зато он может читать проверенную, более достоверную информацию с более высоких уровней.

Аксиома 20-целостности *(The *-Integrity Axiom)* — субъекту данного уровня целостности запрещено выполнять операцию «записывать» по отношению к объектам более высокого уровня целостности

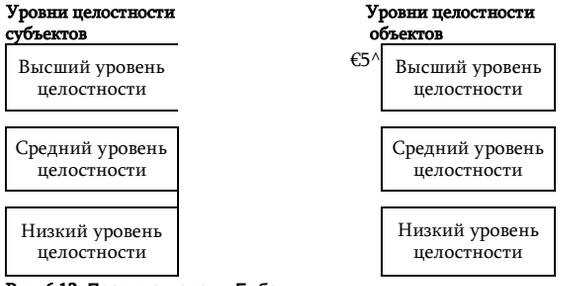


Рис. 6.12. Правила модели Биба

(правило «по write up»). Субъект, доверие к которому ограничивается некоторым уровнем целостности, не должен иметь возможность записывать данные в объекты более высокого уровня, так как он сможет внести в них искажения, неточности и тем самым снизить безопасность системы. Поток данных субъекта, направленный «вниз», не может ухудшить степень целостности объектов, имеющих более низкий уровень целостности.

Как видим, правила модели Биба, направленные на обеспечение целостности данных, прямо противоположны правилам модели Белла-ЛаПадулы, гарантирующим конфиденциальность данных. Помимо этих двух моделей разработаны также другие формальные модели безопасности, в частности модели Clark—Wilson, Take-Grant, Graham-Denning.

Аутентификация и авторизация на основе справочной службы

Процедуры авторизации и аутентификации в крупных организациях часто реализуются на основе распределенной справочной службы. Это делает их надежными, производительными и удобными в использовании и управлении.

20 Directory — спра

Назначение справочной службы

Подобно большой организации, большая компьютерная сеть нуждается в хранении и удобном доступе и как можно более полной справочной информации о самой себе. Решение многих задач в сети опирается на информацию о пользователях сети — их именах, применяемых для логического входа в систему, паролях, правах доступа к ресурсам сети, а также на информацию о ресурсах и компонентах сети — серверах, клиентских компьютерах, маршрутизаторах, шлюзах, томах файловых систем, принтерах и т. п. К числу таких таких задач относятся аутентификация и авторизация.

Результатом развития систем хранения справочной информации стало появление в сетевых ОС специальной подсистемы — справочной службы.

Справочная служба (directory services*), называемая также *службой каталогов,* имеет основной целью хранение информации, относящейся к сети, в которой эта служба установлена, с тем, чтобы предоставлять эту информацию по запросам всем пользователям и приложениям, имеющим права на доступ к этим данным.

В некоторых случаях справочная служба может быть использована и для хранения информации, не связанной с функционированием сети. Например, она может включать персональные данные служащих, такие как фамилия, имя отчество, должность, заработная плата, домашний адрес, телефон, дата рождения и т. п. Или содержать технические характеристики и данные о наличии оборудования (не обязательно сетевого) в разных подразделениях предприятия. Важно, чтобы характер этих данных совпадал с характером справочной информации, для хранения которой предназначена справочная служба, а именно данные должны быть потенциально полезными для потребителей в пределах всей сети, меняться относительно редко и в небольших масштабах.

Справочная служба хранит информацию обо всех пользователях и ресурсах сети в виде унифицированных объектов, снабженных определенными атрибутами, а также отражает взаимосвязи хранимых объектов, такие как принадлежность пользователей к определенной группе, права доступа пользователей к компьютерам и разделяемым ресурсам, вхождение нескольких узлов в одну подсеть, коммуникационные связи между подсетями, производственная принадлежность серверов и т. д.

Справочная служба позволяет выполнять над хранимыми объектами набор некоторых базовых операций, таких как *добавление* и *удаление* объекта, *изменение значений атрибута* объекта, *чтение атрибутов* и некоторые другие. Объекты справочной службы могут быть организованы в иерархические структуры, что делает возможным выполнение групповых операций над объектами. Например, администратор может определять права доступа сразу для группы пользователей или единовременно выполнять переименование/удаление сразу группы ресурсов.

Сетевая служба регулирует взаимодействие между сетевыми объектами, предоставляя доступ к информации в соответствии с заданными в ее базе данных правами доступа для разных типов клиентов этой службы. Клиентами справочной службы являются администраторы, пользователи, приложения, сетевые службы и сетевые устройства.

Альтернативой единой справочной службе сети является применение нескольких автономных справочных служб узкого назначения: одной — для аутентификации, другой — для управления сетью, третей — для разрешения имен компьютеров и т. д. Однако в крупной сети такой подход оказывается неэффективным. Даже если каждая из таких служб хорошо организована и сочетает централизованный интерфейс с распределенной базой данных, большое число справочных служб приводит к дублированию информации, усложняет администрирование и управление сетью. Например, до 2000 года в операционной системе Windows NT компании Microsoft, имелось, по крайней мере, пять различных типов справочных баз данных. Главный справочник домена (NT Domain Directory Service) хранил информацию о пользователях, требуемую для их логического входа в сеть. Данные о тех же пользователях могли содержаться и в другом справочнике, используемом электронной почтой Microsoft Mail. Еще три базы данных поддерживали разрешение адресов: служба WINS устанавливала соответствие Netbios-имен IP-адресам, справочник DNS — соответствие доменных имен IP-адресам, справочник протокола DHCP служил для автоматического назначения IP-адресов компьютерам сети. Очевидно, что такое разнообразие справочных служб усложняло жизнь администратора и приводило к дополнительным ошибкам, например когда учетные данные одного и того же пользователя нужно было ввести в несколько баз данных. Поэтому в сменивших Windows NT операционных системах на смену всем этим разрозненным справочным службам пришла интегрированная с системой DNS распределенная справочная служба Active Directory, способная хранить и поддерживать всю справочную информацию о системе. Далее в главе 16 мы подробно рассмотрим работу этой справочной службы.

Архитектура справочной службы

Для типичной справочной службы характерно использование *модели клиент-сервер*: выделенные серверы хранят базу справочной информации, которой пользуются клиенты справочной службы, передавая серверам по сети соответствующие запросы.

В соответствии с выбранной архитектурой различают следующие типы справочной службы:

- децентрализованная;
- централизованная;
- распределенная.

Децентрализованная модель была характерна для первых реализаций справочных служб (тогда для их обозначения еще не использовался данный термин). Каждый компьютер был оснащен собственной справочной службой, работающей независимо от остальных компьютеров сети; информационные запросы, порожденные на каком-либо компьютере, обрабатывались на нем же и касались ресурсов и пользователей, связанных с этим компьютером (рис. 6.13).

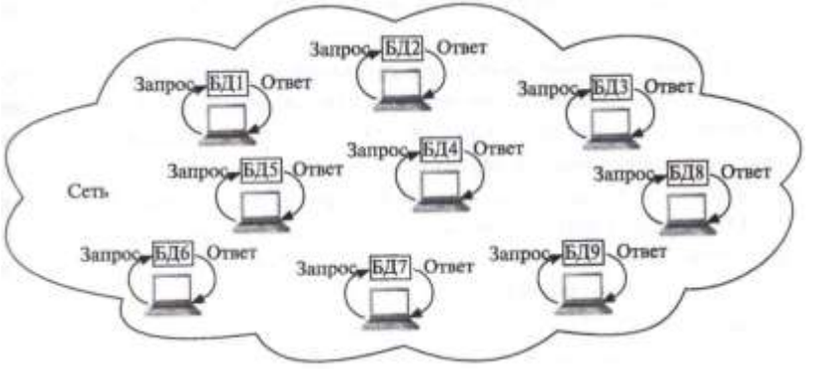


Рис. 6.13. Схема децентрализованной справочной службы

Рис. 6.14. Схема централизованной справочной службы

Децентрализованный подход в большой сети приводит к дублированию значительного объема справочных данных, а также к недопустимо большим затратам на администрирование. Последний недостаток отчасти смягчается тем обстоятельством, что децентрализованная справочная служба позволяет легко разделить работу по администрированию между несколькими специалистами.

Централизованная модель является естественной альтернативой децентрализованной модели. В соответствии с этой моделью вся справочная информация о сети и пользователях хранится и обрабатывается на одном компьютере.

На рис. 6.14 показан сервер справочной службы, обслуживающий центральную базу данных, которая объединяет справочную информацию БД1 — БД9, относящуюся к каждому из компьютеров сети. Клиентские компоненты справочной службы установлены на всех остальных компьютерах. Используя клиентскую программу, каждый пользователь и приложение, работающие на некотором компьютере сети, могут сделать запрос и получить данные о ресурсах всех других компьютеров.

При этом нет необходимости в дублировании информации. Вместо того чтобы заводить для пользователей учетные записи на тех компьютерах, на которых каждый из них работает, администратор создает и поддерживает единую базу данных для всех пользователей сети, которые обращаются к этой БД для аутентификации. Такая централизованная процедура не «привязывает» пользователя к определенному компьютеру и резко снижает избыточность и сложность ведения учетной информации.

Однако эта модель хорошо работает только в небольшой сети. Реализация справочной службы как локальной базы данных, хранящейся в виде одной копии на одном из серверов сети, не подходит для большой системы в первую очередь вследствие низкой производительности и низкой надежности такого решения.

Производительность оказывается низкой из-за того, что запросы к справочной службе от всех пользователей и приложений сети будут

поступать на единственный сервер, который обязательно перестанет справляться с их обработкой при превышении определенного порога количества запросов. Кроме того, централизация приведет к росту внутрисетевого трафика. Ситуация усугубляется, если сеть включает медленные глобальные связи. Процедура выполнения запроса к серверу может стать неприемлемо длительной из-за задержки передачи запроса, времени пребывания запроса в очереди к серверу и времени, затраченного на поиск информации в базе данных, которое может оказаться значительным, если центральная БД имеет большой объем. Другими словами, такое решение ***плохо масштабируется*** в отношении количества обслуживаемых пользователей и разделяемых ресурсов.

Надежность также не может быть высокой в системе с единственной копией данных. Отказ аппаратуры или программного обеспечения сервера, на котором поддерживается эта база данных, приведет к параличу справочной службы в масштабах всей сети.

Смягчить недостатки централизованной модели можно ***резервированием,*** т. е. поддержанием нескольких копий базы данных на разных компьютерах.

На рис. 6.15 база данных справочной службы представлена двумя идентичными экземплярами, что дает возможность повысить как ее производительность, так и надежность.

За повышение надежности и производительности централизованная система с резервированием расплачивается избыточностью и сложностью поддержания нескольких копий. Кроме того, она не ре-

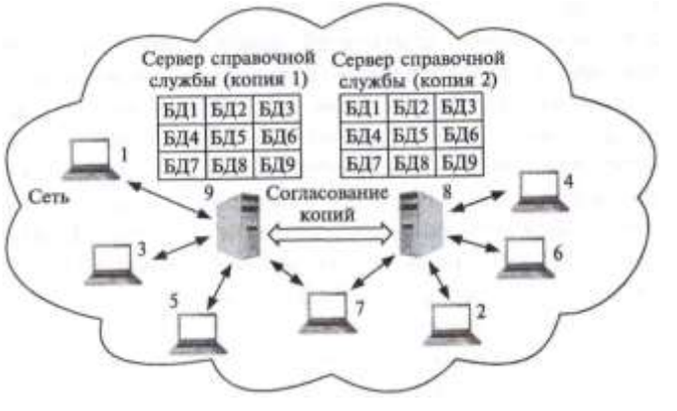


Рис. 6.15. **Схема централизованной справочной службы с резервированием**

Рис. 6.16. Декомпозиция справочной службы на не связанные между собой справочные службы доменов

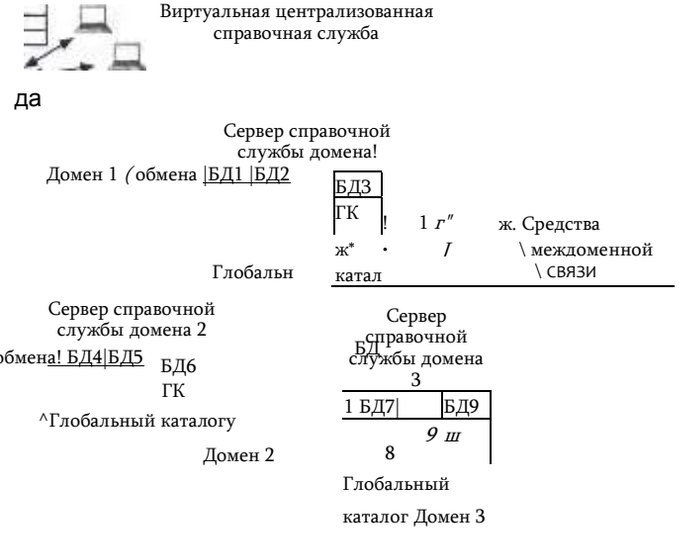
шает проблему плохой масштабируемости, характерную для любой централизованной модели.

Исходя из того, что в небольшой сети централизованная схема работает эффективно, одним из возможных решений могло бы быть разделение большой сети на части — ***домены*** — и реализация во всех доменах отдельных, не связанных между собой централизованных справочных служб. На рис. б. 16 показана сеть, разделенная на три домена, в каждом из которых работает собственная централизованная справочная служба. Базы данных, размещенные на серверах справочной службы в каждом домене, содержат лишь часть справочных данных сети, а именно те данные, которые относятся к ресурсам и пользователям соответствующих доменов.

Справочные службы доменов обладают всеми преимуществами, свойственными централизованным системам, а главный недостаток централизованных систем — плохая масштабируемость — преодолевается разделением сети на домены. Действительно, хотя увеличение размера сети и ведет к росту общего объема справочных данных, размер каждой отдельной БД может поддерживаться в разумных границах за счет образования новых доменов и связанных с ними новых баз данных. При необходимости повышения производительности и надежности в каждом домене может быть применено резервирование.

Коренным недостатком декомпозиционного подхода является то, что из-за изолированности справочных служб доменов пользователи и приложения получают удобный доступ к справочной информации только в пределах своего домена. В отсутствие какого-либо механиз

а
а



ма объединения доменов пользователю придется самому решать, где может находиться искомая информация и по какому адресу следует посылать запрос. Понятно, что такой способ организации справочной службы в виде нескольких не связанных между собой справочных служб отдельных доменов нельзя признать эффективным.

Распределенная модель предполагает наличие нескольких физически разделенных баз данных, виртуально представленных для клиентов в виде единой центральной базы данных (рис. 6.17). В данном случае виртуализация означает, что пользователь любого домена может получить информацию о любом объекте сети вне зависимости от того, с какой рабочей станции поступил запрос и где находится требуемая информация. Такая справочная служба должна скрывать от пользователей различные физические параметры сети: местонахождение серверов, применяемые коммуникационные протоколы, маршруты перемещения запросов, характеристики коммуникационного оборудования и др.

Существуют различные механизмы связывания доменных справочных служб в единую службу сети. Например, в справочной службе Active Directory компании Microsoft таким механизмом является ***глобальный каталог (global catalog),*** схему применения которого иллюстрирует рис. 6.17.

В то время как доменные базы данных содержат *полную* информацию об объектах соответствующего домена, в глобальном ката-

логе представлена *частичная* информация обо всех объектах сети. В качестве обязательной информации глобальный каталог хранит для каждого объекта атрибуты, которые могут быть использованы для определения местонахождения полной информации о данном объекте.

Копии глобального каталога размещают на сервере справочной службы в каждом домене. При поступлении запроса пользователя к ресурсам, находящимся вне его домена, справочная служба, пользуясь информацией из локальной копии глобального каталога, переадресует запрос к базе данных того домена, где находится интересующий пользователя объект. Все эти действия скрыты от пользователей справочной службы и выполняются автоматически.

Распределенная организация справочной службы является наиболее эффективной для крупных сетей. К числу ее достоинств можно отнести следующие:

Удобство доступа пользователей к справочной информации. В такой распределенной системе для пользователя поддерживается иллюзия единого централизованного хранилища всей информации, когда степень сложности доступа к любому объекту сети не зависит от того, с какого компьютера поступил запрос.

Удобство администрирования. Для каждой части распределенной базы данных, например домена, можно назначить отдельного администратора и наделить его правами доступа только к части информации обо всей системе.

Надежность. Распределенная система по определению имеет несколько хранилищ и центров обработки информации, а значит, при отказе одного из них система может продолжать функционирование, возможно, в ограниченном объеме. Кроме того, надежность может быть повышена за счет поддержания в каждом домене нескольких копий баз данных этого домена. Необходимые для этого процедуры согласования копий требуют значительно меньщих затрат, чем в централизованных системах, так как проводятся в пределах домена, а не всей сети.

Высокая производительность. Разделение данных между несколькими серверами снижает нагрузку на каждый сервер. Количество серверов не ограничивается числом доменов, так как в каждом домене могут быть установлены серверы, поддерживающие копии доменных БД. Повышению производительности может также способствовать приближение баз данных к источникам запросов за счет рационального разбиения сети на домены.

Хорошая масштабируемость. Распределенная служба продолжает эффективно функционировать даже в очень крупных сетях за счет возможности логической декомпозиции сети на домены. Это, в частности, позволяет ограничить объем БД, снизить вычислительные
затраты на поддержание копий БД, приблизить серверы к клиентам, уменьшить сетевой трафик, ускорить время выполнения запросов.

Вопросы к главе 6

1. Ограничения доступа могут быть сформулированы и представлены в виде:

а) правила:

б) матрицы прав доступа:

в) соответственным образом сконфигурированного пользовательского интерфейса;

г) ACL.

2. Избирательный метод доступа обозначают следующей аббривиатурой:

а) DAC;

б) MAC:

в) RBAC;

г) ACL.

3. Какие свойства из перечисленных характеризуют дискреционный метод управления доступом:

а) описание прав доступа дается в виде *списков ACL.*

б) право назначать права на доступ к объектам *делегируются* отдельным пользователям-владельцам объектов:

в) права доступа субъектов к объектам определяются правилом;

г) данная система управления доступом страдает от невозможности гарантированно проводить общую политику.

4. Какие свойства из перечисленных могут быть отнесены к мандатному методу управления доступом:

а) права доступа отдельного пользователя описываются ACE;

б) решение о предоставлении права доступа принимается операционной системой *на основе правила'.*

в) субъект может по своему усмотрению передать часть своих полномочий другим субъектам;

г) данная система управления доступом позволяет гарантированно проводить общую политику;

д) ни одного из них.

5. Какие из утверждений справедливы ВСЕГДА:

а) субъекту разрешается доступ к объекту, если уровень его допуска равен или выше уровня секретности объекта;

б) матрица доступа позволяет задать права доступа субъектов к объектам;

в) управление доступом в MAC осуществляет операционная система;

г) правило доступа в MAC учитывает только соотношение уровней допуска и секретности;

д) ни одно из них.

6. В методе MAC метка доступа включает:

а) ACL;

б) классификацию;

в) категорию;

г) приоритет;

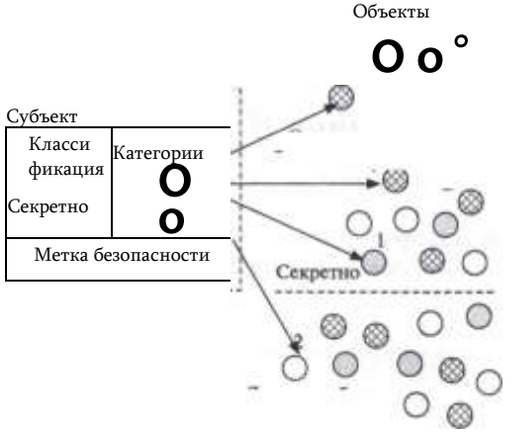


Рис. 6.18. К вопросу 7

Д) ACE.

7. Укажите, в каком случае (1-4, см. рис. 6.18) попытка запроса субъекта будет удачной?

8. Какие из следующих утверждений относительно RBAC верны?

а) одному пользователю может быть преписано несколько ролей;

б) несколько пользователей могут выступать в одной роли;

в) должно выполняться взаимнооднозначное соответствие между ролями и пользователями;

г) пользователь может делегировать свои права доступа другим пользователям;

д) права доступа назначаются ролям, а не пользователям;

е) роль — это совокупность прав доступа, которые были назначены администратором каждому пользователю, приписанному данной роли.

9. Если роль A является наследницей роли B, то:

а) права, относящиеся к роли A, добавляются к правам роли B;

б) права, относящиеся к роли B, добавляются к правам роли A;

в) пользователи, относящиеся к роли A, добавляются к роли B;

г) пользователи, относящиеся к роли B, добавляются к роли A.

10. Какие из следующих характеристик метода RBAC справедливы?

а) метод RBAC является хорошо масштабируемым;

б) RBAC сочетает в себе черты мандатного и дискреционного способа управления доступом;

в) ролевую систему управления доступом легче администрировать, чем дискреционную;

г) RBAC является централизованным методом управления доступом;

д) по возможностям настройки прав доступа RBAC находится ближе к MAC.

11. Какая из моделей безопасности управления доступом ориентирована на обеспечение конфиденциальности?

а) DAC;

б) Biba;

в) MAC;

г) Bell—LaPadula;

д) RSA.

12. Какие правила доступа определяет модель Biba?

а) «no read down»;

б) «no write down»;

в) «no write up»;

г) «no read up».

13. Какими достоинствами обладает распределенная справочная служба?

а) на ее основе может быть реализована процедура единого логического входа;

б) она обладает высокой надежностью;

в) она хорошо масштабируема;

г) администратор создает на одном сервере единую базу аутентификационных данных для всех пользователей сети.

7 *ТЕХНОЛОГИИ ЗАЩИЩЕННОГО КАНАЛА*

Известно, что задачу защиты данных можно разделить на две подзадачи: защиту данных ***внутри*** компьютера и защиту данных в процессе их ***передачи*** от одного компьютера к другому. Для обеспечения безопасности данных при их передаче по публичным сетям используются различные технологии защищенного канала.

Технология защищенного канала обеспечивает защиту трафика между двумя точками в открытой транспортной сети, например в Интернете. Защищенный канал подразумевает выполнение трех основных функций:

- ***взаимная аутентификация*** абонентов при установлении соединения, которая может быть выполнена, например, обменом паролями;
- защита передаваемых по каналу сообщений от ***несанкционированного доступа,*** например шифрованием;
- подтверждение ***целостности*** поступающих по каналу сообщений, например передачей одновременно с сообщением его дайджеста.

Способы образования защищенного канала

В зависимости от места расположения программного обеспечения защищенного канала различают две схемы его образования:

- схема с конечными узлами, взаимодействующими через публичную сеть (рис. 7.1,а);
- схема с оборудованием поставщика услуг публичной сети, который расположен на границе между частной и публичной сетями (рис. 7.1,б).

В первом случае защищенный канал образуется программными средствами, установленными на двух удаленных компьютерах, принадлежащих двум разным локальным сетям одного предприятия и связанных между собой через публичную сеть. Преимуществом этого подхода является полная защищенность канала вдоль всего пути следования, а также возможность использования любых протоколов

I Мша 7. Технологии защищенного канала

Рис. 7.1. Два подхода к образованию защищенного канала

создания защищенных каналов, лишь бы на конечных точках канала поддерживался один и тот же протокол. Недостатки заключаются в избыточности и децентрализованности решения. Избыточность состоит в том, что вряд ли стоит создавать защищенный канал на всем пути следования данных: уязвимыми для злоумышленников обычно являются сети с коммутацией пакетов, а не каналы телефонной сети или выделенные каналы, через которые локальные сети подключены к территориальной сети. Поэтому защиту каналов доступа к публичной сети можно считать избыточной. Децентрализация заключается в том, что для каждого компьютера, которому требуется предоставить услуги защищенного канала, необходимо отдельно устанавливать, конфигурировать и администрировать программные средства защиты данных. Подключение каждого нового компьютера к защищенному каналу требует выполнять эти трудоемкие операции заново.

Во втором случае клиенты и серверы не участвуют в создании защищенного канала — он прокладывается только внутри публичной сети с коммутацией пакетов, например внутри Интернета. Так, канал может быть проложен между сервером удаленного доступа поставщика услуг публичной сети и пограничным маршрутизатором корпоративной сети. Это хорошо масштабируемое решение, управляемое централизовано администраторами как корпоративной сети, так и сети поставщика услуг. Для компьютеров корпоративной сети канал прозрачен — программное обеспечение этих конечных узлов остается без изменений. Такой гибкий подход позволяет легко образовывать новые каналы защищенного взаимодействия компьютеров независимо от места их расположения. Реализация этого подхода сложнее — ну

жен стандартный протокол образования защищенного канала, требуется установка у всех поставщиков услуг программного обеспечения, поддерживающего такой протокол, необходима поддержка протокола производителями пограничного коммуникационного оборудования. Однако вариант, когда все заботы по поддержанию защищенного канала берет на себя поставщик услуг публичной сети, оставляет сомнения в надежности защиты: во-первых, незащищенными оказываются каналы доступа к публичной сети, во-вторых, потребитель услуг чувствует себя в полной зависимости от надежности поставщика услуг.

Иерархия технологий защищенного канала

Защищенный канал можно построить с помощью системных средств, реализованных на разных уровнях модели OSI (табл. 7.1).

Если защита данных осуществляется средствами верхних уровней (прикладного, представления или сеансового), то такой способ защиты не зависит от технологий транспортировки данных (IP или IPX, Ethernet или ATM), что можно считать несомненным достоинством. В то же время приложения при этом становятся зависимыми от конкретного протокола защищенного канала, так как в них должны быть встроены явные вызовы функций этого протокола.

Защищенный канал, реализованный на самом высоком (прикладном) уровне, защищает только вполне определенную сетевую службу, например файловую, гипертекстовую или почтовую. Так, протокол S/MIME защищает исключительно сообщения электронной почты. При таком подходе для каждой службы необходимо разрабатывать собственную защищенную версию протокола.

Таблица 7.1

Протоколы, формирующие защищенный канал на разных уровнях модели OSI

Уровень OSI	Протокол защищенного канала	Свойства протоколов защищенного канала
Прикладной уровень Уровень представления Сеансовый уровень Транспортный уровень	S/MIME SSL, TLS	Непрозрачность для приложений, независимость от транспортной инфраструктуры
Сетевой уровень Канальный уровень	IPsec PPTP	Прозрачность для приложений, зависимость от транспортной инфраструктуры
Физический уровень		

11опулярный ***протокол SSL21*** (Secure Socket Layer — слой защи- мн

21 95 % веб-сайтов в Великобритании, принимающих от клиентов информацию о кредитных и дебетовых карточках, используют для

иных сокетов) работает на уровне представления и создает защищенный канал, используя следующие технологии безопасности:

- взаимная аутентификация приложений на обоих концах защищенного канала выполняется с помощью обмена сертификатами (стандарт X.509);
- для контроля целостности передаваемых данных используются дайджесты;
- секретность обеспечивается шифрацией с средствами симметричных ключей сеанса.

Протокол SSL разработан компанией Netscape Communications для защиты данных, передаваемых между веб-сервером и веб-брау- юром, но он может быть использован и любыми другими приложениями. Работа протокола защищенного канала на уровне представления делает его более универсальным средством, чем протокол безопасности прикладного уровня. Однако для того чтобы приложение | могло воспользоваться протоколом уровня представления, в него по- прежнему приходится вносить исправления, хотя и не столь существенные, как в случае протокола прикладного уровня. Модификация приложения в данном случае сводится к встраиванию явных обращений к API соответствующего протокола безопасности.

Средства защищенного канала становятся прозрачными для приложений в тех случаях, когда безопасность обеспечивается на сетевом и канальном уровнях. Однако здесь мы сталкиваемся с другой проблемой — зависимостью сервиса защищенного канала от протокола нижнего уровня. Например, протокол PPTP, не являясь протоколом канального уровня, защищает кадры протокола PPP канального уровня, упаковывая их в IP-пакеты. При этом не имеет никакого значения, пакет какого протокола в свою очередь упакован в данном PPP-кадре: IP, IPX, SNA или NetBIOS. С одной стороны, это делает сервис PPTP достаточно универсальным, так как клиент сервиса защищенного канала может задействовать любые протоколы в своей сети. С другой стороны, такая схема предъявляет жесткие требования к типу протокола канального уровня, используемому на участке доступа клиента к защищенному каналу — для протокола PPTP таким протоколом может быть ***только PPP.*** Хотя протокол PPP очень распространен в линиях доступа, сегодня конкуренцию ему составляют протоколы Gigabit Ethernet и Fast Ethernet, которые все чаще работают не только в локальных, но и в глобальных сетях.

Работающий на сетевом уровне протокол IPSec является компромиссным вариантом. С одной стороны, он прозрачен для приложений, с другой — может работать практически во всех сетях, так как основан на широко распространенном протоколе IP и использует любую технологию

передачи такого рода данных протокол SSL.

канального уровня (PPP, Ethernet, ATM и т. д.).

Туннелирование

Туннелирование, или *инкапсуляция* — это нестандартный (отличающийся от принятого в модели OSI порядка) способ размещения пакетов некоторого протокола двух объединяемых сетей или узлов в пакеты протокола транзитной сети на ее границе и передача пактов объединяемых сетей через ***транзитную*** сеть. Туннелирование применяется не только для того, чтобы получить возможность передачи данных через промежуточную сеть, не поддерживающую протокол объединяемых сетей, т. е. обеспечить связность маршрута. Этот прием используется также для ***защиты разных подсетей от взаимного влияния,*** так как туннелированный трафик проходит через транзитную сеть прозрачным образом, изолируя ее от какого-либо воздействия со стороны объединяемых сетей.

Данное описание подходит к стандартной схеме, описанной в модели OSI, если под протоколом объединяемых сетей понимать протокол IP, а под протоколом транзитной сети — любой протокол канального уровня, например Ethernet. Действительно, IP- пакеты могут инкапсулироваться на границе сети в кадры Ethernet и передаваться в этих кадрах через транзитную сеть Ethernet в неизменном виде. А при выходе из транзитной сети IP-пакеты извлекаются из кадров Ethernet и дальше уже обрабатываются маршрутизатором.

Для того чтобы понять, в чем состоит нестандартность инкапсуляции, сначала заметим, что в этом процессе принимают участие три типа протоколов:

- протокол-пассажир;
- несущий протокол,
- ;ротокол инкапсуляции.

При стандартной работе составной сети, описанной в модели OSI (и повсеместно применяемой на практике), протоколом-пассажиром является протокол IP, а несущим протоколом — один из протоколов канального уровня отдельных сетей, входящих в составную сеть, например Frame Relay или Ethernet. Протоколом инкапсуляции также является протокол IP, для которого функции инкапсуляции описаны в стандартах RFC для каждой существующей технологии канального уровня.

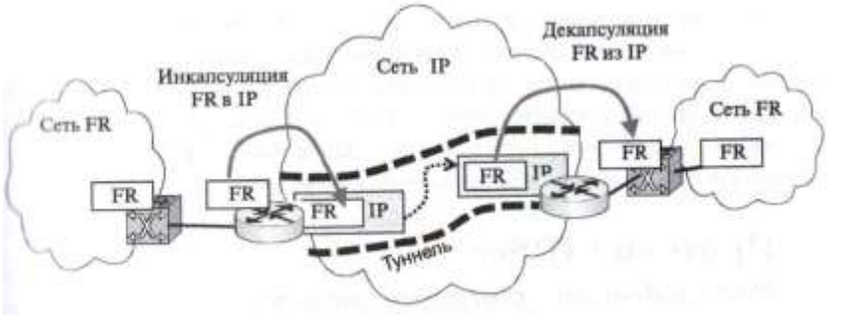


Рис. 7.2. Туннелирование трафика Frame Relay через IP-сеть

При туннелировании протоколом-пассажиром является протокол объединяемых сетей, это может быть протокол канального уровня, не поддерживаемый транзитной сетью, или же протокол сетевого уровня, например протокол IPv6, отличный от протокола сетевого уровня транзитной сети.

На рис. 7.2 показан пример сети, в которой трафик сетей Frame Relay передается по туннелю через транзитную IP-сеть, канальный уровень которой эту технологию не поддерживает, так как построен на технологии Ethernet.

Таким образом, протоколом-пассажиром в этом случае является протокол FR, а несущим протоколом — протокол IP. Пакеты протоко- ла-пассажира помещаются в поле данных пакетов несущего протокола | помощью протокола инкапсуляции. Инкапсуляция FR-кадров в IP- пакеты не является стандартной операцией для 1P-маршрутизаторов. Это дополнительная для маршрутизаторов функция описывается отдельным стандартом и должна поддерживаться пограничными маршрутизаторами транзитной сети, если мы хотим организовать такой туннель.

Инкапсуляцию выполняет пограничное устройство (обычно маршрутизатор или шлюз), которое располагается на границе между исходной и транзитной сетями. Пакеты протокола-пассажира при транспортировке их по транзитной сети никак не обрабатываются. Извлечение пакетов-пассажиров из несущих пакетов выполняет второе пограничное устройство, которое находится на границе между транзитной сетью и сетью назначения. Пограничные маршрутизаторы указывают в IP-пакетах, переносящих трафик туннеля, свои IP-адреса в качестве адресов назначения и источника.

В связи с популярностью Интернета и стека TCP/IP ситуация, когда несущим протоколом транзитной сети обычно выступает протокол IP, а протоколом-пассажиром — некоторый канальный протокол, является очень распространенной. Вместе с тем применяются и

защищенными другими схемами инкапсуляции, такие как инкапсуляция IP в IP, Ethernet в MPLS, Ethernet в Ethernet. Подобные схемы инкапсуляции нужны не только для того, чтобы согласовать транспортные протоколы, но и для других целей, например для шифрования исходного трафика или для изоляции адресного пространства транзитной сети провайдера от адресного пространства пользовательских сетей.

Протокол IPSec

Распределение функций между протоколами IPSec

Протокол IPSec называют в стандартах Интернета ***системой.*** Действительно, IPSec — это согласованный набор открытых стандартов, имеющий сегодня вполне очерченное ядро, которое в то же время может быть достаточно просто дополнено новыми функциями и протоколами.

Ядро IPSec составляют три протокола:

- ***AH*** (Authentication Header — заголовок аутентификации) — гарантирует целостность и аутентичность данных;
- ***ESP*** (Encapsulating Security Payload — инкапсуляция зашифрованных данных) — шифрует передаваемые данные, обеспечивая конфиденциальность, может также поддерживать аутентификацию и целостность данных;
- ***IKE*** (Internet Key Exchange — обмен ключами Интернета) — решает вспомогательную задачу автоматического предоставления конечным точкам защищенного канала секретных ключей, необходимых для работы протоколов аутентификации и шифрования данных.

Как видно из сказанного выше, возможности протоколов AH и ESP частично перекрываются (табл. 7.2). В то время как AH отвечает только за обеспечение целостности и аутентификации данных, ESP может шифровать данные и, кроме того, выполнять функции протокола AH (хотя, как увидим позднее, аутентификация и целостность обеспечиваются им в несколько урезанном виде). ESP может поддерживать функции шифрования и аутентификации/целостности в любых комбинациях, то есть либо всю группу функций, либо только аутентификацию/целостность, либо только шифрование.

Разделение функций защиты между протоколами AH и ESP вызвано применяемой во многих странах практикой ограничения экспорта и/или импорта средств, обеспечивающих конфиденциальность данных шифрованием. Каждый из этих протоколов может использоваться как самостоятельно, так и одновременно с другим, так что в тех

Таблица 7.2

Распределение функций между протоколами IPSec

Выполняемая функция	Протокол	
Обеспечение целосности Обеспечение аутентичности Обеспечение конфиденциальности (шифрование)	AH	ESP
Распределение секретных ключей	IKE	

мучаях, когда шифрование из-за действующих ограничений применить нельзя, систему можно поставлять только с протоколом AH. Естественно, подобная защита данных во многих случаях оказывается недостаточной. Принимающая сторона получает лишь возможность проверить, что данные были отправлены именно тем узлом, от которого они ожидаются, и дошли в том виде, в котором были отправлены. Однако от несанкционированного просмотра данных на пути их следования по сети протокол AH защитить не может, так как не шифрует их Для шифрования данных необходим протокол ESP.

Безопасная ассоциация

Для того чтобы протоколы AH и ESP могли выполнять свою работу по защите передаваемых данных, протокол IKE устанавливает между двумя конечными точками логическое соединение (рис. 7.3), которое в стандартах IPSec носит название ***безопасной ассоциации*** (Security Association, SA).

Стандарты IPSec позволяют конечным точкам защищенного канала использовать как одну безопасную ассоциацию для передачи трафика всех взаимодействующих через этот канал хостов, так и создавать для этой цели произвольное число безопасных ассоциаций, например по одной на каждое TCP-соединение. Это дает возможность выбирать нужную степень детализации защиты — от одной общей ассоциации для трафика множества конечных узлов до индивидуально настроенных ассоциаций для защиты каждого приложения.

Безопасная ассоциация в протоколе IPSec представляет собой однонаправленное (симплексное) логическое соединение, поэтому, если требуется обеспечить безопасный двусторонний обмен данными, необходимо установить две безопасные ассоциации. Эти ассоциации в общем случае могут иметь разные характеристики, например в одну

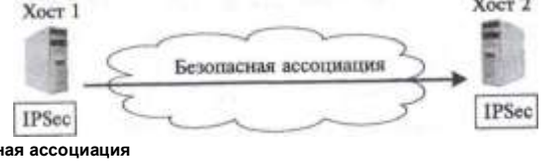


Рис. 7.3. Безопасная ассоциация

защищенно.

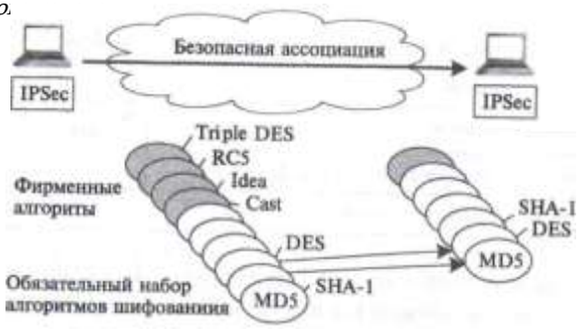


Рис. 7.4. Согласование параметров в протоколе ESP

сторону при передаче запросов к базе данных достаточно только аутентификации, а для ответных данных, несущих ценную информацию, дополнительно нужно обеспечить конфиденциальность.

Установление безопасной ассоциации начинается с взаимной аутентификации сторон, потому что все меры безопасности теряют смысл, если данные передаются или принимаются не тем лицом или не от того лица. Выбираемые далее параметры SA определяют, какой из двух протоколов, AH или ESP, будет применяться для защиты данных, какие функции будет выполнять протокол (например, можно выполнять только аутентификацию и проверку целостности или, сверх того, еще и обеспечивать конфиденциальность). Очень важными параметрами безопасной ассоциации являются также секретные ключи, используемые в работе протоколов AH и ESP.

Протокол IPSec допускает как автоматическое, так и ручное установление безопасной ассоциации. При ручном способе администратор конфигурирует конечные узлы так, чтобы они поддерживали согласованные параметры ассоциации, включая секретные ключи. При автоматической процедуре установления SA протоколы IKE, работающие по разные стороны канала, выбирают параметры в ходе переговорного процесса. Для каждой задачи, решаемой протоколами AH и ESP, предлагается несколько схем аутентификации и шифрования (рис. 7.4). Это делает протокол IPSec очень гибким средством. Заметим, что выбор дайджест-функции для решения задач целостности и аутентификации никак не влияет на выбор функции шифрования, обеспечивающей конфиденциальность данных.

Для обеспечения совместимости в стандартной версии IPsec определен некоторый обязательный «инструментальный» набор, в частности, для аутентификации данных всегда может быть использована одна из стандартных дайджест-функций MD5 либо SHA-1, а в число алгоритмов шифрования непременно входит AES. При этом производители продуктов, в которых используется IPSec, вольны расширять

протокол включением других алгоритмов аутентификации и симметричного шифрования, что они с успехом и делают. Например, многие реализации IPSec поддерживают популярный алгоритм шифрования Inple DES, а также сравнительно новые алгоритмы: Blowfish, Cast, CDMF, Idea, RC5.

Транспортный и туннельный режимы

Протоколы AH и ESP могут защищать данные в двух режимах:

- в *транспортном режиме,* при котором передача IP-пакета через сеть выполняется с помощью оригинального заголовка этого пакета;
- в *туннельном режиме,* при котором исходный пакет помещается в новый IP-пакет, и передача данных по сети выполняется на основании заголовка нового IP-пакета.

Применение того или иного режима зависит от требований, предъявляемых к защите данных, а также от роли, которую играет в сети узел, завершающий защищенный канал. Так, узел может быть хостом (конечным узлом) или шлюзом (промежуточным узлом). Соответственно, имеются три схемы применения протокола IPSec: хост-хост, шлюз-шлюз или хост-шлюз.

В схеме *хост-хост* защищенный канал, или, что в данном контексте одно и то же, безопасная ассоциация, устанавливается между двумя конечными узлами сети (см. рис. 7.3). Тогда протокол IPSec работает на конечных узлах и защищает данные, передаваемые от хоста 1 к хосту 2. Для схемы хост-хост чаще всего используется транспортный режим защиты.

В соответствии со схемой *шлюз-шлюз* защищенный канал устанавливается между двумя промежуточными узлами, так называемыми *шлюзами безопасности* (Security Gateway, SG), на каждом из которых работает протокол IPSec (рис. 7.5). Защищенный обмен данными

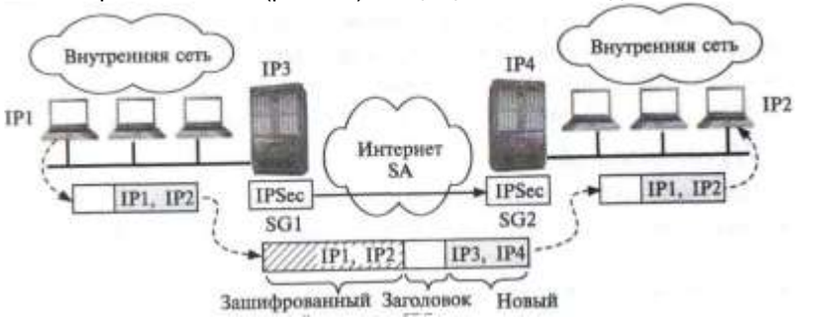


Рис. 7.5. **Работа защищенного канала по схеме шлюз-шлюз** в **туннельном режиме**

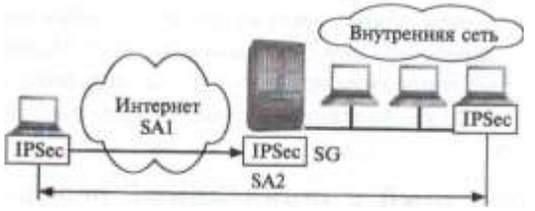


Рис. 7.6. Схема защищенного канала хост-шлюз

может происходить между любыми двумя конечными узлами, подключенными к сетям, которые расположены позади шлюзов безопасности. От конечных узлов поддержка протокола IPSec не требуется, они передают свой трафик в незащищенном виде через заслуживающие доверие внутренние сети предприятий. Трафик, направляемый в общедоступную сеть, проходит через шлюз безопасности, который и обеспечивает его защиту с помощью протокола IPSec. Шлюзам доступен только туннельный режим работы.

На рис. 7.5 пользователь компьютера с адресом IP1 посылает пакет по адресу IP2, используя туннельный режим протокола IPSec. Шлюз SG1 зашифровывает пакет целиком, вместе с заголовком, и снабжает его новым заголовком IP, в котором в качестве адреса отправителя указывает свой адрес — IP3, а в качестве адреса получателя — адрес IP4 шлюза SG2. Вся передача данных по составной IP-сети выполняется на основании заголовка внешнего пакета, а внутренний пакет становится при этом полем данных для внешнего пакета. На шлюзе SG2 протокол IPSec извлекает инкапсулированный пакет и расшифровывает его, приводя к исходному виду.

Схема ***хост-шлюз*** часто применяется при удаленном доступе. В этом случае защищенный канал прокладывается между удаленным хостом, на котором работает протокол IPSec, и шлюзом, защищающим трафик для всех хостов, входящих во внутреннюю сеть предприятия. Эту схему можно усложнить, создав параллельно еще один защищенный канал — между удаленным хостом и каким-либо хостом, принадлежащим внутренней сети, защищаемой шлюзом (рис. 7.6). Такое комбинированное использование двух безопасных ассоциаций позволяет надежно защитить трафик и во внутренней сети.

Протокол AH

Протокол AH позволяет приемной стороне убедиться, что:

- пакет был отправлен стороной, с которой установлена безопасная ассоциация;
- содержимое пакета не было искажено в процессе его передачи по сети;
- пакет не является дубликатом уже полученного пакета.

Две первые функции обяза-

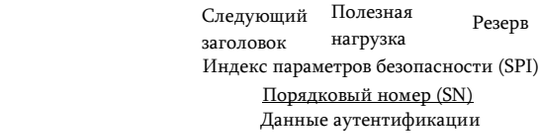


Рис. 7.7. Структура заголовка протокола AH

тельны для протокола AH, а последняя выбирается при установлении ассоциации по желанию. Для выполнения этих функций протокол AH использует специальный заголовок (рис. 7.7).

В поле *«Следующий заголовок»* (next header) указывается код протокола более высокого уровня, т. е. протокола, сообщение которого размещено в поле данных IP-пакета. Скорее всего, им будет один из протоколов транспортного уровня (TCP или UDP) или протокол ICMP, но может встретиться и протокол ESP, если он используется в комбинации с AH.

В поле *«Длина»* (payload length) содержится длина заголовка AH.

Поле *«Индекс параметров безопасности»* (Security Parameters Index, SPI) служит для связи пакета с предусмотренной для него безопасной ассоциацией. Немного позже мы обсудим это более подробно.

Поле «*Порядковый номер»* (Sequence Number, SN) указывает на порядковый номер пакета и применяется для защиты от его ложного воспроизведения (когда третья сторона пытается повторно использовать перехваченные защищенные пакеты, отправленные реально аутентифицированным отправителем). Отправляющая сторона последовательно увеличивает значение этого поля в каждом новом пакете, передаваемом в рамках данной ассоциации, так что приход дубликата обнаружится принимающей стороной (если, конечно, в рамках ассоциации будет активирована функция защиты от ложного воспроизведения). Однако в любом случае в функции протокола AH не входит восстановление утерянных и упорядочивание прибывающих пакетов — он просто отбрасывает пакет, когда обнаруживает, что аналогичный пакет уже получен. Чтобы сократить требуемую для работы протокола буферную память, используется механизм скользящего окна — на предмет дублирования проверяются только те пакеты, чей номер находится в пределах окна. Окно обычно выбирается размером в 32 или 64 пакета.

Поле «*Данные аутентификации»* (authentication data), которое содержит так называемое ***значение проверки целостности*** (Integrity Check Value, ICV), служит для аутентификации и проверки целостности пакета. Это значение является дайджестом, вычисляемым с помощью одной из двух обязательно поддерживаемых протоколом AH

защищенного канала

односторонних функций шифрования MD5 или SAH-1, но может использоваться и любая другая функция, о которой стороны договорились в ходе установления ассоциации. При вычислении дайджеста

Рис. 7.5. **Работа защищенного канала по схеме шлюз-шлюз** в **туннельном режиме**

Рис. 7.9. Структура IP-пакета, обработанного протоколом АН в туннельном режиме

Рис. 7.8. Структура IP-пакета, обработанного протоколом АН в транспортном режиме

пакета в качестве параметра ОФШ выступает симметричный секретный ключ, который был задан для данной ассоциации вручную или автоматически с помощью протокола IKE. Так как длина дайджеста зависит от выбранной ОФШ, это поле имеет в общем случае переменный размер.

Протокол АН старается охватить при вычислении дайджеста как можно большее число полей исходного IP-пакета, но некоторые из них в процессе передачи пакета по сети меняются непредсказуемым образом, поэтому не могут быть включены в аутентифицируемую часть пакета. Например, целостность поля времени жизни (TTL) в приемной точке канала оценить нельзя, так как оно уменьшается на единицу каждым промежуточным маршрутизатором и никак не может совпадать с исходным.

Местоположение заголовка АН в пакете зависит от того, в каком режиме — транспортном или туннельном — сконфигурирован защищенный канал. Результирующий пакет в транспортном режиме выглядит так, как показано на рис. 7.8.

При использовании туннельного режима, когда шлюз IPSec принимает проходящий через него транзитом исходящий пакет и создает для него внешний IP-пакет, протокол АН защищает все поля исходного пакета, а также неизменяемые поля нового заголовка внешнего пакета (рис. 7.9).

Протокол ESP

Протокол ESP решает две группы задач. К первой относятся задачи обеспечения аутентификации и целостности данных на основе дайджеста, аналогичные задачам протокола АН, ко второй — защита передаваемых данных от несанкционированного просмотра путем их шифрования.

Как видно на рис. 7.10, заголовок ESP делится на две части, разделяемые полем данных. Первая часть, называемая собственно

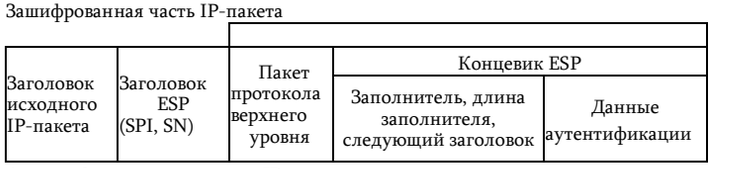


Рис. 7.10. Структура IP-пакета, обработанного протоколом ESP в транспортном режиме

заголовком ESP, образуется двумя полями (SPI и SN), назначение которых аналогично одноименным полям протокола АН, и размещается перед полем данных. Остальные служебные поля протокола ESP, называемые *концевиком ESP,* расположены в конце пакета.

Два поля концевика — *«Следующий заголовок»* и *«Данные аутентификации»* — также аналогичны полям заголовка АН. Поле данных аутентификации отсутствует, если при установлении безопасной ассоциации принято решение не использовать возможности протокола ESP, касающиеся обеспечения целостности. Помимо этих полей концевик содержит два дополнительных поля — *«Заполнитель»* и *«Длина заполнителя».* Заполнитель может понадобиться в трех случаях. Во-первых, для нормальной работы некоторых алгоритмов шифрования необходимо, чтобы шифруемый текст содержал кратное число блоков определенного размера. Во-вторых, формат заголовка ESP требует, чтобы поле данных заканчивалось на границе четырех байтов. И, наконец, заполнитель можно использовать, чтобы скрыть действительный размер пакета в целях обеспечения так называемой частичной конфиденциальности трафика. Правда, возможность маскировки ограничивается сравнительно небольшим объемом заполнителя — 255 байтов, поскольку большой объем избыточных данных может снизить полезную пропускную способность канала связи.

На рис. 7.10 показано размещение полей заголовка ESP в *транспортном режиме.* В этом режиме ESP не шифрует заголовок IP- пакета, иначе маршрутизатор не сможет прочитать поля заголовка и корректно осуществить продвижение пакета между сетями. В число шифруемых полей не попадают также поля SPI и SN, которые должны передаваться в открытом виде для того, чтобы прибывший пакет можно было отнести к определенной ассоциации и предотвратить ложное воспроизведение пакета.

В *туннельном режиме* заголовок исходного IP-пакета помещается после заголовка ESP и полностью попадает в число защищаемых полей, а заголовок внешнего IP-пакета протоколом ESP не защищается (рис,- 7.11).

защищенного канала

Зашифрованная часть IP-пакета

Заголовок внешнего IP-пакета	Заголовок ESP (SPI, SN)	Заголовок исходного IP-пакета	Пакет протокола верхнего уровня	Концевик ESP	
				Заполнитель, длина заполнителя, следующий заголовок	Данные аутентификации

Аутентифицированная часть IP-пакета

Рис. 7.11. Структура IP-пакета, обработанного протоколом ESP в туннельном режиме

Базы данных SAD И SPD

Итак, технология IPSec предлагает различные методы защиты трафика. Каким же образом протокол IPSec, работающий на хосте или шлюзе, определяет способ защиты, который он должен применить к трафику? Решение основано на использовании в каждом узле, поддерживающем IPSec, двух типов баз данных:

- безопасных ассоциаций (Security Associations Database, SAD);
- политики безопасности (Security Policy Database, SPD).

При установлении безопасной ассоциации, как и при любом другом логическом соединении, две стороны принимают ряд соглашений, регламентирующих процесс передачи потока данных между ними. Соглашения фиксируются в виде набора параметров. Для безопасной ассоциации такими параметрами являются, в частности, тип и режим работы протокола защиты (AH или ESP), методы шифрования, секретные ключи, текущий номер пакета в ассоциации и другая информация. Наборы текущих параметров, определяющих все активные ассоциации, хранятся на обоих оконечных узлах защищенного канала в виде баз данных безопасных ассоциаций (SAD). Каждый узел IPSec поддерживает две базы SAD — одну для исходящих ассоциаций, другую для входящих.

Другой тип базы данных —■ база данных политики безопасности (SPD) — определяет соответствие между IP-пакетами и установленными для них правилами обработки. Записи SPD состоят из полей двух типов — полей селектора пакета и полей политики защиты для пакета с данным значением селектора (рис. 7.12).

Селектор в SPD включает следующий набор признаков, на основании которых можно с большой степенью детализации выделить защищаемый поток:

- IP-адреса источника и приемника могут быть представлены как в виде отдельных адресов (индивидуальных, групповых или широковещательных), так и диапазонами адресов, заданными с помощью верхней и нижней границ либо с помощью маски;
- порты источника и приемника (т. е. TCP- или UDP-порты);

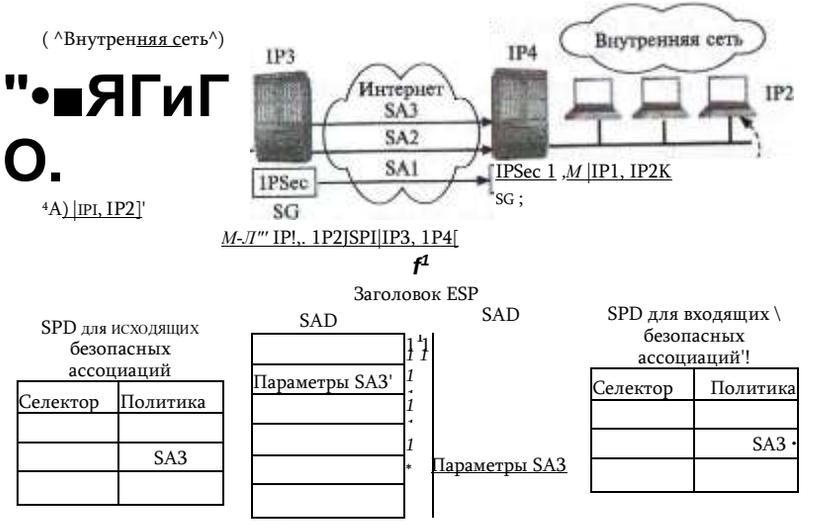


Рис. 7.12. Использование баз данных SPD и SAD

- тип протокола транспортного уровня (TCP, UDP);
- имя пользователя в формате DNS или X.500;
- имя системы (хоста, шлюза безопасности и т. п.) в формате DNS или X.500.

Для каждого нового пакета, поступающего в защищенный канал, IPSec просматривает все записи в базе SPD и сравнивает значение селекторов этих записей с соответствующими полями IP-пакета. Если значение полей совпадает с каким-либо селектором, то над пакетом выполняются действия, определенные в поле политики безопасности данной записи. Политика предусматривает передачу пакета без изменения, отбрасывание или обработку средствами IPSec.

В последнем случае поле политики защиты должно содержать ссылку на запись в базе данных SAD, в которую помещен набор параметров безопасной ассоциации для данного пакета (на рис. 7.12 для исходящего пакета определена ассоциация SA3). На основании заданных параметров безопасной ассоциации к пакету применяется соответствующие протокол (на рисунке — ESP), функции шифрования и секретные ключи.

Если к исходящему пакету нужно применить некоторую политику защиты, но указатель записи SPD показывает, что в настоящее время нет активной безопасной ассоциации с требуемой политикой, то IPSec создает новую ассоциацию с помощью протокола IKE, помещая новые записи в базы данных SAD и SPD.

Базы данных политики безопасности создаются и администрируются, либо пользователем (этот вариант больше подходит для хоста),

защищенного *канала*

либо системным администратором (вариант для шлюза), либо автоматически (приложением).

Ранее мы выяснили, что установление связи между исходящим IP-пакетом и заданной для него безопасной ассоциацией происходит путем селекции. Однако остается другой вопрос: как ***принимающий*** узел IPSec определяет способ обработки прибывшего пакета, ведь при шифровании многие ключевые параметры пакета, отраженные в селекторе, оказываются недоступными, а значит, невозможно определить соответствующую запись в базах данных SAD и SPD и, следовательно, тип процедуры, которую надо применить к поступившему пакету? Именно для решения этой проблемы в заголовках AH и ESP предусмотрено поле SPI. В это поле помещается указатель на ту строку базы данных SAD, в которой записаны параметры соответствующей безопасной ассоциации. Поле SPI заполняется протоколом AH или ESP во время обработки пакета в отправной точке защищенного канала. Когда пакет приходит в конечный узел защищенного канала, из его внешнего заголовка ESP или AH (на рисунке — из заголовка ESP) извлекается значение SPI и дальнейшая обработка пакета выполняется с учетом всех параметров заданной этим указателем ассоциации.

Таким образом, для распознавания пакетов, относящихся к разным безопасным ассоциациям, используются:

- на узле-отправителе — селектор;
- на узле-получателе — индекс параметров безопасности (SPI).

После дешифрирования пакета приемный узел IPSec проверяет его признаки (ставшие теперь доступными) на предмет совпадения с селектором записи SPD для входящего трафика, чтобы убедиться, что ошибки не произошло и выполняемая обработка пакета соответствует политике защиты, заданной администратором.

Использование баз SPD и SAD для защиты трафика позволяет достаточно гибко сочетать механизм безопасных ассоциаций, который предусматривает установление логического соединения, с дейтаграммным характером трафика протокола IP.

Вопросы к главе 7

1. Защищенный канал подразумевает выполнение следующих основных функций:

а) аутентификация отправителя получателем;

б) аутентификация получателя отправителем;

в) обеспечение конфиденциальности передаваемого сообщения;

г) подтверждение целостности передаваемого сообщения.

2. Какие цели преследует туннелирование?

а) защита транзитной сети от воздействия со стороны сетей, связываемых туннелем;

б) согласование протоколов в неоднородной сети;

в) защита трафика, передаваемого по туннелю, от воздействия со стороны транзитной сети;

г) повышение скорости передачи данных.

3. Какой адрес назначения указавается в заголовке несущего протокола при туннелировании?

а) адрес узла, которому предназначается инкапсулированное сообщение;

б) адрес пограничного устройства;

в) ни тот, ни другой;

г) адрес пограничного устройства, который в то же время может являться адресом конечного узла.

4. Какие протоколы включает IPsec?

а) SSL; б) ESP; в) AH; г) IKE; д) PGP.

5. Для передачи данных между несколькими конечными точками защищенного канала IPsec:

а) можно установить несколько безопасных ассоциаций;

б) всегда достаточно одной безопасной ассоциации;

в) можно использовать как одну, так и несколько безопасных ассоциаций;

г) число безопасных ассоциаций зависит от поставленных целей;

д) для двустороннего обмена обязательно требуется установление двух безопасных ассоциаций.

6. Какие из следующих функций являются обязательными для протокола AH?

а) подтверждение того, что пакет был отправлен стороной, с которой установлена безопасная ассоциация;

б) доказательство целостности содержимого пакета после его передачи по сети;

в) доказательство того, что пакет не является дубликатом уже полученного пакета;

г) аутентификация отправителя;

д) ни одна из них.

7. Какие задачи решает протокол ESP?

а) обеспечения аутентификации;

б) обеспечение целостности данных;

в) защита передаваемых данных от несанкционированного просмотра;

г) ни одну из них.

8. База данных политики безопасности (SPD) определяет:

а) соответствие между IP-пакетами и установленными для них правилами обработки;

б) тип и режим работы протоколов защиты AH или ESP;

в) методы шифрования;

г) секретные ключи;

д) значение текущего номера пакета в ассоциации;

е) ничего из перечисленного.

9. С какой целью в семействе протоколов IPSec функции обеспечения целостности и аутентичности данных дублируются в в двух протоколах — AH и ESP?

10. Отметьте в таблице все возможные комбинации режимов работы IPsec.

	Хост-хост	Шлюз-шлюз	Хост-шлюз
Транспортный режим			
Туннельный режим			

8 *Технологии анализа трафика и состояния сети*

Аудит
Подотчетность

Одним из важнейших требований безопасности является *подотчетность* В стандарте22 подотчетность определяется как «свойство, обеспечивающее однозначное прослеживание действий любого логического объекта». Возможность фиксировать деятельность субъектов системы, а затем ассоциировать их с индивидуальными идентификаторами пользоваелей позволяет выявлять нарушения безопасности и определять ответственных за эти нарушения.

Для обеспечения свойства подотчетности в компьютерных сетях используются различные программно-аппаратные средства, способные анализировать состояние и параметры элементов системы. К таким средствам относятся подсистемы аудита и регистрации ОС, файерволы, системы обнаружения и предотвращения вторжений, антивирусные системы, сетевые мониторы.

В рамках обеспечения подотчетности решаются следующие задачи:

- Формирование правил отбора подлежащих регистрации событий. Эту задачу выполняет администратор сети. Как правило, к числу таких событий относят попытки успешного и неуспешного логического входа в систему, запуска программ, доступа к защищаемым ресурсам, попытки изменения атрибутов объектов и полномочий пользователей и др.
- Селективная *регистрация* событий в *журнале регистрации событий* (Log on), называемая также ***протоколированием*** или ***журнализацией.*** Журнал — это совокупность хронологически упорядоченных записей о событиях, отобранных для регистрации. Нарушители могут сделать попытку «стереть следы» своей преступной деятельности, поэтому данные журнала регистрации должны

22 Часто в понятие аудит включают не только анализ, но и сбор данных.

быть надёжно защищены от модификации и разрушения неавторизованными субъектами.

- Анализ накопленной информации — *аудит*.* Аудит — по своей природе является реактивным (а не проактивным) действием — т. е. записи становятся достоянием специалиста по безопасности уже по прошествии некоторого времени после того, как эти события произошли. В тех случаях, когда анализ информации о событиях проводится в реальном времени, мы имеем дело с системами *обнаружения* и *предупреждения вторжений* — они позволяют пресечь атаку раньше, чем она нанесет большой ущерб.

Задачи аудита

Аудит (auditing) — это набор процедур учета и анализа всех событий, представляющих потенциальную угрозу для безопасности системы. Аудит позволяет «шпионить» за выбранными объектами и выдавать сообщения тревоги, когда, например, какой-либо рядовой пользователь попытается прочитать или модифицировать системный файл. Если кто-то пытается выполнить действия, выбранные системой безопасности для мониторинга, то система аудита пишет сообщение в журнал регистрации, идентифицируя пользователя. Системный менеджер может готовить отчеты безопасности, которые содержат информацию из журнала регистрации. Для «сверхбезопасных» систем предусматриваются аудио- и видеосигналы тревоги, устанавливаемые на машинах администраторов, отвечающих за безопасность.

Термин «аудит безопасности» может использоваться в более широком смысле для обозначения процедур анализа событий, угрожающих безопасности не только компьютерной сети, но *информационной системы* предприятия в целом. В этом случае аудит включает обследование IT-инфраструктуры, оценку принятых на предприятии политик безопасности и реализующих эти политики инструкций. Важную часть аудита составляет оценка рисков.

Поскольку никакая система безопасности не гарантирует защиту на уровне 100 %, последним рубежом в борьбе с нарушениями оказывается система аудита. Эту мысль изящно выражает популярное изречение «Prevention is ideal, but detection is a must!», которое говорит о том, что, бесспорно, предупреждение нарушений — это желательная, но, увы, часто недостижимая цель, в то время как обнаружение и фиксация нарушений — это то, что должно быть сделано обязательно.

Действительно, после того как злоумышленнику удалось провести успешную атаку, пострадавшей стороне не остается ничего другого, как обратиться к службе аудита. Если при настройке службы аудита были правильно заданы события, которые требуется отслеживать,

то подробный анализ записей в журнале может дать много полезной информации. Эта информация, возможно, позволит найти злоумышленника или, по крайней мере, предотвратить повторение подобных атак устранением уязвимых мест в системе защиты. Аудит действует также как угроза потенциальным нарушителям, предупреждая их о том, что в случае несанкционированных действий, они легко могут быть выведены на чистую воду.

Данные аудита могут использоваться для разрешения правовых споров, например сотрудники предприятия могут посчитать некоторые действия по сбору данных нарушающими их законные права. Закон может не разрешить администрации сети использование реального имени пользователя при трассировке его обращений к тем или иным сайтам Интернета. Но даже, если аудит не выходит за рамки закона, администрация предприятия должна учитывать тот факт, что в результате неправильно реализованной политики аудита на предприятии может сложится атмосфера «слежки»,которая отрицательно сказывается на эффективности работы предприятия. Во избежание этого рекомендуется заранее предупреждать пользователей сети о том, какие из их действий регистрируются.

Аудиту должны подлежать не только события, инициированные пользователями, но и действия администраторов сети, которые тоже должны быть подотчетны наравне со всеми. Поэтому время от времени аудит должен проводится сторонними организациями.

Движущей силой процесса накопления данных о системе может быть как *осуществление* того или иного события, так и *наступление некоего заранее заданного момента,* когда от системы требуется выполнение определенных действий по сбору данных. В некоторых случаях это «расписание» является настолько плотным, что процесс сбора данных о состоянии системы можно назвать непрерывным. Функциональная компонента аудита, которая выполняет процесс непрерывного наблюдения за параметрами системы, называется подсистемой ***мониторинга.*** Объектами мониторинга могут выступать, например, загрузка процессора, интенсивность сетевого трафика, статус сетевого интерфейса (активный или пассивный), включенное или выключенное устройство печати. Специфическим видом мониторинга является мониторинг нажатия клавиш пользователем. Это средство используется при проведении расследований, когда имеется явный подозреваемый.

При реализации аудита в больших сложных системах, состоящих из множества подсистем со своими собственными средствами протоколирования событий, иногда необходимо приложить специальные усилия по «синхронизации» журналов регистрации. В частности, поскольку в разных частях большой системы одни и те же объекты иметь разные имена, и напротив, разные объекты — одинаковые имена, администраторам, возможно, придется принять специальное соглашение об однозначном именовании объектов.

Ведение журнала событий может потребовать слишком много *ресурсов вычислительной системы* (дискового пространства, вычислительной мощности процессора), а также *рабочего времени персонала.* Поэтому очень важно соблюдать баланс между количеством различных видов событий, подлежащих регистрации, с одной стороны, и затрачиваемыми на их протоколирование ресурсами и возможностью их анализа, с другой стороны. Отбор событий должен учитывать предысторию процесса функционирования системы и специфику текущей ситуации.

Записи журнала регистрации событий могут использоваться и с целью расследования какого-либо уже произошедшего инцидента с помощью реконструкции последовательности событий. Однако анализ может оказаться трудной задачей уже при продолжительности периода наблюдений в несколько дней. Даже самые простые журналы событий содержат такое огромное количество сведений, что их практически невозможно анализировать «вручную», без специальных средств.

Для обработки и анализа данных журнала регистрации могут быть использованы следующие средства:

- средства предварительной обработки данных аудита, предназначенные для сжатия информации журнала регистрации за счет удаления из него малоинформативных записей, которые только создают ненужный «шум»;
- средства выявления аномальных ситуаций, которые используют один из самых эффективных приемов распознавания нарушений. Он заключается в том, что постоянно отслеживаются *среднестатистические* значения параметров системы, которые сравниваются с их *текущими* значениями. Например, факт входа в систему в часы, нетипичные для режима работы какого-либо сотрудника, уже является поводом для выдачи предупреждения администратору. В некоторых случаях подозрительные всплески активности могут помочь распознать готовящуюся или проводимую в данный момент атаку. (Удивительно, что копирование сотен тысяч файлов за короткое время, как это было в случаях Сноудена и Мэннинга, осуществивших массивные утечки информации из сетей правительственных и военных учреждений США, не вызвало подозрений у лиц, регулярно проводивших аудит этих сетей.);
- средства распознавания атак по их сигнатурам, т е. по характерным признакам атаки, выражающимся в виде специфическо

го фрагмента кода, типичном поведении, аномально частому обращению к какому-либо сетевому порту компьютера. Атака детектируется сравнением последовательности событий в реальной системе с той, которая содержится в сигнатуре.

Файерволы

Файервол является пограничным устройством, которое защищает одну часть от другой. Эффективность файервола прямо зависит от архитектуры сети, от того, насколько правильно была проведена ***логическая структуризация*** (***сегментация***) сети. Другой базовой технологией, лежащей в основе файервола, является фильтрация трафика.

Сегментация сети

Сегментация состоит в разбиении сети на части, которые часто называют подсетями.23 Эта задачу решают специально предназначенные сетевые устройства — коммутаторы, маршрутизаторы, шлюзы, разрешая или запрещая прохождение информационных потоков между узлами сети. Разрешения выдаются в зависимости от значений различных параметров трафика. Правильно сегментирование сети может положительно сказаться на ее производительности и управляемости, что косвенным образом приводит к улучшению защищенности сети.

Некоторые способы логической структуризации сети ***прямо*** влияют на сетевую безопасность. Так, структуризация IP-сетей на основе масок, технологии виртуальных локальных и глобальных сетей, трансляция сетевых адресов NAT позволяют скрыть внутреннюю структуру сети предприятия от внешнего наблюдения и тем самым повышают ее защищенность.

При правильной сегментации сети и управлении трафиком появляется возможность разместить наиболее критичные ресурсы и сервисы в те фрагменты, куда не поступают потоки потенциально опасной информации.

Другой целью, которая может быть достигнута правильным структурированием сети, является организация некоторого ***небольшого*** числа «контрольных пунктов», через которые должен быть направлен весь сетевой трафик. В этих точках, называемых иногда *choke point* (узкий проход), должны быть сконцентрированы средства анализа и управления трафиком. Таким способом устанавливается жесткий контроль над информационными потоками в сети, позволяющий выявить угрозы и атаки. В качестве контрольных пунктов могут выступать различные пограничные устройства: маршрутизаторы, перенаправляющие трафик из одной подсети в другую, или файерволы и прокси-серверы, которые выполняют работу по фильтрации входных и выходных потоков подсетей или приложений.

Фильтрация трафика

23 Не путать с термином «подсеть», использующимся в протоколах TCP/IP, который там имеет более узкое значение.

Как выше было сказано, разделение сети на части выполняют сетевые устройства анализируя трафик. Решение пропускать информацию или не пропускать принимается на основе заданного устройству правила, которое называется *правилом фильтрации,* или просто *фильтром.*

Разные типы сетевых устройств выполняют фильтрацию в соответствии с разными правилами, определяющими их функциональность. Например, стандартное функционирование концентратора заключается в том, что любой кадр, поступивший на любой его интерфейс независимо от адреса назначения кадра и других параметров, ***повторяется*** на всех остальных его интерфейсах. Здесь мы сталкиваемся с вырожденным случаем фильтрации, при котором фильтрации как таковой нет. Работа устройства на основе такого «правила» обеспечивает прямое взаимодействие всех узлов, подключенных к его интерфейсам, что является очень полезным свойством. В то же время в сети могут возникать ситуации, когда такая тотальная доступность узлов нежелательна. Примером может служить сервер финансового отдела, доступ к которому желательно разрешить только с компьютеров нескольких конкретных сотрудников этого отдела. Конечно, доступ можно ограничить на уровне операционной системы или системы управления базой данных самого сервера, но для надежности желательно иметь несколько эшелонов защиты и ограничить доступ еще и на уровне сетевого трафика.

Для решения задачи логической структуризации сети используются более развитые сетевые устройства — коммутаторы и маршрутизаторы.

На рис. 8.1 показан ***коммутатор*** с четырьмя интерфейсами, к каждому из которых подключена свой фрагмент большой сети. Любой кадр, поступающий на интерфейс маршрутизатора, подвергается анализу и продвижению в соответствии со следующим стандартным правилом фильтрации: кадр, имеющий некоторый адрес назначения (на рисунке A1), повторяется только на том интерфейсе (на рисунке интерфейс 1), к которому подключена подсеть, имеющая в своем составе узел с данным адресом (на рисунке подсеть 1). Таким образом фильтрация, разделяя трафик на четыре отдельных потока, позволяет изолировать одну часть сети от другой.

Узлы сети.

(A8j

(A4^) Подсеть V

Подсеть 4

Коммутатор

Адрес	Интерфейс
A1 A4 A2	
A3 A5	2
A6	3
A7 A8	

Адресная таблица

Подсеть 2

Рис. 8.1. **Схема фильтрации трафика коммутатором**

Стандартное функционирование ***маршрутизаторов*** определяется другим, более сложным правилом фильтрации на основе адресной таблицы маршрутизатора, в котором учитываются другие параметры кадров (называемых в этом случае пакетами).

Наряду со *стандартными правилами фильтрации,* базирующимися на адресных таблицах, многие модели сетевых устройств — маршрутизаторов и коммутаторов — позволяют администраторам задавать ***дополнительные*** условия фильтрации кадров. Такие фильтры называют пользовательскими.

Пользовательский фильтр, который также часто называют *списком доступа24* (access list), предназначен для создания дополнительных барьеров на пути кадров, что позволяет ограничивать доступ определенных групп пользователей к отдельным службам сети. Пользовательский фильтр — это набор условий, которые ограничивают обычную логику передачи кадров коммутаторами.

Наиболее простыми являются пользовательские *фильтры канального уровня* на основе MAC-адресов станций, работающие в коммутаторах локальных сетей. Коммутатор выполняет свои прямые обязанности по сегментации сети, анализируя MAC-адрес, который является частью заголовка кадра канального уровня. Поэтому у администратора есть удобный способ создавать подобные фильтры, прос

24 В данном контексте термин «список доступа» имеет смысл, отличный от терминов «список управления доступом» и «AC1_», которые используются при описании дискреционного способа разграничения прав доступа.

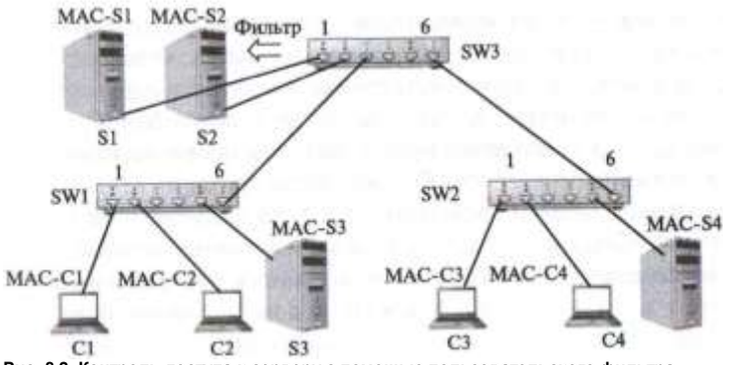


Рис. 8.2. Контроль доступа к серверу с помощью пользовательского фильтра

тавляя некоторые условия в дополнительном поле адресной таблицы, например условие отбрасывать кадры с определенным адресом.

Рассмотрим применение пользовательского фильтра на примере сети, показанной на рис. 8.2.

Пусть мы хотим разрешить доступ к серверу S1 только с компьютеров C1 и C3, кадры от всех остальных компьютеров до этого сервера доходить не должны. Список доступа, который решает эту задачу, может выглядеть так:

10 permit MAC-C1 MAC-SI 20
permit MAC-C3 MAC-SI 30
deny any any

Числа 10, 20 и 30 — это номера строк данного списка. Строки нумеруются с интервалом 10 для того, чтобы в дальнейшем была возможность добавить в этот список другие записи, сохраняя исходную последовательность строк. Первое условие разрешает (permit) передачу кадра, если его адрес источника равен MAC-C1, а адрес назначения — MAC-S1; второе условие делает то же, но для кадра с адресом источника MAC-C3, третье условие запрещает (deny) передачу кадров с любыми (any) адресами.

Для того чтобы список доступа начал работать, его нужно применить к трафику определенного направления на какому-либо порту коммутатора: либо к входящему, либо к исходящему. В нашем примере нам нужно применить список доступа к исходящему трафику порта 1 коммутатора SW3, к которому подключен сервер S1. Коммутатор SW3 перед тем, как передать кадр на порт 1, будет просматривать условия списка доступа по очереди. Если какое-то условие из списка соблюдается, то коммутатор выполняет действие этого условия для обрабатываемого кадра, и на этом применение списка доступа для данного кадра заканчивается.

Поэтому, когда от компьютера C1 приходит кадр, адресованный серверу S1, то соблюдается первое условие списка, которое разрешает передачу кадра, так что коммутатор выполняет стандартное действие по продвижению кадра, и тот доходит до сервера S2. С кадром от компьютера C3 совпадение происходит при проверке второго условия, и он также передается. Однако, когда приходят кадры от других компьютеров, например компьютера C2, то ни первое, ни второе условия не соблюдаются, зато соблюдается третье условие, поэтому кадр не передается, а отбрасывается. Списки доступа коммутаторов не имеют возможности блокировать кадры с широковещательными адресами Ethernet, такие кадры всегда передаются на все порты коммутатора.

Иногда администратору требуется задать более тонкие условия фильтрации, например запретить некоторому пользователю печатать свои документы на сервере печати Windows, находящемуся в чужом сегменте, а остальные ресурсы этого сегмента сделать доступными. Для реализации подобного фильтра нужно запретить передачу кадров, которые удовлетворяют следующим условиям: во-первых, имеют определенный MAC-адрес, во-вторых, содержат в поле данных пакеты SMB, в-третьих, в соответствующем поле этих пакетов в качестве типа сервиса указана печать. Коммутаторы не анализируют протоколы верхних уровней, такие как SMB, поэтому администратору приходится для задания условий фильтрации «вручную» расчитывать координаты поля, по значению которого нужно осуществлять фильтрацию. В качестве признака фильтрации администратор указывает пару «смещение-размер» относительно начала поля данных кадра канального уровня, а затем еще приводит шестнадцатеричное значение этого поля. Сложные условия фильтрации обычно записываются в виде булевых выражений, формируемых с помощью логических операторов AND и OR.

Фильтрация пользовательского трафика ***маршрутизаторами*** аналогична по принципу действия фильтрации, выполняемой коммутаторами локальных сетей. Однако условия фильтрации маршрутизаторов обычно существенно сложнее, и в них учитывается гораздо больше признаков, чем у коммутаторов локальных сетей. Например, это могут быть:

- IP-адреса источника и приемника;
- MAC-адреса источника и приемника;
- идентификатор интерфейса, с которого поступил пакет;
- тип протокола, сообщение которого несет IP-пакет (т. е. TCP, UDP, ICMP или OSPF);.
- номер порта TCP/UDP (т. е. тип протокола прикладного уровня).

При наличии фильтра маршрутизатор сначала проверяет совпадение условия, описанного этим фильтром, с признаками пакета и при положительной проверке выполняет над пакетом ряд нестандартных действий. Например, пакет может быть ***отброшен*** (drop); ***направлен*** к следующему маршрутизатору, отличающемуся от того, который укатан в таблице маршрутизации; ***помечен*** как вероятный кандидат на отбрасывание при возникновении перегрузки. Одним из таких действий может быть и

обычная передача пакета в соответствии с записями таблицы маршрутизации. Подробнее методы фильтрации трафика на сетевом уровне рассмотрены в главе 10.

Определение файервола

Файервол (межсетевой экран, или ***брандмауэр)*** — это комплекс программно-аппаратных средств, осуществляющий информационную защиту одной части компьютерной сети от другой с помощью анализа и фильтрации проходящего между ними трафика.

Исходным значением термина «файервол» (англ. Firewall) является элемент конструкции дома, а именно стена, сделанная из огнеупорного материала и препятствующая распространению огня между частями дома (обычно принадлежащими разным собственникам). Термин «брандмауэр» (нем. Brandmauer) много лет назад пришел в русский язык из немецкого. Изначально он обозначал перегородку в поезде, отделяющую область топки паровоза от пассажирского отделения. Интересно, что немецкие специалисты в области безопасности для обозначения межсетевого экрана используют англоязычный firewall. В русском языке для термина файервол используются и другие транслитерации: файрволл. файрвол, фаервол.

Файервол осуществляет экранирование защищаемого объекта и формирует его внешнее представление. Современные файерволы достигли очень высокого уровня защищенности, удобства использования и администрирования; в сетевой среде они являются первым и весьма мощным рубежом обороны.

Для того чтобы фильтровать трафик, файервол должен иметь по крайней мере два сетевых интерфейса: с внутренней сетью и с внешней сетью (рис. 8.3). Файервол защищает внутреннюю сеть (например, локальную сеть предприятия или, как вырожденный случай, отдельный компьютер пользователя) от угроз, исходящих из внешней сети (мы будем, как правило, подразумевать под такой сетью Интернет). Файервол может также защищать одну внутреннюю сеть предприятия от другой, если в соответствии с принципом минимума полномочий пользователям этих сетей не требуется полный взаимный доступ к ресурсам друг друга.

Для эффективного выполнения файерволом его главной функции — анализа и фильтрации трафика — необходимо, чтобы через него проходил ***весь*** трафик, которым обмениваются узлы защищаемой части сети с узлами Интернета. В том случае, когда сеть связана с

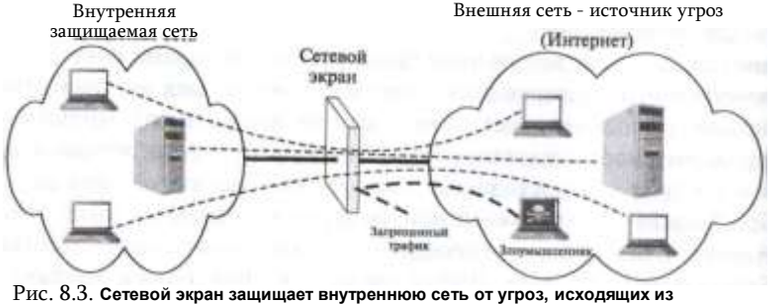


Рис. 8.3. **Сетевой экран защищает внутреннюю сеть от угроз, исходящих из внешней сети**

внешними сетями несколькими линиями связи, ***каждая*** линия связи должна быть защищена файерволом.

Файервол защищает сеть не только от несанкционированного доступа и атак внешних злоумышленников, но от ошибочных действий пользователей защищаемой сети, например таких, как передача во внешнюю сеть конфиденциальной информации.

ПРИМЕР-АНАЛОГИЯ. Функционально сетевой экран можно сравнить с системой безопасности современного аэропорта. Аналогии здесь достаточно очевидные (рис. 8.4) — защищаемой внутренней сети соответствует самолет, а внешняя сеть, из которой приходит потенциально опасный трафик, — реальному миру, откуда прибывают будущие пассажиры самолета, готовящегося к полету, при этом не все они приезжают с чистыми и ясными намерениями.

В потоке пассажиров, постоянно входящих в здание аэропорта, могут встречаться различные злоумышленники. Наиболее зловещие — террористы — пытаются пронести на борт взрывчатку (в мире компьютерных сетей — пакеты, несущие во внутреннюю сеть вирусы, способные «взорвать» серверы и компьютеры пользователей) или оружие для захвата самолета в воздухе (атака по захвату управления удаленным компьютером). Контрабандисты несут с собой незадекларированные ценности (запрещенный контент), а некоторые личности пытаются попасть в самолет по поддельным документам (несанкционированный доступ к внутренним ресурсам сети). Между злоумышленниками и службой безопасности постоянно происходит состязание в коварстве, с одной стороны, и находчивости — с другой. Новые трюки вызывают появление новых способов проверки. Например, пронос взрывчатки в подошве ботинка вызвал к жизни не очень приятную обязательную процедуру прохождения металлоискателя в носках, а использование террористами флаконов для маскировки жидких компонентов бомбы лишило пассажиров возможности брать с собой в кабину шампуни и другие любимые жидкости в больших объемах.

Для того чтобы отфильтровать трафик пассажиров, система безопасности аэропорта пропускает всех пассажиров и их багаж через единственно возможный путь — *зону контроля.* Также поступают при защите сети, направляя весь входящий трафик через *choke point*, образованный межсетевым экраном.

В зоне контроля аэропорта применяются разнообразные средства проверки пассажиров и их багажа. Здесь происходит просвечивание сумок и чемоданов; проход пассажиров через металлодетекторы, а при первом подозрении — вытряхивание всех вещей; дотошная ручная проверка сумок и прощупывание пассажиров. Пассажиры не сколько раз проходят процедуру аутентификации. Лица пассажиров сравниваются с

Рис. **8.4. Зона контроля аэропорта как аналогия сетевого экрана**

фотографиями в паспортах, информация из паспортов сравнивается с компьютерной базой данных. Решение пропускать или не пропускать пассажира может приниматься с учетом их предыстории: например, гражданин РФ может быть остановлен, если в базе данных имеется информация о его невыполненных финансовых обязательствах, иностранцу может быть отказано в прохождении на самолет из-за того, что в предыдущий раз он превысил определенный визой период пребывания в стране, и т. п. Кроме того, процедура проверки может учитывать характеристики не только отдельного пассажира, но и характеристики текущей ситуации, например, если на аэровокзале объявлено состояние повышенной террористической угрозы.

Именно такой способ фильтрации с учетом состояния соединений реализуется в файерволах. Файерволы используют всевозможные средства и методы для противостояния разнообразным угрозам. С помощью паролей и цифровых сертификатов они проверяют аутентичность внешних узлов, пытающихся установить соединения с внутренними; отслеживают логику обмена пакетами для того, чтобы отразить атаки, основанные на искажении этой логики; «просвечивают» содержимое электронных писем и загружаемых

документов, пытаясь блокировать запрещенный контент; сканируют загружаемые программы, проверяя их на наличие известных вирусов. Так же, как и в зоне контроля аэропорта, здесь постоянно идет соревнование между хакерами, все время изобретающими новые методы атак, и разработчиками сетевых экранов, старающихся эти атаки обнаружить и пресечь.

Основными функциями файервола являются:

- ***фильтрация трафика в целях*** защиты внутренних ресурсов сети;
- ***аудит***— файервол должен фиксировать все события, связанные с обнаружением и блокировкой подозрительных пакетов.

Наряду с этими двумя базовыми функциями на сетевой экран могут быть возложены и другие вспомогательные функции защиты, в частности:

- антивирусная защита;
- шифрование трафика;
- логическое посредничество между внутренними клиентами и внешними серверами (функция прокси-сервера);
- фильтрация сообщений по содержимому, включая типы передаваемых файлов, имена DNS и ключевые слова;
- предупреждение и обнаружение вторжений и сетевых атак;
- функции VPN;
- трансляция сетевых адресов (NAT).

Как можно заметить, большинство из перечисленных функций часто реализуются в виде отдельных продуктов или в составе систем защиты других типов. Так, функции пакетной фильтрации встроены практически во все маршрутизаторы, задача обнаружения вирусов решается множеством разнообразных программ, шифрование трафика — неотъемлемый элемент технологий защищенных каналов и т. д., и т. п. Прокси-серверы часто поставляются в виде приложений, более того, они сами часто интегрируют в себе многие функции, свойственные сетевым экранам, такие, например, как фильтрация по содержимому (контенту) или трансляция сетевых адресов.

Отсюда возникают сложности при определении понятия «файервол». Например, довольно распространено мнение, что файервол — это пограничное устройство, выполняющее пакетную фильтрацию (т. е. маршрутизатор), а прокси-сервер — это совершенно отличный от файервола инструмент защиты. Другие настаивают, что прокси-сервер является непременным и неотъемлемым атрибутом любого файервола. Третьи считают, что файерволом может быть названо только такое программное или аппаратное устройство, которое способно отслеживать состояние потока пакетов в рамках соединения. Мы же в этой книге будем придерживаться широко распространенной точки зрения о том, что файервол— это программно-аппаратный комплекс, выполняющий разнообразные функции по защите внутренней сети, набор которых может меняться в зависимости от типа, модели и конкретной конфигурации сетевого экрана, при этом минимальный набор функций должен включать фильтрацию трафика для предотв ращения сетевых атак и аудит событий, связанных с фильтрацией.

Типы файерволов

Одним из признаков классификации файервола служит способ его

реализации:

- ***программный:*** файервол реализован как программная система, работающая под управлением универсальной ОС, такой как Microsoft Windows, Linux или Mac OS (возможно, имеющая версии для нескольких универсальных ОС);
- ***аппаратный:*** файервол реализован как набор дополнительных функций фильтрации маршрутизатора (реже — Ethernet-комму- татора);
- ***программно-аппаратный:*** файервол включает как программную систему, так и специализированный сервер, операционная система которого и аппаратура имеют конфигурацию и настройки, оптимизированные для работы файервола. Чаще всего в качестве такой специализированной платформы используется универсальная ОС с набором специфических настроек, обеспечивающих максимальный уровень безопасности, а также сервер, сертифицированный для работы с программным обеспечением файервола. Файерволы различают также по масштабу защищаемой сети:
- ***персональный*** файервол, защищающий один компьютер пользователя или же его домашнюю сеть;
- файервол ***масштаба группы или отдела',***
- ***корпоративный*** файервол, защищающий периметр сети кампуса.

Классификация файерволов по функциональным признакам делит их на типы в зависимости:

- от ***способа фильтрации,*** здесь различаются файерволы без запоминания состояния (stateless) и файерволы с запоминанием состояния (stateful); а также
- от того, на каком ***уровне модели OSI*** они анализируют и фильтруют трафик, по этому признаку различают файерволы сетевого, сеансового и прикладного уровней.

Как правило, имеется корреляция между различными характеристиками файерволов: например, корпоративной файервол скорее всего представляет собой программно-аппаратный комплекс, работающий на всех уровнях стека TCP/IP и поддерживающий фильтрацию с запоминанием состояния, что обеспечивает наиболее полную защиту (и наиболее затратную), а персональный файервол может быть программой, работающей в среде определенной операционной системы и входящая в стандатный комплект поставки этой ОС (например, утилита Linux iptables или Microsoft Windows Firewall).

Режим работы (stateful или stateless) файервола обычно связан

с уровнем модели OSI, на котором он работает25. Так, файерволы, работающие на сеансовом и прикладном уровнях, чаще относятся к разряду файерволов с запоминанием состояния, а более простые файерволы сетевого уровня — без запоминания.

- *Файерволы без запоминания состояния* (***stateless***) выполняют фильтрацию на основе статических правил, при этом не отслеживаются состояния соединений (сеансов), такой режим называют также *stateless packet inspection.*
- *Файерволы с запоминанием состояния* (***stateful)*** принимают решения динамически с учетом текущего состояния сеанса и его предыстории, такой режим называют *stateful packet inspection.* Файерволы с запоминанием состояния для каждого сеанса, который удовлетворяет условиям некоторого активного правила, создают динамическую структуру данных в специальной таблице состояний файервола. После прихода очередного пакета контролируемого сеанса, состояние сеанса корректируется и принимается решение о выполнении заданного действия с пакетом — пропускать или отбрасывать.

Отслеживание состояний сессий протоколов требует использования больших объемов ресурсов, именно поэтому реализация файервола с запоминанием состояния требует использования программноаппаратных комплексов с большими объемами оперативной памяти для хранения таблицы состояний сеансов и процессоров высокого быстродействия для обработки данных поступающих пакетов в реальном времени. Если же ресурсы такого файервола оказываются недостаточными, то он вместо пользы может принести вред, когда внутренние серверы оказываются недоступными не из-за атак на них, а из-за заторов трафика в интерфейсах файервола.

Файерволы с запоминанием состояния относятся к высшему классу файерволов, помимо высокой производительности они оснащаются удобным графическим интерфейсом, позволяющим администратору не запоминать опции командного языка, как при использовании маршрутизаторов, а переводить правила политики безопасности предприятия в правила файервола с помощью интуитивно понятных директив.

Наиболее употребительной характеристикой файервола является ***уровень протоколов,*** на котором он работает (или же самый высокий из всех уровней, если он работает на нескольких уровнях), поэтому далее мы приводим типичные сочетания характеристик файерволов, упорядоченные по уровням.

К *файерволам канального уровня* могут быть условно отнесены управляемые коммутаторы, обладающие расширенным набором функций, в том числе возможностью фильтрации кадров канального уровня на основе

25 Выражение «файервол работает на уровне протокола N» означает, что данное устройство способно анализировать поля заголовков сообщена данного протокола.

задаваемых администратором списков доступа.

Файерволы сетевого уровня, называемые также *файерволами с фильтрацией пакетов* (packet filtering firewall), в полном соответствии со своим названием решают задачу фильтрации пакетов по IP- адресам (как источника, так и назначения), а также по значению поля «протокол верхнего уровня» — в пакет сетевого уровня могут быть вложены сообщения протоколов TCP, UDP, ICMP и др. Более того, несмотря на свое название, такие файерволы работают и на более высоком, транспортном уровне, т. е. на уровне портов TCP и UDP, но только на основе ***статических*** правил, при которых не отслеживаются состояния соединений, т. е. в режиме ***stateless packet inspection.*** Поэтому с помощью файервола сетевого уровня можно заблокировать доступ к определенному приложению, запретив прохождение пакетов с определенными номерами портов TCP или UDP, но нельзя защитить сеть от искаженной сессии TCP или HTTP, потому что это требует отслеживания последовательности шагов в сессии и, следовательно, запоминания состояния сессии — а этого файерволы сетевого уровня делать не умеют.

Этому типу файерволов соответствуют маршрутизаторы, поддерживающие пользовательские фильтры, и программные персональные файерволы операционных систем. Опытный администратор может задать достаточно изощренные правила фильтрации, учитывающие многие требования, касающиеся защиты ресурсов внутренней сети, тем не менее, этот тип сетевых экранов уступает по степени защиты другим типам. Преимуществами файерволов сетевого уровня являются простота, невысокая стоимость и минимальное влияние на производительность сети (т. е. их дополнительная работа по фильтрации трафика не замедляет маршрутизацию пакетов между двумя сетями).

Файерволы сеансового уровня отслеживают состояние соединений за счет запоминания состояний сеансов протоколов, т. е., другими словами, выполняют операцию ***stateful packet inspection*** на уровнях, ниже прикладного.

Прежде всего имеется в виду состояние сеанса протокола TCP, его начальной трехшаговой процедуры установления соединения. ***Отслеживание состояний соединений*** заключается в том, что сетевой экран проверяет, насколько соответствует последовательность обмена сообщениями контролируемому протоколу. То есть, например, если клиент посылает TCP-сообщение *SYN,* запрашивающее TCP- соединение, сервер должен отвечать TCP-сообщением *ACK SYN,* а не посылать в ответ, например, свой TCP-запрос *SYN.* После того
как сетевой экран установил допустимость TCP-соединения, он начинает работать простым передаточным звеном между клиентом и сервером. Таким образом, файервол сеансового уровня может защитить сеть от различных

типов TCP-атак, в которых нарушается логика установления соединения — SYN Flood, RST, ACK Flood.

Для того чтобы контролировать процесс установления соединения, сетевой экран должен фиксировать для себя текущее состояние соединения, т. е. *запоминать,* какое последнее сообщение отправил клиент и какое сообщение он ожидает получить. Поддержка запоминания состояния сессий протоколов позволяет этому типу файервола защищать сеть не только от атак на протокол TCP, но и других видов атак, например от атаки Ping flood, которую можно распознать по слишком маленькому интервалу между эхо-запросами от одного и того же источника (т. е. необходимо установить предельно допустимый минимальный интервал между эхо-запросами, а затем запоминать время прихода очередного запроса и фильтровать его, если оно оказывается меньше предельного). Таким образом, запоминание состояния сессий может обобщаться и на протоколы, работающие без установления соединения (ICMP, UDP, DNS), но атаки на которые можно распознать и остановить, анализируя не отдельные пакеты, а их последовательность. А это означает, что в отличие сетевых файерволов, файерволы сеансового типа способны защитить сеть от некоторых видов DoS-атак, даже если они и не используют протокол TCP.

Сетевые экраны прикладного уровня способны интерпретировать, анализировать и контролировать содержимое сообщений, которыми обмениваются приложения. К этому уровню относят прокси-серверы, о которых мы будем говорить подробнее далее. Они также работают на основе фильтрации с запоминанием состояния, но анализируют состояния не только протоколов нижних уровней вплоть до транспортного, но и прикладного уровня, таких как SSH, FITTP, FTP, SQL.

Особым типом файерволов это уровня является *прокси-сервер,* который перехватывает запросы клиентов к внешним серверам с тем, чтобы потом отправить их от своего имени. Этот тип сетевых экранов обеспечивает самый высокий уровень защиты, хотя и имеет свои недостатки, например требует больших вычислительных затрат. Кроме того, прокси-серверы могут скрывать адрес «доверившегося» ему клиента, что снижает эффективность других средств защиты.

Системы обнаружения вторжений

Типы систем обнаружения вторжений

Система обнаружения вторжений (*Intrusion Detection* ***System*** *IDS*) — это программное или аппаратное средство, предназначении для предупреждения, выявления и протоколирования некоторых типов сетевых атак.

В отличие от файерволов и прокси-серверов, которые строят защиту сети исключительно на основе анализа сетевого трафика, системы обнаружения вторжений учитывают в своей работе различные подозрительные события, происходящие в системе.

Существуют ситуации, когда сетевой экран оказывается проницаемым для злоумышленника, например когда атака идет через туннель VPN из взломанной сети или инициатором атаки является пользователь внутренней сети и т. п. И дело здесь не в плохой конфигурации межсетевого экрана, а в самом принципе его работы. Экран, несмотря на то что обладает памятью и анализирует последовательность событий, конфигурируется на блокирование трафика с заранее предсказуемыми признаками, например по IP-адресам или протоколам. Так что факт взлома внешней сети, с которой у него был установлен защищенный канал и которая до сих пор вела себя вполне корректно, в правилах экрана отразить нельзя, точно так же, как и неожиданную попытку легального внутреннего пользователя скопировать файл с паролями или повысить уровень своих привилегий. Подобные подозрительные действия может обнаружить только система со встроенными агентами во многих точках сети, причем она должна следить не только за трафиком, но и за обращениями к критически важным ресурсам операционных систем отдельных компьютеров, а также иметь информацию о перечне подозрительных действий (сигнатур атак) пользователей. Таковой и является система обнаружения вторжений. Она не дублирует действия файервола, а дополняет их, производя, кроме того, автоматический анализ всех журналов событий, имеющихся у сетевых устройств и средств защиты, чтобы попытаться найти следы атаки, если ее не удалось зафиксировать в реальном времени.

Главным же отличием IDS от файерволов является то, что в обязанности IDS не входит блокировка подозрительного трафика. IDS только пытается выявить подозрительную активность и поднять тревогу — обычно путем предупреждения администратора сети электронным сообщением. Кроме поднятия тревоги, IDS протоколирует подозрительные пакеты, помещая их в журнал.

Существуют также *системы предупреждения вторжений* (*Intrusion Prevention Systems, ЮР),* которые выполняют автоматические действия по прекращению атаки в случае ее обнаружения. Часто такие системы перепоручают эту работу файерволу, передавая ему новое правило для блокировки подозрительного трафика.

IDS сетевого уровня выполняет работу по выявлению угроз на основе анализа трафика, проходящего через один или несколько локальных сегментов корпоративной сети. IDS анализирует все поля

пакетов, в том числе и поле данных, которое переносит информацию приложений, поэтому ее возможности по обнаружению подозрительной активности гораздо больше, чем у систем, которым доступны только поля заголовков протоколов Ethenet и IP.

Кроме IDS сетевого уровня, существуют *IDS уровня хоста,* которые анализируют события, происходящие в операционной системе и приложениях.

Функциональная схема IDS

Типовая функциональная архитектура IDS показана на рис. 8.5 [RFC 4766].

Источниками данных для сетевой IDS являются маршрутизаторы, коммутаторы и хосты локальной сети, словом, все элементы сети, которые передают, генерируют и принимают трафик.

Датчик копирует пакеты, циркулирующие в сети, и передает их анализатору для выявления подозрительной активности. Датчик может представлять собой отдельный компьютер, подключенный к зеркализованному порту коммутатора (как показано на рис. 8.5), или же это может быть программная компонента маршрутизатора, которая имеет доступ к пакетам, буферизуемым в его интерфейсах. Датчик может осуществлять первичную фильтрацию пакетов, отбирая только те пакеты, которые удовлетворяют некоторым очевидным критериям, например те, которые направлены к публичном веб-сервисам, которые атакуются наиболее часто.

Анализатор является «мозгом» IDS, он получает данные от датчиков и проверяет их на наличие угроз и подозрительной активности в сети. Анализатор работает на основе правил, составленных ***администратором*** системы безопасности предприятия в соответствии с политикой безопасности. При выполнении условия одного из правил анализатор вырабатывает сообщение «тревога» и передает ее *менеджеру* IDS — программной компоненте, которая хранит конфигурацию

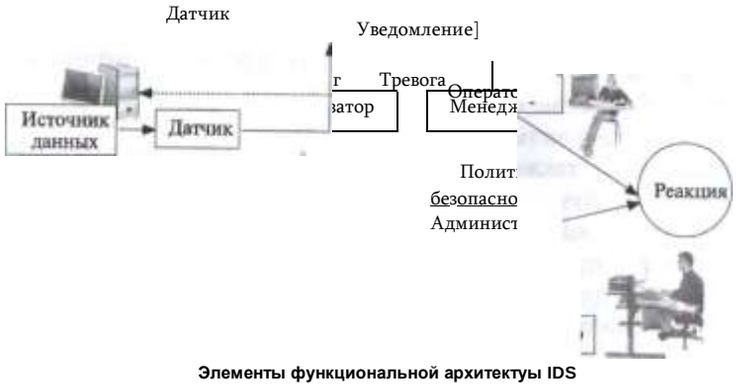


Рис. 8.6. Элементы функциональной архитектуы IDS

IDS и поддерживает удобный интерфейс с оператором IDS. Менеджер IDS оповещает оператора IDS о тревоге в виде некоторого уведомления, привлекающего внимание, например текстовой строки на экране с мерцающим символом, звукового сигнала, продублированного электронным письмом, и т. п.

Оператор IDS на основе данных уведомления принимает решение о реакции сети на подозрительную активность — это может быть отключение сетевого интерфейса, через который поступает подозрительный трафик, изменение правил файервола для блокировки определенных пакетов или же игнорирование уведомления, если оператор считает, что вероятность вторжения очень мала. В любом случае все данные о потенциальном вторжении протоколируются в журнале менеджера и могут быть использованы впоследствии для повторного анализа ситуации. Если же IDS выполняет также функции IDP, то менеджер может автоматически передать команды на маршрутизатор или файервол для блокировки подозрительного трафика.

Описанная выше архитектура является функциональной, в реальной IDS эти функции не обязательно реализуются в отдельных блоках или модулях системы. В минимальном варианте все функции IDS могут быть сосредоточены в программном обеспечении единственного компьютера, сетевой адаптер которого выполняет роль датчика за счет того, что он присоединен к зеркализованному порту коммутатора или маршрутизатора.

Более масштабируемой является реализация IDS с несколькими датчиками, подключенными к различным сегментам сети, и посылающие захваченный трафик центральному анализатору. Такие датчики могут быть дополнительным программным обеспечением маршрутизатора или коммутатора, или же представлять собой отдельные аппаратные устройства.

Правила обнаружения атак

В IDS для обнаружения вторжений применяются правила нескольких типов правил.

Правила, основанные на подписи атаки ***(signature rules)*** используют характерную для атаки последовательность символов в данных пакета. Например, правило может диктовать поиск строки 'user root' в полях пакета ftp — как известно, этот протокол передает пароли пользователей в открытом виде и использование его суперпользователем root считается грубым нарушением политики безопасности предприятия, так что IDS должна отслеживать такие случаи. Чаще всего подписи атак относятся к прикладным протоколам, для обнаружения вторжения на транспортном уровне они менее пригодны. Для эффективной работы IDS должен иметь обширную постоянно пополняемую базу данных подписей атак.

Правила, основанные на анализе протоколов ***(protocol rules)*** контролируют логику работы протокола и фиксируют отклонения от него. Так как каждый протокол обладает специфической логикой, то IDS обычно имеет библиотеку программных модулей, каждый из которых может выполнять анализ поведения определенного протокола. Правила анализа протоколов написать существенно сложнее, чем правила анализа подписи атаки, так как для этого нужно хорошо знать логику протокола и возможные попытки ее измененияе. Реализация правил анализа протоколов требует большого быстродействия IDS, в противном случае процедура обнаружения вторжений может значительно замедлиться и IDS перестанет быть системой реального времени.

Правила, основанные на статистических аномалиях трафика проверяют такие характеристики трафика, как ***Top 10 sessions, Top 10 Data,*** которые будут подробнее рассмотрены в главе 10 при описании технологии анализа данных NetFlow. В принципе любая статистика активности пользователей корпоративной сети может использоваться для этой цели. Например, если 10 % трафика пользователей отдела планирования всегда направлено к серверу базы данных финансового отдела, то появление пользователя, у которого 90 % трафика идет на работу с этим сервером, может вызвать подозрение — возможно компьютер этого пользователя захвачен злоумышленником, удаленно пытающимся похитить чувствительные финансовые данные предприятия.

Вопросы к главе 8

1. Какие функции системы безопасности из перечисленных направлены на обеспечение подотчетности?

а) аутентификация;
б) авторизация;
в) аудит;
г) протоколирование событий;
д) мониторинг;
е) журнализация событий.

2. К какому типу средств вы бы отнесли аудит?

а) к средствам устрашения (сдерживания);
б) к превентивным средствам обеспечения безопасности;
в) к воспитательным средствам;
г) к средствам обучения;
д) к средствам расследования.

3. Что дает сегментация сети?

а) повышение производительности;
б) упрощение администрирования;
в) повышение безопасности.

4. Пользовательский фильтр:

а) устанавливается в соответствии с информацией, содержащейся в адресной таблице устройства;

б) применяется для фильтрации трафика, порождаемого пользовательскими процессами;

в) используется для фильтрации трафика, поступающего в компьютеры пользователей;

г) устанавливается администратором в соответствии с дополнительными ограничениями, накладываемыми на логику работы устройства.

5. При фильтрации пакет может быть:

а) передан обычным образом, без изменения маршрута;
б) отброшен;
в) переправлен назад отправителю;
г) помечен некоторым маркером;
д) передан по маршруту, отличающемуся от указанного в таблице маршрутизации;
е) сохранен в буфере до того, пока администратор не решит, что с ним делать.

6. Комплекс программно-аппаратных средств, осуществляющий информационную защиту одной части компьютерной сети от другой путем анализа и фильтрации проходящего между ними трафика, называют:

а) межсетевой экран;
б) файервол;
в) брандмауер;
г) маршрутизатор, поддерживающий пользовательские фильтры.

7. Какими признаками скорее всего будет обладать корпоративный файервол?

а) программно-аппаратная реализация;
б) выполняет прокси-сервис;
в) работает на всех уровнях стека транспортных протоколов выше сетевого;
г) поддерживает фильтрацию с запоминанием (stateful);
д) выполняет шифрование.

8. Файерволы сеансового уровня способны выполнять следующие действия:

а) определять, насколько последовательность обмена сообщениями соответствует контролируемому протоколу;
б) запоминать в рамках одного сеанса связи, какое последнее сообщение отправил клиент и какое сообщение он ожидает получить;
в) анализировать интервал между эхо-запросами от одного и того же источника;
г) контролировать некоторые протоколы, работающие без установления соединения.

9. В число основных функций IDS входят:

а) блокировка подозрительного трафика;
б) выполнение автоматических действий по прекращению атаки;
в) анализ всех полей пакетов, в том числе поля данных, переносящего информацию приложений;
г) протоколирование событий, относящихся к безопасности;
д) анализ событий, связанных с безопасностью ОС и приложений.

10. Какие из перечисленных ниже атак могут быть обнаружены файерволом?

а) внедрение вируса;
б) атака, использующая для передачи своих пакетов защищенный канал IPsec;
в) попытка несанкционированного доступа к данным сети со стороны легального пользователя внутренней сети;
г) атака отказ в обслуживании.

В этой части мы рассмотрим уязвимости транспортной инфраструктуры

Часть III

Защита транспортной инфраструктуры сети

сети, атаки на нее и методы ее защиты. В транспортную инфраструктуру включаются все промежуточные узлы сети, представляющие собой маршрутизаторы, коммутаторы, а также транспортные средства операционных систем серверов и пользовательских компьютеров, установленных в конечных узлах сети.

Атаки на транспортную инфраструктуру сети направлены на искажение, разрушение или несанкционированный доступ к данным, которыми компьютеры обмениваются между собой по сети. Сетевой обмен данными является сегодня неотъемлемой частью большинства информационных сервисов, любые нарушения нормального хода этого процесса приводят к деградации или полной остановке сервиса. Все мы, к сожалению, знакомы с такими ситуациями, когда какой-то сайт становится недоступным из-за распределенной атаки «Отказ в обслуживании» (DDoS-атаки), а нужное электронное письмо оказывается погребено под горами спама. Поэтому проблемы защиты транспортной инфраструктуры сети являются сегодня не менее важными, чем проблемы защиты программного обеспечения компьютера, которым традиционно уделяется больше внимания.

9 Транспортная инфраструктура и ее уязвимости

Современная транспортная инфраструктура сети представляет собой составную IP-сеть, включающую сети *Интернет-провайдеров* ***(Internet Service Provider, ISP)*** и их клиентов, как корпоративных, так и индивидуальных. Фрагмент инфраструктуры показан на рис. 9.1.

Здесь мы видим сети четырёх Интернет-провайдеров (ISP A, B, C и D), при этом сеть провайдера ISP A и его клиенты показаны более подробно. Провайдер ISP A имеет одного крупного корпоративного клиента, один дата-центр, а также некоторое количество индивидуальных клиентов (показаны как компьютеры на границе сети

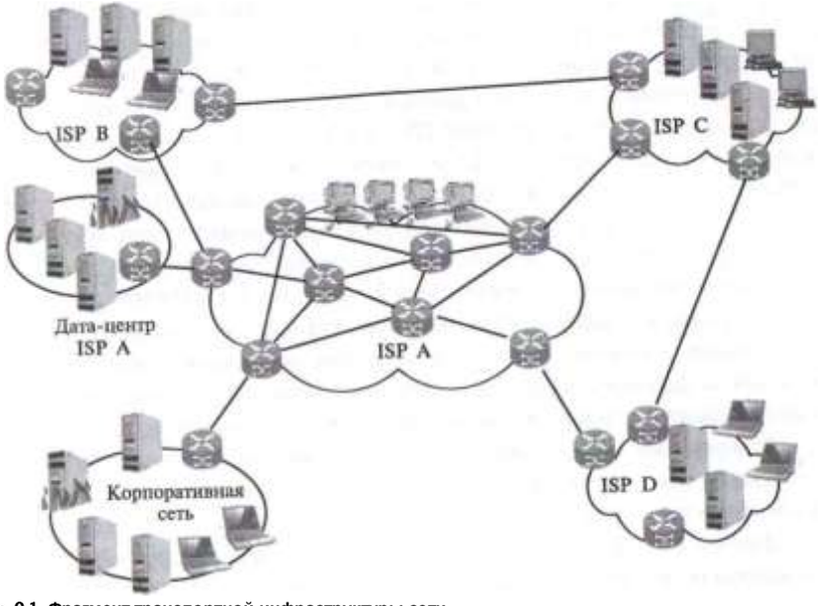


Рис. 9.1. Фрагмент транспортной инфраструктуры сети

провайдера). В дата-центре находятся серверы провайдера, часть из которых предназначена для представления разнообразных услуг клиентам, таких как хостинг виртуальных машин, веб-сайтов, приложений, облачные сервисы и т. д., а часть выполняет служебные функции по поддержанию сети провайдера, например на них может работать *служба DNS (Domain Name System*) или же *система управления сетью NMS (Network Management System).* Корпоративная сеть предприятия включает принадлежащие этой организации серверы и клиентские компьютеры.

Показанная на рис. 9.1 схема транспортной инфраструктуры составной сети очень упрощена, здесь не отражены многие ее элементы, например иерархия провайдеров Интернет и внутренняя структура корпоративной сети, однако такой уровень детализации вполне достаточен для иллюстрации основных атак на транспортную инфраструктуру и способов их отражения, рассматриваемых ниже в этой главе.

Компьютерная сеть основана на передаче пакетов данных между ее конечными узлами — серверами и клиентскими компьютерами. Для того чтобы такая передача стала возможна, все *конечные узлы* сети соединены друг с другом *линиями связи* и промежуточными узлами — *маршрутизаторами.*

Конечные узлы, линии связи и маршрутизаторы образуют транспортную инфраструктуру любой сети, в том числе и сети сетей — Интернет. Очевидно, что если транспортная инфраструктура сети работает со сбоями или же вообще не работает, то компьютеры этой сети перестают «видеть» друг друга, превращаясь в изолированные центры хранения и обработки данных — ситуация вполне приемлемая для большинства пользователей 70-х и 80-х годов, но совершенно нетерпимая сегодня (хотя иногда и полезная, как некоторые яды в небольших дозах). Естественно, что *транспортная инфраструктура сети является заманчивой мишенью* для злоумышленников, ведь нарушив ее работу на некотором участке пути к серверу-жертве, они лишают доступа к нему всех или значительной части пользователей.

Нарушения в работе транспортной инфраструктуры могут быть различными, например полное прекращение достижимости сети, подсети или отдельного узла, замедление работы части сети или узла, перенаправление трафика к ложным узлам назначения.

Атакуемыми объектами транспортной инфраструктуры сети являются как конечные узлы, так и маршрутизаторы (на рис. 9.1 атакуемые объекты обозначены языками пламени).

Для того чтобы атаковать транспортную инфраструктуру, зло-

умышленник использует особенности протоколов, обеспечивающих передачу пакетов в сети. Поэтому для распознавания этого вида атак

и борьбы с ними необходимо знать эти протоколы. Сегодня все се- (свые узлы, как конечные (серверы, клиентские компьютеры), так и промежуточные (маршрутизаторы), относящиеся как к сетям провайдеров, так и к корпоративным сетям, поддерживают протоколы ***стека ГСР/1Р.*** Именно протоколы этого стека обеспечивают связность отдельных сетей в единое

целое — Интернет. Краткие сведения о стеке протоколов TCP/IP приведены Приложении 2.

Основой архитектуры TCP/IP является протокол ***IP*** (*Internet Protocol).* Он обеспечивает перемещение пакетов между *узлами* составной сети, образованной объединением множества сетей. Протокол IP является тем клеем, который соединяет воедино в общем случае разнородные сети, каждая из которых может работать на основе самых разных транспортных технологий нижележащих уровней: Ethernet, SDH, MPLS, OTN, DWDM.

Протокол IP работает как на конечных узлах сети, так и на маршрутизаторах, основным назначением которых является чтение адреса назначения пакета и передача пакета следующему маршрутизатору, который находится на пути следования пакета.

Протокол IP — это *дейтаграммный* протокол, т. е. протокол, работающий без установления соединени. Это означает, что любой узел Интернета может направлять пакеты любому другому узлу без предварительной процедуры получения разрешения «поговорить». Можно сказать, что протокол IP является краеугольным камнем «демократии» Интернета — все узлы в Интернет равны, каждый может общаться с каждым, при этом отказаться от такого общения с помощью средств самого протокола IP нельзя, если уж кто-то отправил пакет, предназначенный для вашего компьютера, то он до него скорее всего дойдет, если только вы или ваш провайдер заранее не предприняли специальных мер безопасности и по какой-то причине не заблокировали прием пакетов от этого конкретного отправителя.

И вот это свойство IP — обеспечение взаимодействия каждого узла с каждым без предварительного установления соединения — и представляют одну из *главных уязвимостей* IP-сетей (и, естественно, Интернета). Действительно, в IP-сети любой злоумышленник в общем случае имеет доступ к любому узлу сети, а значит, имеет возможность организовать атаку. Все методы предотвращения атак на транспортную инфраструктуру сети основаны на ограничении этого свойства, в частности, именно этим и занимаются файерволы.

Нужно заметить, что не все транспортные протоколы являются такими демократичными, например существуют такие протоколы, как X.25 (сейчас забытый, но в свое время более популярный, чем IP), frame relay, ATM, MPLS, которые включают процедуру предварительного установления соединения между источником и узлом назначения. Такое соединение, называемое постоянным, устанавливается администратором сети, а не пользователем, поэтому риск атаки здесь существенно меньше. Можно сказать, что открытость и демократичность протокола IP были одной из причин успеха Интернета, но, с другой стороны, они создали благоприятную среду для злоумышленников всех сортов.

Для того чтобы протокол IP мог доставлять пакеты любому узлу составной сети, эти узлы должны иметь адреса, уникальные в пределах данной составной сети. Такие адреса называются *IP-адресами.* Блоки IP-

адресов централизованно выделяются каждому провайдеру Интернета, который, в свою очередь, распределяет их между своими клиентами, что обеспечивает уникальность IP-адресов узлов. В результате каждый узел составной сети, который намерен обмениваться данными с другими узлами составной сети, имеет IP-адрес наряду с адресом, назначенным ему на канальном уровне (например, MAC- адресом Ethernet). И когда клиентский компьютер, имеющий, например, IP-адрес 195.155.201.17, направляет пакет серверу с адресом 173.194.67.94, то сеть однозначно знает, какому серверу из миллионов серверов, подключенных к Интернету, нужно доставить этот пакет и какому клиенту вернуть ответ.

Протоколы транспортного уровня ***TCP*** *(Transmission Control Protocol)* и ***UDP*** (*User Datagram Protocol)* работают «над» протоколом IP, используя его как инструмент для решения своих задач — передачи данных между *приложениями* двух взаимодействующих по сети конечных узлов. Программные модули, реализующие протоколы TCP и UDP, работают как на конечных узлах (хостах в терминологии TCP/IP), так и на маршрутизаторах. Однако в последнем случае они выполняют только вспомогательные функции, поддерживая удалённый доступ администратора к маршрутизатору для его конфигурирования и обслуживания. Поэтому можно сказать, что протоколы TCP и UDP — это протоколы конечных узлов, так как даже в том случае, когда они работают на маршрутизаторе, последний выступает как конечный узел в паре «компьютер администратора» — «конфигурируемый маршрутизатор».

Рассмотрим упрощенно работу протокола TCP. Приложение-отправитель направляет протоколу TCP поток байтов, которые необходимо передать приложению-получателю, работающему на некотором другом конечном узле сети. Протокол TCP делит поступающий поток данных на сегменты и передает «вниз» протоколу IP. Протокол IP узла отправителя упаковывает сегменты в пакеты и отправляет на ближайший маршрутизатор, протокол IP которого передает их дальше. Пакеты перемещаются по сети от одного маршрутизатора к другому. Когда пакеты достигают узел назначения, протокол IP конечного учла передает пакеты «наверх» протоколу TCP, который организует передачу данных приложению-получателю.

Протокол TCP обеспечивает гарантированную доставку данных, для этого в нем предусмотрена процедура установления *логического соединения* (часто называемого также *сессией)* между протоколами I CP, работающими на узлах отправителя и получателя. В рамках соединения протокол TCP нумерует пакеты, отсылает квитанции под- 1верждения их приема, в случае потери данных организует повторные передачи, распознает и уничтожает дубликаты, и после получения всех сегментов передает приложению-получателю данные в том порядке, в котором они были отправлены.

UDP является дейтаграммным протоколом и, следовательно, не может гарантировать доставку данных. Он используется в тех случаях, когда задача надежного обмена данными либо вообще не ставится, либо решается средствами более высокого уровня — прикладным уровнем или пользовательскими приложениями.

Для того, чтобы данные попали именно тому приложению, которому они предназначаются, в заголовках сегментов TCP и UDP имеются два поля: ***порт источника*** и ***порт назначения.*** Эти порты являются программными точками входа-выхода операционной системы; с каждым портом связана какая-то прикладная программа, которая обменивается данными по сети. Порты идентифицируются номерами; при этом за первыми 1023 номерами централизованно закреплены определенные сетевые приложения: например, порт 21 присвоен сервису передачи файлов ftp, а порт 22 — протоколу удаленного доступа ssh. Как видно из этого описания, хотя сами протоколы TCP и UDP относятся к транспортной инфраструктуре сети, их заголовки несут информацию о приложениях, которые этими протоколами пользуются, поэтому порты TCP и UDP всегда фигурируют в описаниях конфигураций файерволов для защиты от атак на прикладном уровне — мы рассмотрим такие конфигурации в следующей главе, посвященной приложениям.

Протокол TCP является более уязвимым для атак, чем протокол UDP, из-за того, что он включает процедуру установления логического соединения между конечными узлами. Именно эту процедуру пытаются использовать злоумышленники для организации DOS/DDOS- атак, направленных на исчерпание ресурсов транспортной подсистемы конечного узла.

Для того чтобы знать, какому следующему маршрутизатору передать IP пакет, чтобы он дошел до адресата, маршрутизаторы строят и используют ***таблицы маршрутизации.*** Таблица маршрутизации состоит из записей, в которых IP-адресу назначения (или же диапазону адресов) ставится в соответствие выходной порт маршрутизатора.

Таблицы маршрутизации могут быть созданы вручную, либо построены автоматически с помощью ***протоколов маршрутизации.***

Провайдеры Интернета используют два типа протоколов маршрутизации: так называемые *внутренние протоколы маршрутизации* (Interior Gateway Protocol, IGP), к которым относятся протоколы ***OSPF*** и ***IS-IS,*** и *внешние протоколы маршрутизации* (Exterior Gateway Protocol, EGP), сегодня представленные только одним протоколом ***BGP.*** Протоколы IGP применяются для построения таблиц маршрутизации, отражающих пути пакетов внутри сети провайдера (называемой в этом случае автономной системой), а протоколы EGP используются для построения таблиц маршрутизации, отражающих маршруты пакетов между сетями провайдеров.

Протоколы маршрутизации являются важным элементом транспортной инфраструктуры сети и одним из ее уязвимых мест — ведь нарушив нормальный процесс построения таблицы маршрутизации, можно полностью нарушить транспортировку пакетов через сеть. Несколько печально известных инцидентов такого рода уже случилось в истории Интернета. Так, в 1997 году из-за ошибки в конфигурировании протокола BGP *одного* маршрутизатора таблицы маршрутизации большого числа провайдеров оказались искажены и нормальная работа Интернета для их пользователей оказалась прерванной на несколько часов.

Протокол ***ICMP*** (*Internet Control Message Protocol),* играет роль протокола обратной связи для узлов сети при передаче пакетов. При этом он не предназначен для исправления возникших при передаче пакета проблем: если пакет потерян, ICMP не может послать его заново. Задача ICMP другая — он является *средством оповещения* отправителя о «несчастных случаях», произошедших с его пакетами на пути следования или по прибытии на узел назначения. Если, например, маршрутизатор не знает, как передать пакет в сеть назначения из-за того, что адрес этой сети не содержится в его таблице маршрутизации, то он отправляет узлу, пославшему пакет, сообщение «Узел назначения недостижим» с кодом «Сеть назначения неизвестна». Протокол ICMP может также корректировать поведение конечного узла или маршрутизатора, отправив сообщение «Перенаправление маршрута», считая, что выбранный маршрут не является рациональным. Очевидно, что злоумышленник может использовать сообщения протокола ICMP, чтобы «скорректировать» маршрутизацию пакетов в сети выгодным для себя образом.

Протокол ***DNS*** (*Domain Name System)* выполняет очень важную для пользователей Интернета функцию — он отображает понятные пользователю символьные (называемые также доменными) имена узлов сети, такие как www.google.com, в IP-адреса этих узлов. Дело в **нім, что** маршрутизаторы передают пакеты на основании IP-адресов, і имвольные имена он не понимают, в IP-пакетах использовать их в качестве адреса назначения нельзя. Служба DNS нужна для того, что- fii.i преобразовывать символьные имена, вводимые в окно браузера ими в поле адреса электронной почты, в IP-адреса, понятные маршрутизаторам. Это преобразование инициирует узел-отправитель, точ- 1114' — компонента его операционной системы, называемая ***резольвером*** Запрос резольвера идет к одному из известных конечному узлу DNS-серверов, который сам или же с помощью других DNS-серверов, работающих в Интернете, находит искомое соответствие и возвращаем IP-адрес конечному узлу. После этого конечный узел формирует II' пакет с найденным IP-адресом назначения и отправляет его в сеть. Служба DNS представляет собой распределенную иерархическую ба- зу данных, при этом иерархия серверов DNS отражает иерархию доменных имен. Серверы нижнего уровня иерархии хранят адреса имен соответствующего домена, например сервер домена cisco.com хранит чдреса имен вида xyz.cisco.com. Серверы следующего уровня иерархи хранят уже не адреса конечных узлов, а адреса серверов DNS доменов нижнего уровня; например, DNS-сервер домена сот хранит адрес DNS-сервера домена cisco.com, а также всех DNS-серверов других доменов, входящих в домен сот, например google.com, ibm.com, championat.com и т. п.

Очевидно, что служба DNS является весьма уязвимым элементом транспортной инфраструктуры. Если она не работает, то клиенты оказываются отрезанными от сайтов Интернета, так как резольверы их компьютеров не могут сделать первый шаг в отправке запроса на сайты, поскольку символьное имя сайта не отображается на его IP- адрес. Ну, а

другой вариант атаки — это подмена действительного IP- адреса на ложный, тогда сеть вроде бы работает и сайт отвечает, но >то не тот сайт, с которым пользователь хотел бы взаимодействовать.

Атаки на траспортную инфраструктуру

За время существования Интернета было осуществлено очень много атак на его транспортную инфраструктуру, и в последние годы их количество и качество постоянно растут: так, по данным провайдера Prolexic, специализирующегося на защите серверов клиентов от DDOS-атак, количество таких атак во втором квартале 2013 года выросло на 33 % по сравнению со вторым кварталом 2012 года, их средняя продолжительность возросла за этот период на 123 % (38 часов против 17), а средняя интенсивность трафика атаки выросла почти в

десять раз — с 4,47 до 49,24 Гбит/с26. Наряду с DDoS-атаками осуществляются и атаки взлома протоколов и служб транспортной системы, ведущих к ее некорректной работе. Рост числа и интенсивности атак отражает растущую значимость сетевых сервисов для бизнеса и политики, а также растущие мощности компьютеров, большинство из которых сегодня оснащены интерфейсами 1 GE, а некоторые — и интерфейсами 10 GE, так что сгенерировать мощный поток данных для них не является проблемой.

В приведенном ниже обзоре основных типов атак на сетевую транспортную инфраструктуру особое внимание уделяется следующим обстоятельствам:

- какой элемент транспортной инфраструктуры атакуется: компьютер (точнее, его транспортные средства), маршрутизатор или сервер DNS;
- какой протокол используется для атаки;
- происходит ли взлом используемого в атаке протокола, т. е. нарушается его нормальная работа, или же протокол только является инструментом для порождения трафика DoS/DDoS-атаки;
- на исчерпание какого типа ресурсов направлена DoS/DDoS-атака: количества соединений с конечным узлом или пропускной способности интерфейса.

TCP-атаки

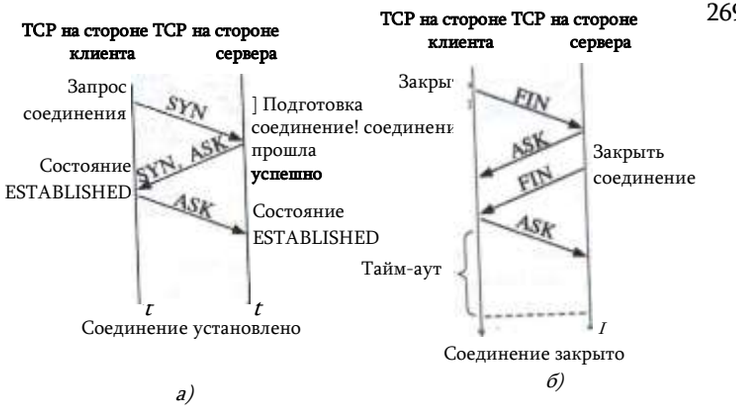
Затопление SYN-пакетами

Этот тип DoS-атаки активно применяется злоумышленниками на протяжении многих лет; впервые он был подробно описан (с приведением кода атаки) в 1996 году, и уже в том же году началось его практическое «применение», которое продолжается и по сей день. Атакуемым является конечный узел, как правило, сервер, работающий с клиентами по протоколу TCP.

Атака *SYN Flood* (flood — затопление) использует уязвимость процедуры установления логического соединения протокола TCP.

Как было сказано ранее, протокол TCP устанавливает логические соединения между прикладными процессами, работающими на двух компьютерах — клиенте и сервере. Например, для передачи данных очередной веб-страницы приложение — веб-браузер — устанавливает новое TCP-соединение с веб-сервером. За веб-серверным процессом закреплен программный порт 80, поэтому серверный процесс при своем старте уведомляет операционную систему, что он будет использовать для сетевого обмена именно этот порт. После этого операционная система переводит порт 80 в состояние «прослушивание сети»

26 www.prolexic.com/knowledge-center-dos-and-ddos-attack-reports.html



1'ис. 9.2. Процедура установления и разрыва логического соединения TCP при нормальном течении процесса

(listen), т. е. в состоянии ожидания прихода TCP-запроса на установление соединения по порту 80.

В ходе установления логического соединения модули TCP двух 1'пмпьютеров договариваются между собой о параметрах процедуры обмена данными, таких как максимальный размер сегмента, который клждая сторона готова принимать, максимальный объем данных (возможно несколько сегментов), которые она разрешает другой стороне передавать в свою сторону, даже если та еще не получила квитанцию и.) предыдущую порцию данных (размер окна), начальный порядковый номер байта, с которого она начинает отсчет потока данных в рамках данного соединения. В дальнейшем эти параметры позволяют узлам контролировать надежность передачи данных и в случае необходимости восстанавливать потерянные сегменты.

Процедура установления соединения основана на использовании флагов *SYN* и *ACK,* переносимых в заголовке каждого TCP-сегмента. Она может быть представлена в виде трех шагов (рис. 9.2,а):

1. Соединение инициируется отправкой с компьютера-клиента сегмента TCP, содержащего флаг SVN, установленный в 1.

2. Получив запрос, модуль TCP на стороне сервера пытается создать «инфраструктуру» для обслуживания нового клиента. Он обращается к операционной системе с просьбой о выделении определенных системных ресурсов для организации буферов, таймеров, (.нетчиков. Эти ресурсы закрепляются за соединением с момента его создания и до момента разрыва; в спецификации TCP (RFC 789) эти ресурсы объединяются в ***Transmission Connection Block, TCB.*** Если на стороне сервера все необходимые ресурсы были получены и все необходимые действия

выполнены, то модуль TCP посылает клиенту

A .. CVM

Рис. 9.3. **Проведение DoS-атаки, в которой используются особенности протокола TCP:** *а* **— нормальный порядок установления TCP-соединения;** *б* **— DDoS-атака за счет создания множества незакрытых TCP-соединений**

3. В ответ клиент посылает сегмент с флагом АСК и переходит в состояние установленного логического соединения (состояние ESTABLISHED). Когда сервер получает флаг АСК, он также переходит в состояние ESTABLISHED. На этом процедура установления соединения заканчивается и стороны могут переходить к обмену данными.

Отметим, что с одним и тем же программным портом может быть установлено несколько одновременно существующих соединений, что весьма характерно для серверов, например веб-серверов, одновременно обслуживающих большое количество клиентов.

Соединение может быть разорвано в любой момент по инициативе любой стороны. Для этого клиент и сервер должны обменяться сегментами FIN и АСК в последовательности, показанной на рис. 9.2***,6*** (здесь инициатором является клиент). Соединение считается закрытым по прошествии некоторого времени, в течение которого сторона-инициатор убеждается, что ее завершающий сигнал АСК дошел нормально и не вызвал никаких «аварийных» сообщений со стороны сервера.

А теперь рассмотрим, как строится DoS-атака с использованием процедуры установления TCP-соединения.

Для выполнения атаки злоумышленник организует передачу на сервер массированного потока пакетов с флагом ***SYN,*** каждый из которых инициирует создание нового TCP-соединения (рис. 9.3,6). Получив сегмент с флагом ***SYN,*** сервер выделяет для нового соединения необходимые ресурсы — блоки TCB — и в полном соответствии с протоколом отвечает клиенту сегментом с флагами ***ASK*** и ***SYN.*** После этого, установив таймаут, он начинает ждать от клиента завершающий сегмент с флагом ***ASK,*** который, увы, так и не приходит. Аналогичным образом создается множество других «недоустановлен- ных» соединений. Обычно операционная система сервера имеет лимит на количество одновременно поддерживаемых «недоустановлен- ных» TCP соединений (глобально или для каждого программного пор-

1.) отдельно), так как каждое открытое соединение требует выделения памяти ядра ОС для нового блока TCB (размеры которого могут сос- ивлять от 280 до 1300 байтов в зависимости от ОС). При достижении этого лимита ОС начинает отвергать все последующие запросы на установление TCP-соединений, а следовательно, отказывает в обслуживании и легальным клиентам сервера. По истечению тайм-аута ОС удаляет из памяти TCB-блоки «недоустановленных» соединений и снова начинает принимать новые соединения, поэтому атакующий для поддержания атаки должен продолжать посылать SYN-сегменты.

Для осуществления атаки SYN Flood IP-пакеты атакующего узла могут использовать как реальный (прямая SYN-атака), так и «поддельный» адрес или адреса отправителя (т. е. когда атакующий использует спуфинг). В обоих случаях атакующий должен заблокировать нормальную реакцию своего компьютера на получение от атакуемого сервера сегмента с флагами SYN/АСК, которая состоит в отправке серверу ответного сегмента с флагом АСК. Если этого не сделать и позволить протоколу TCP работать стандартным образом, то атакуемый сервер посчитает процедуру установления соединения TCP завершенной, удалит соответствующий блок TCP из списка «недоустановленных» соединений и начнет принимать новые соединения. В случае прямой атаки атакующий обычно фильтрует входящий трафик, отсеивая ответы SYN/АСК атакуемого сервера. В случае использования «поддельного» адреса отправителя атакующий должен выбрать такие адреса, которые исключают реакцию на сегменты SYN/ACK, например адреса несуществующих узлов.

Обычно атака SYN Flood обнаруживает себя несоответствием в трафике количества SYN-сегментов количеству АСК-сегментов, идущих от того же источника. При этом заметного всплеска трафика может быть и не быть, поскольку лимит «недоустановленных» соединений сам по себе не столь велик. Метод борьбы с атакой SYN Flood основан на фильтрации трафика от источника SYN Flood-пакетов, для чего нужно определить адрес атакующего узла. Очевидно, что при использовании спуфинга это сделать сложнее, так как это может потребовать взаимодействия нескольких Интернет-провайдеров, через сети которых проходит трафик атаки.

Спуфинг IP-адресов источника используется во многих типах атак, поэтому борьба с ним — естественный элемент обеспечения сетевой безопасности. Основным средством противостояния спуфингу является применение на маршрутизаторах техники ***«Проверки обратного пути»*** (***Reverse Path Check, RPC).*** Идея этой проверки достаточно проста — пакет должен передаваться маршрутизатором в соответствии с его адресом назначения только в том случае, если его адрес источника имеется в таблице маршрутизации для интерфейса,

с которого этот пакет получен. Действительно, если компьютер злоумышленника подключен к сети 212.100.100.0/24, но генерирует пакеты с адресом источника 25.0.30.18, то маршрутизатор провайдера, к которому подключена сеть 212.100.100.0/24 легко может проверить, что через интерфейс, на который был получен пакет с подделанным адресом, достичь сеть 25.0.30.18 нельзя, а значит пакет нужно отбросить. Однако RPC может приводить к отбрасыванию пакетов легального пользователя, если его сеть имеет несколько подключений к сетям разных провайдеров.

Распределенная атака ***DDoS SYN Flood*** организует затопление атакуемого компьютера силами десятков и сотен зараженных компьютеров, поэтому использование выше описанных методов защиты — фильтрации трафика и RPC — в данном случае становится проблематичным.

В таких случаях могут быть привлечены другие, более сложные в реализации способы защиты, которые требуют изменения параметров протокола TCP —увеличения предельного числа «недоустанов- ленных» соединений, уменьшения тайм-аута вытеснения старых «недоустановленных» соединений, а также усложнения логики трехшаговой процедуры установления соединения. В последнем случае при приеме запроса SYN сервер создает, но ***не сохраняет в своей памяти блок TCB,*** а посылает его в компрессированном виде клиенту с ответом SYN/ACK. При нормальном ходе установления соединения клиент отвечает сегментом АСК, в котором повторяет полученный компрессированный TCB, поэтому, получив его, сервер знает данные TCB для этого соединения и создает соответствующий TCB в памяти своего ядра. Так как начальная стадия соединения не требует запоминания данных об этом соединении на сервере, а значит не тратятся его ресурсы, то и исчезают условия для проведения атаки SYN Flood.

Разновидностью атаки SYN Flood является атака ***ACK Flood*** В этой атаке злоумышленник посылает SYN-пакеты с адресом жертвы в поле адреса источника на большое количество серверов. Эти серверы отвечают на SYN-пакетами с установленным битом АСК, которые бомбардируют компьютер-жертву и исчерпывают пропускную способность его входного интерфейса.

Подделка TCP-сегментпа

Для защиты сегментов некоторого TCP-соединения от смешения с сегментами других соединений для каждого соединения случайным образом выбирается номер первого байта передаваемого потока данных. Затем каждый сегмент данных идентифицируется сдвигом относительно начала потока. В ходе переговорного процесса модули TCP обоих участвующих в обмене сторон договариваются между собой о параметрах процедуры обмена данными. Одним из таких параметров чиляется ***начальный номер байта,*** с которого будет вестись отсчет и течение времени существования данного соединения. При приеме очередного сегмента протокол TCP проверяет, находится ли его порядковый номер в разрешенном для данного соединения диапазоне, и только в случае положительного результата такой проверки

добавляет принятые данные к ранее принятым в ходе данного соединения **ГСР** байтам.

Однако этот механизм защиты не так уж надежен, чем и поль- |уются злоумышленники. Атака «Подделка TCP-сегмента» состоит м генерации сегментов TCP, все атрибуты которых имеют значения, легитимные для некоторого существующего TCP-соединения атакуемого компьютера, т. е. IP-адреса, номера TCP-портов источника и назначения, а также порядковые номера из текущего диапазона. Принимающая сторона не может отличить такие «поддельные» сегменты от настоящих и помещает информацию злоумышленника в поток пользовательских данных, а значит, злоумышленник может добиться желаемого эффекта, например поместить ложную информацию в базу данных, заразить атакуемый компьютер вирусом и т. п.

Для того чтобы «поддельный» сегмент выглядел как настоящий, атакующий может либо прослушивать трафик, либо просто перебирать все возможные значения адресов, портов и порядковых номеров сегментов. Прослушивание трафика представляет собой самостоятельную нетривиальную задачу, связанную с перенаправлением трафика, атаки такого типа мы рассмотрим ниже. Перебор параметров TCP-соединения требует большой вычислительной мощности компьютера атакующего, но это в последнее время не является проблемой. В обоих случаях более уязвимыми являются длительные TCP-соединения, например соединения, установленные для загрузки больших видеофайлов;

Повторение TCP-сегментов

Если злоумышленник смог каким-то образом перехватить трафик между двумя участниками TCP-соединения, то впоследствии он может просто повторно использовать перехваченные сегменты, пересылая их участникам соединения. Злоумышленник может использовать эту технику для разных целей, например он может вызвать таким образом нарушение работы некоторого приложения, пользующегося TCP как транспортом, за счет представления устаревшей информации (перехваченной) как новой.

Сброс TCP-соединения

Эта атака является разновидностью предыдущей. Она использует флаг RST (ReSeT) в заголовке сегмента TCP. Этот флаг предназ

начен для аварийного прекращения TCP-соединения, при его приеме узел должен немедленно завершить сессию, к которой сегмент, несущий флаг RST, относится, и удалить все данные, полученные в ходе этого соединения. Разработчики протокола TCP ввели этот флаг для отработки аварийных ситуаций, например если в одном из узлов произошел сбой во время TCP-соединения, то после восстановления сбоя он может воспользоваться этим признаком и уведомить узел- собеседник, что сессия не может быть продолжена и все принятые ранее данные недействительны.

Для того чтобы атака удалась, злоумышленник должен «подделать» заголовок сегмента TCP, как и в предыдущем случае.

Интересно, что техника «Сброс соединения» используется не только злоумышленниками, но и разработчиками средств защиты, например некоторые файерволы используют ее для прекращения атаки. Известен также случай, когда ее применил провайдер (Comcast) для того, чтобы бороться с программами обмена файлами, работающими на пользовательских компьютерах и нарушающими авторские права владельцев звуко- и видеозаписей. Сторонники принципа нейтральности Интернета осудили эту практику, а через некоторое время ее признала незаконной и Федеральная комиссия по связи США.

Борьба с атаками «Подделка TCP-сегмента» и «Сброс TCP- соединения» может вестись по двум направлениям: во-первых, недопущением вспомогательной атаки прослушивания трафика, во-вторых, внесением изменений в протокол TCP, например, включением в него аутентификации каждого сегмента с использованием цифровой подписи.

Существуют два стандарта, описывающие механизм цифровой подписи сегментов TCP: TCP MD5 (RFC 2386) и TCP AO (RFC 5925). Стандарт TCP AO заменяет стандарт TCP MD5, являясь более мощным и гибким, но на практике они применяются оба (но не одновременно).

Стандарт *TCP MD5* описывает процедуру цифровой подписи заголовков IP, TCP и поля данных TCP с помощью алгоритма хеширования MD5, который использует разделяемый секрет в каждом из маршрутизаторов. Как мы знаем, цифровая подпись не обеспечивает конфиденциальности, так как содержимое защищаемых полей не шифруется, но гарантирует тот факт, что сегмент TCP не был изменен третьей стороной.

Более поздний стандарт *TCP AO* позволяет использовать несколько функций хеширования, при этом обязательной является либо хеш-функция SHA-1, либо AES-128, так как они производят криптографически более стойкие цифровые подписи, чем MD5. Кроме использования более стойких алгоритмов подписи, стандарт TCP AO

V иливает также защиту от атак повторения, которые особенно опасны для длительных TCP-сессий. Для этого вместо единственного и,

| /н'довательно, неизменного в течение сессии разделяемого секрета, и< пользуется ***мастер-ключ*** с некоторыми параметрами, на основе ко- трого можно вырабатывать и периодически менять ключи сессии.

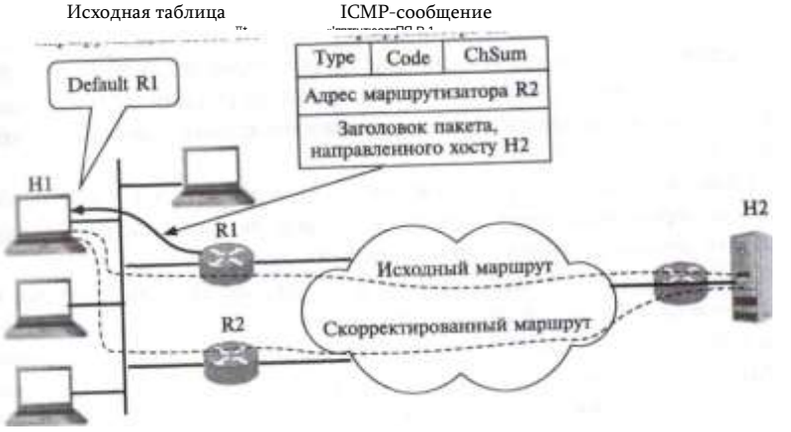
ICMP-атаки

Перенаправление трафика

Перенаправление трафика можно осуществить разными способами и в разных целях, одну из них мы только что рассмотрели в атаках,

| низанных с подделкой сегментов TCP. В пределах локальной сети это можно также сделать с помощью протокола ICMP. В соответствии с данным протоколом при отказе некоторого маршрута или в случаях, когда обнаруживается, что для некоторого адреса назначения хост использует нерациональный маршрут, маршрутизатор посылает хосту *ICMP-сообщение о перенаправлении маршрута.*

На рис. 9.4 применяемый по умолчанию маршрутизатор R1, помучив от хоста H1 пакет, адресованный хосту H2, определяет, что наилучший маршрут к хосту H2 пролегает через другой маршрутизатор данной локальной сети, а именно через маршрутизатор R2. Маршру- 1изатор R1 отбрасывает полученный пакет и помещает его заголовок и ICMP-сообщение о перенаправлении маршрута, которое посылает хосту H1. В сообщении содержится IP-адрес альтернативного маршрутизатора R2, который хост теперь должен использовать, посылая данные хосту H2. Хост H1 вносит изменения в свою таблицу маршрутизации и с этого момента



отправляет пакеты хосту H2 по новому скорректированному маршруту.

маршрутизации хоста H1 **маршрутизатора R1**

Рис. 9.4. Перенаправление маршрута маршрутизатором по умолчанию

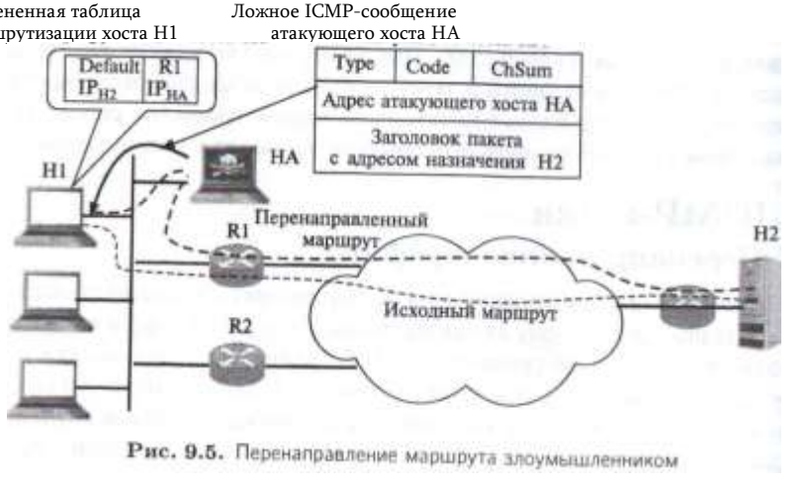


Рис. 9.5. Перенаправление маршрута злоумышленником

Для перехвата трафика, направляемого хостом H1 хосту H2, злоумышленник должен сформировать и послать хосту H1 пакет, маскирующийся под ICMP-сообщение о перенаправлении маршрута (рис. 9.5). В этом сообщении содержится запрос о корректировке таблицы маршрутизации хоста H1, так чтобы во всех пакетах с адресом $1P_H2$ адресом следующего маршрутизатора стал адрес $1P_{HA}$. являющийся адресом хоста-злоумышленника HA.

Для того чтобы хост «поверил» этому сообщению, в поле IP- адреса отправителя должен быть помещен адрес маршрутизатора R1, являющегося маршрутизатором по умолчанию. Когда пакеты, передаваемые введенным в заблуждение хостом, начнут поступать на узел злоумышленника, он может либо захватывать и не передавать эти пакеты дальше, имитируя для поддержания диалога приложение, которому эти пакеты предназначались, либо организовать транзитную передачу данных по указанному адресу назначения $1P_H2$- Читая весь трафик между узлами H1 и H2, злоумышленник получает все необходимую информацию для несанкционированного доступа к серверу H2.

Сами маршрутизаторы также могут реагировать на ICMP-сообщения о перенаправлении маршрута, но обычно провайдеры отключают эту опцию для предотвращения атак данного типа.

Заметим, что простейший вариант перенаправления трафика в локальной сети может быть осуществлен путем отправки в сеть ***ложного ARP-ответа.*** В данном случае схема очевидна: получив широковещательный ARP-запрос относительно некоторого IP-адреса, злоумышленник посылает ложный ARP-ответ, в котором сообщается, что данному IP-адресу соответствует его собственный MAC-адрес.

ICMP Smurf-атака

ICMP Smurf — это DDoS-атака, использующая функцию эхо-запроса

протокола ICMP. Название атаки произошло от имена файла imurf.c, содержащего код атаки и получившего распространение в 1998 году.

Эхо-запросы и ответы ICMP больше известны по утилите ***ping*,***

■ помощью которой можно проверить достижимость узла Интернета Утилита ping посылает тестируемому узлу пакет ICMP, в котором указан тип сообщения 8 — «эхо-запрос». Получив его, тестируемый узел отправляет в обратном направлении пакет ICMP с кодом 0 — «эхо-ответ».

Атака Smurf использует тот факт, что эхо-запрос может быть послан не только по индивидуальному, но и ***широковещательному*** (broadcast) адресу некоторой сети. Например, если адрес сети 200.200.100. 0/24, то ее широковещательным адресом будет 200.200.100.255, и пакет с таким адресом назначения должен быть доставлен всем узлам этой сети.

Атаку иллюстрирует рис. 9.6. Компьютер злоумышленника с адресом 167.50.31.17 находится в сети 167.50.31.0/24, а атакуемый компьютер имеет адрес 195.204.20.145 и подключен к сети 195.204.20. 0/24. Компьютер злоумышленника генерирует эхо-запросы ICMP с адресом назначения 200.200.100.255 и адресом источника 195.204.20. 145. Эхо-запросы передаются через Интернет в сеть 200.200.100.0.24 и принимаются всеми узлами этой сети, которые отвечают на запросы ICMP эхо-ответами. В том случае, когда в сети 200.200.100.255 имеется достаточно большое количество активных узлов (понятно, что их не может быть более 254), на атакуемый узел 195.204.20.145 посы-

Рис. 9.6. Компоненты ICMP Smurf-атаки

См. Приложение 2 раздел «Протокол ICMP. Утилита ping».

лается интенсивный поток эхо-ответов, так как именно его адрес был указан в эхо-запросах как адрес источника, которому надо ответить.

В результате сетевой интерфейс атакуемого компьютера оказывается затоплен эхо-ответами и при превышении интенсивности этого потока некоторой величины его пропускная способность оказывается исчерпанной.

Атака ICMP Smurf использует характерный прием — усиление атаки за счет ***отражения*** посланного пакета большим количеством компьютеров. Этот прием превращает DoS атаку в DDoS без использования сети ботов, так как все компьютеры, отвечающие на эхо-запросы, работают в обычном режиме протокола ICMP, они не должны быть предварительно заражены каким-то вирусом. Как мы увидим далее, усиление отражением используется и в других типах атак.

Атака ICMP Smurf представляет сегодня скорее исторический интерес, так как она основана на передаче через Интернет IP-пакета с широковещательным (в пределах некоторой сети) адресом. До 1999 года такая передача была обязательной для маршрутизаторов Интернета, но из-за атак, подобных ICMP Smurf, в стандарты было внесено изменение и сегодня режимом по умолчанию является фильтрация пакетов с широковещательными адресами. Кроме того, промежуточная сеть, узлы которой используются для отражения эхо-запроса, может быть экранирована с помощью файервола от эхо-запросов, поступающих со стороны внешних сетей.

В том случае, когда атака начинается из сети, к которой принадлежит атакуемый компьютер, экранирование запросов не может быть выполнено. В этом случае атаку можно предотвратить, запретив компьютерам сети реагировать на широковещательные эхо-запросы.

Ping-смерти и ping-затопление

Эта атака с несколько драматическим названием ***Ping of Death*** состоит в отправке на атакуемый компьютер эхо-запроса с длиной IP-пакета, превышающей его максимально возможный размер 65 535 байтов (согласно RFC 791). Если соответствующий буфер ядра ОС не рассчитан на такой размер, то ОС терпит крах, отсюда и такое название атаки. Так как эта атака основана на превышении размера буфера при сборке фрагментированного пакета IP, то она является частным случаем атак, использующих фрагментацию IP, которые рассмотрены ниже.

Атака Ping of Death уже давно имеет только исторический интерес, так как разработчики ОС в середине 90-х годов ввели в стек IP необходимую проверку длины собираемого фрагментированного IP- пакета и тем самым ликвидировали саму основу атаки.

Атака ***Ping Flood*** достаточно примитивна — злоумышленник просто использует утилиту ping своей операционной системы для для затопления атакуемого компьютера эхо-запросами. Если скорость сете- пого интерфейса его компьютера больше, что у атакуемого компьютера, то атака удается, так как вся входная пропускная ^способность интерфейса компьютера-жертвы оказывается исчерпаной. К тому же атакуемый компьютер будут успевать отвечать на часть эхо-запросов тхо-ответами, что дополнительно приведет к частичному исчерпанию пропускной способности в выходном направлении, а также к замедлению работы программ из-за отвлечения центрального процессора на обработку эхо-запросов.

UDP-атаки

UDP-затопление

Протокол UDP работает без установления соединений, это свойство используется при организации DoS-атаки ***UDP Flood,*** направленной на исчерпание пропускной способности интерфейса атакуемого компьютера, так как атакуемый компьютер обязан принимать все направляемые ему UDP-дейтаграммы и не может заставить передающий компьютер ограничить скорость направляемых ему пакетов, как это можно сделать в протоколе TCP, уменьшив размер окна приема. В атом отношении UDP DoS-атака похожа на атаку Ping Flood, так как заключается просто в направлении интенсивного потока UDP- дсйтаграмм на атакуемый компьютер. Злоумышленник может использовать аппаратный генератор трафика для того, чтобы генерировать UDP-трафик с максимально возможной скоростью выходного интерфейса, игнорируя ответные ICMP-сообщения в тех случаях, когда программный порт, указанный в пакетах UDP, у атакуемого компьютера не открыт.

Однажды один из авторов этой книги нечаянно «организовал» подобную атаку, участвуя в тестах по исследованию качества обслуживания разных классов трафика. В одном из тестов UDP-трафик интенсивности около 90 Мбит/с вместо того, чтобы быть адресованным хосту тестируемой подсети, был по ошибке направлен на хост библиотеки Университета в Саусгэмптоне. Так как локальная сеть библиотеки была подключена к интерфейсу маршрутизатора со скоростью 100 Мбит/с (это было довольно давно, в 2004 году, сегодня такое подключение библиотеки не сможет удовлетворить пользователей и без DoS-атаки), то длившийся 15 минут UDP-поток вызвал настолько заметное ммедление работы браузеров посетителей библиотеки, что некоторые особенно требо- гытельные из них тут же позвонили сетевому администратору, который, в свою очередь, ^фиксировал этот инцидент как реальную атаку.

Слабостью этого вида атак является принципиальное ограничение ьштрнгпвности атаки интерфейсом атакующего компьютера. Если

-----ч-и- 1 гк„т /,-

то невозможно затопить UDP-пакетами сервер с интерфейсом 10 Гбит/с. Однако злоумышленник может преодолеть это ограничение, заполучив в свое распоряжение сеть ботов. Как мы увидим ниже, именно таким образом в 2007 году была осуществлена массированная DDoS-атака UDP на корневые серверы DNS, при этом трафик создавался примерно 5000 ботами.

Кроме прямой UDP-атаки, существуют также атаки отражения, когда UDP-трафик используется для инициирования большого количества ответных

пакетов, которые и атакуют компьютер жертвы.

ICMP/ UDP-затопление

DoS-атака ***ICMP/UDP Flood*** имеет двойное имя, так как использует два протокола. Злоумышленник направляет UDP-пакеты, в которых в поле адреса источника указан адрес компьютера-жертвы, на программные порты компьютеров вспомогательной сети (для этого может быть использован широковещательный адрес). В пакетах UDP указываются номера портов, находящихся в *пассивном состоянии,* т. е. с указанными портами не связаны приложения, слушающие сеть. При получении UDP-пакета с номером пассивного порта компьютеры вспомогательной сети в соответствии с логикой работы стека TCP/IP отвечают «источнику» UDP-пакетов (атакуемому компьютеру) диагностическими сообщением протокола ICMP «Порт назначения недостижим». Как видим, атака построена на отражении трафика от компьютеров вспомогательной сети; в случае использования широковещательного адреса она становится DDoS-атакой. В числе мер предотвращения входят те же меры, что и для ICMP Smurf-атаки, плюс пропуск файерволом только тех UDP-пакетов, порты которых соответствуют активным приложениям компьютеров сети. Кроме того, можно ввести ограничение на интенсивность сообщений «Порт назначения недостижим» компьютеров сети.

UDP/echo/chargen-затопление

В атаке ***Echo/Chargen Flood*** подобно предыдущей также используется отражение UDP-пакетов, но в заголовках пакетов указываются *активные* порты UDP-сервисов: echo (порт 7) и chargen (порт 19). Сервис *chargen* при обращении к нему генерирует в ответ строку случайных символов случайной длины от 0 до 512 и посылает ее обратившемуся хосту. Это сервис был встроен в ОС Unix для отладки ее сетевых функций. Аналогичное назначение имеет сервис *echo* (не путать с эхо-запросами и эхо-ответами протокола ICMP), он просто возвращает строку любого запроса по адресу обратившегося хоста.

Идея атаки Echo/Chargen Flood очень проста. В простейшем случае атакующий посылает UDP-пакеты на порт 7 и/или 19 некоторого промежуточного хоста и указывает в качестве обратного адреса адрес атакуемого хоста. Промежуточный хост начинает бомбардировать атакуемый хост ответами сервисов chargen и/или echo. Правда, усиления атаки не происходит, так как размер ответов невелик; для усиления можно использовать широковещательный адрес промежуточной сети. Любопытной выглядит атака, когда атакующий посылает промежуточному хосту пакет, в котором в качестве порта назначения указывает номер 19, а в качестве порта-отправителя — 7. В этом случае единственный пакет атакующего вызывает бесконечный обмен пакетами между сервисом chargen промежуточного хоста и сервисом echo атакуемого хоста.

IP-атаки

Протокол IP сам по себе не предоставляет злоумышленникам много

шансов для атак, так как он работает без установления соединений и достаточно прост в обработке как маршрутизаторами, так и конечными узлами. Тем не менее, некоторые возможности для атак существуют. При описании атак мы рассматриваем отдельно версию IPv4 и версию IPv6 протокола IP. Версия IPv6, несмотря на ее более чем 10-летний возраст, пока еще не заменила IPv4, но уже работает в сетях подавляющего числа провайдеров Интернета параллельно с IPv4. Используемая при этом концепция двойного стека (dual stack) позволяет пользователям обращаться к одним и тем же сайтам как по протоколам стека IPv4, так и IPv6.

Атака IP-опции

Эта атака представляет собой DoS-атаку на маршрутизаторы и использует поле дополнительных опций протокола IP.

В соответствии со стандартом RFC 791 заголовок IP-пакета версии 4 может включать *поле опций,* которые задают некоторую нестандартную обработку пакета маршрутизатором. Например, существует опция «Строгая маршрутизация от источника», которая позволяет отправителю IP- пакета задать точный список адресов промежуточных маршрутизаторов, через которые должен проходить маршрут доставки пакета, в то время как опция «Свободная маршрутизация от источника» задает только некоторые из промежуточных маршрутизаторов маршрута. Опция «Фиксация маршрута» требует от маршрутизаторов фиксации в пакете адресов промежуточных маршрутизаторов, которые передавали пакет. Существует также возможность для производителей маршрутизаторов определять свои типы опций.

Атака основана на том факте, что у большинства IP-пакетов поле опций отсутствует, поэтому для продвижения таких пакетов использует-специализированные процессоры портов, которые очень быстро

и экономно выполняют свою операцию. А вот если встречается пакет с полем опций, то специализированный процессор его обработать не может и передает пакет центральному процессору маршрутизатора, и обработка пакета существенно замедляется. В результате поток пакетов, у которых присутствует одна или несколько опций, может привести к серьезному замедлению работы маршрутизатора, в предельном случае — к отказам в обслуживании нормальных пакетов. Усугубляет ситуацию присутствие в пакете двух взаимоисключающих опций, например «Строгая маршрутизация от источника» и «Свободная маршрутизация от источника» с разными промежуточными адресами.

Обычная практика борьбы с этой атакой — полная фильтрация (отбрасывание) пакетов, в заголовке которых имеются опции. Возможно также полное игнорирование поля опций. Существует также промежуточная тактика, когда маршрутизатор реагирует только на некоторые типы опций, а остальные игнорирует.

Спецификация IPv6 (RFC 2460) допускает наличие ***нескольких заголовков*** в пакете — основного и дополнительных. Формат основного заголовка IPv6 проще единственного заголовка IPv4, в нем нет полей фрагментации пакетов, а также полей опций. Это сделано для ускорения обработки пакетов IPv6 маршрутизаторами. Однако вместо полей опций в пакете IPv6 могут присутствовать дополнительные заголовки, и один из них — заголовок пошаговых опций (Hop-by-hop Options), который должен обрабатываться маршрутизаторами при передаче пакета. Как и в случае опций IPv4, опции дополнительного заголовка пошаговых опций IPv6 обычно обрабатываются центральным процессором маршрутизатора. В настоящее время определено только небольшое число значений типа пошаговых опций. Поэтому помещение в такой заголовок большого числа опций со случайными значениями их типа будет замедлять работу маршрутизатора IPv6 аналогично случаю IPv4.

Атака IP-фрагментация

Эта атака направлена на конечные узлы IP-сетей, в обязанность которых входит сборка фрагментированного IP-пакета в единое целое. Как показала практика, операция сборки имеет несколько уязвимостей, одна из них уже была упомянута при описании атаки Ping of Death, когда ОС терпит крах из-за превышения длины собранного пакета размера буфера.

Важной особенностью протокола IP (версии IPv4) является его способность выполнять ***динамическую фрагментацию*** пакетов при передаче их между сетями с различными максимально допустимыми значениями длины поля данных кадров ***(Maximum Transmission Unit, MTU).*** Значения MTU зависят как от протокола, так и от настройки сетевых интерфейсов.

Прежде всего, отметим разницу между фрагментацией сообщений ***в узле-отправителе*** и динамической фрагментацией сообщений в ***транзитных узлах*** сети — маршрутизаторах.

В первом случае деление сообщения на несколько более мелких частей (фрагментов) происходит при передаче данных между протоколами одного и того же стека внутри компьютера. Протоколы, выполняющие фрагментацию в ***пределах*** конечного узла, анализируют тип технологии нижнего уровня, определяют ее MTU и делят сообщения на такие части, которые умещаются в кадры канального уровня того же стека протоколов.

В стеке TCP/IP эту задачу решает протокол TCP, который разбивает поток байтов, передаваемый ему с прикладного уровня, на сегменты нужного размера, например по 1460 байт, если на нижнем уровне данной сети работает протокол Ethernet. Протокол IP в узле- отправителе, как правило, не использует свои возможности по фрагментации пакетов.

А вот на транзитном узле — маршрутизаторе, когда пакет необходимо передать из сети с большим значением MTU в сеть с меньшим шачением MTU, способности протокола IP выполнять фрагментацию становятся востребованными. ***Пакеты-фрагменты,*** путешествуя по сети, могут вторично подвергнуться фрагментации на каком-либо из промежуточных маршрутизаторов.

Каждый из фрагментов должен быть снабжен полноценным заголовком IP. Некоторые из полей заголовка (идентификатор, TTL, флаги DF и MF, смещение) непосредственно предназначены для последующей сборки фрагментов в исходное сообщение.

Идентификатор пакета используется для распознавания пакетов, образовавшихся делением на части (фрагментации) исходного паке- тл. Все части (фрагменты) одного пакета должны иметь одинаковое шачение этого поля. Модуль IP, отправляющий пакет, устанавливает н поле идентификатора значение, которое должно быть уникальным для данной пары отправителя и получателя в течение всего времени, пока данный пакет (или любой его фрагмент) может существовать в составной 1P-сети.

Поле смещения фрагмента (13 битов) предоставляет получателю информацию о положении фрагмента относительно начала поля данных исходного нефрагментированного пакета. Так, например, первый Фрагмент будет иметь в поле смещения нулевое значение. В пакете, нс разбитом на фрагменты, поле смещения также имеет нулевое значение. Смещение задается в 8-байтных словах, т. е., например, смещение 10 задает положение фрагмента в 80 байтов от начала пакета.

Однобитный *флаг* ***MF*** (More Fragments — больше фрагментов), установленный в единицу, говорит о том, что данный пакет является промежуточным (не последним) фрагментом. Модуль IP, отправляющий нефрагментированный пакет, устанавливает бит MF в нуль.

Флаг ***DF*** (Do not Fragment — не фрагментировать), установленный в единицу, запрещает маршрутизатору фрагментировать данный пакет. Если помеченный таким образом пакет не может достигнуть получателя без фрагментации, то модуль IP его уничтожает, а узлу- отправителю посылается диагностическое сообщение.

Отметим, что IP-маршрутизаторы не собирают фрагменты пакетов в более крупные пакеты, даже если на пути встречается сеть, допускающая такое укрупнение. Это связано с тем, что отдельные фрагменты сообщения могут перемещаться по составной сети разными маршрутами, поэтому нет гарантии, что все фрагменты на своем пути пройдут через какой-то один определенный маршрутизатор.

А теперь посмотрим, как происходит *сборка фрагментированного пакета на хосте назначения.*

На хосте назначения для каждого фрагментированного пакета отводится отдельный буфер. В этот буфер принимающий протокол IP помещает IP-фрагменты, у которых совпадают IP-адреса отправителя и получателя, а также значения в полях идентификатора. Все эти признаки говорят модулю IP, что данные пакеты являются фрагментами одного исходного пакета. Сборка заключается в помещении данных из каждого фрагмента в позицию, определенную *смещением,* указанным в заголовке фрагмента.

Когда первый фрагмент исходного пакета приходит на хост-получатель, этот хост запускает *таймер,* который определяет максимальное время ожидания прибытия остальных фрагментов данного пакета. В различных реализациях IP применяются разные правила выбора максимального времени ожидания. В частности, таймер может быть установлен на фиксированный период времени (от 60 до 120 секунд), рекомендуемый RFC. Как правило, этот интервал достаточен для доставки пакета от отправителя получателю. В других реализациях максимальное время ожидания определяется с помощью адаптивных алгоритмов измерения и статистической обработки временных параметров сети, позволяющих оценивать ожидаемое время прибытия фрагментов. Наконец тайм-аут может быть выбран на базе значений TTL прибывающих фрагментов. Последний подход основан на том, что нет смысла ожидать, пока прибудут другие фрагменты пакета, если время жизни одного из прибывших фрагментов уже истекло.

Если хотя бы один фрагмент пакета не успеет прийти на хост назначения к моменту истечения таймера, то никаких действий по дублированию отсутствующего фрагмента не предпринимается, а все по-

пученные **к этому времени фрагменты пакета отбрасываются! Хосту,**
послащему **исходный пакет, направляется ICMP-сообщение об ошибке**
Такому **поведению протокола IP вполне соответствует его кредо —**
(тараться **по возможности, но никаких гарантий не давать.**

Признаком окончания сборки является отсутствие незаполненных промежутков в поле данных и прибытие последнего фрагмента (с равным нулю флагом MF) до истечения таймаута. После того как данные собраны, их можно передать вышележащему протоколу, например TCP.

Теперь настало время посмотреть, как злоумышленник может использовать механизм фрагментации для атаки на конечный узел.

Превышение максимальной длины пакета (переполнение буфера сборки). Максимальное значение смещения фрагмента равно (2-13 — 1) • 8 = 8191 • 8 = 65528. Так как максимальная длина IP-пакета равна 65 535 байтов, то, очевидно, что последний фрагмент не должен иметь длину более 7 байтов. Задавая фрагмент с максимальным смещением и размером в 8 и более байтов, злоумышленник переполняет буфер ядра ОС, что может привести к краху ОС. Атака Ping of Death основана на этой уязвимости.

Перекрытие сегментов за счет специального подбора смещений и длин фрагментов. Некоторые ОС не справляются со сборкой таких пакетов и терпят крах. Атака Teardrop использует эту уязвимость.

Замещение фрагментов. Используется для обмана таких защитных средств, как файерволы и системы обнаружения вторжений (IDS). Пакеты атаки фрагментируются и посылаются вместе с фрагментами-дубликатами, в которых содержится безобидная информация. Первым посылается безобидный фрагмент, а потом — фрагмент, содержащий код атаки, но с такими же смещением и длиной. В результате пришедший позже фрагмент атаки замещает безобидный фрагмент. Не все файерволы и системы обнаружения вторжений распознают организованную таким образом атаку.

Незавершенные фрагменты. Эта DoS-атака направлена на исчерпание буферов сборки фрагментов. Атакующий посылает на атакуемый компьютер поток пакетов небольшого размера, при этом каждый пакет разбит на два фрагмента. Первый фрагмент из пары посылается с нулевым смещением, а второй — с максимально возможным, поэтому при сборке они занимают максимальную память, отводимую под буфер. Если количество фрагментированных таким образом пакетов достаточно велико, то за время тайм-аута вся память ядра ОС, отводимая под сборку пакетов, оказывается исчерпанной и наступает отказ в обслуживании фрагментированных пакетов.

Версия IPv6 не предусматривает динамическую фрагментацию пакета маршрутизаторами, поэтому основной заголовок IPv6 не содержит признаков фрагментации. Вместо этого конечным узлам IPv6 рекомендуется использовать технику нахождения минимального значения MTU вдоль пути следования пакетов (техника Path MTU Discovery) и затем не посылать пакеты длиной, превышающей это значение. Тем не менее, фрагментация может использоваться и в протоколе IPv6, но ее выполняют только конечные узлы. Для этого используется дополнительный заголовок фрагментации, который

содержит идентификатор пакета и смещение фрагмента, которое, как и версии IPv4, имеет длину 13 разрядов. Отличием от версии IPv4 является передача начального фрагмента IPv6 пакета вместе с каждым последующим фрагментом.

Атаки фрагментации IPv6 в принципе аналогичны атакам IPv4, но так как опыт предыдущей борьбы был учтен, у злоумышленника имеется меньше вариантов атаки. Например, превышение максимальной длины результирующего сегмента возможно, но современные ОС предотвращают такую возможность при использовании протокола IPv6. Основная цель атак фрагментации IPv6 направлена на обход файерволов и систем обнаружения вторжения, а это возможно в том случае, когда они собирают пакеты с перекрывающимися или повторяющимися сегментами.

DNS-атаки

Организация DNS

Доменная служба имен (Domain Name Service, DNS) позволяет отображать символьные имена узлов сети на их IP-адреса (как IPv4, так и IPv6). В стеке TCP/IP применяется доменная система имен, которая имеет иерархическую древовидную структуру, допускающую наличие в имени произвольного количества частей (рис. 9.7).

Иерархия доменных имен аналогична иерархии имен файлов, принятой во многих популярных файловых системах. Дерево имен начинается с корня, обозначаемого здесь точкой (.). Затем следует старшая символьная часть имени, вторая по старшинству символьная часть имени и т. д. Младшая часть имени соответствует конечному узлу сети. В отличие от имен файлов, при записи которых сначала указывается самая старшая составляющая, затем составляющая более низкого уровня и т. д., запись доменного имени начинается с самой младшей составляющей, а заканчивается самой старшей. Части доменного имени отделяются друг от друга точкой. Например, в имени home.microsoft.com составляющая home является именем одного из компьютеров в домене microsoft.com.

Разделение имени на части позволяет ***разделить административную ответственность*** за назначение уникальных имен между различ-

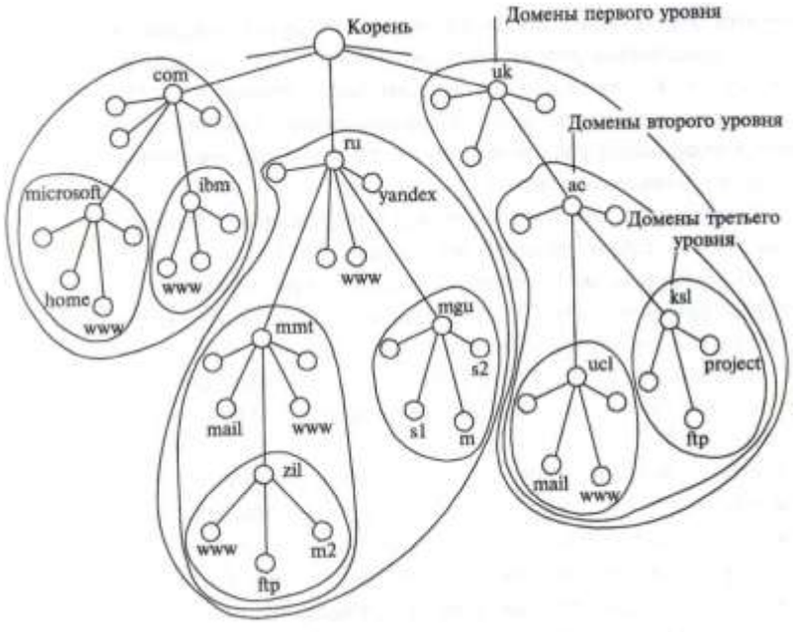


Рис. 9.7. Пространство доменных имен

ными людьми или организациями в пределах своего уровня иерархии. Так, для примера, приведенного на рис. 9.7, некоторая организация может нести ответственность за то, чтобы все имена с окончанием «ги» имели уникальную следующую вниз по иерархии часть. То есть все имена типа www.ru,mail.mmt.ru или m2.zil.mmt.ru отличаются второй по старшинству частью.

Разделение административной ответственности позволяет решить проблему образования уникальных имен без взаимных консультаций между организациями, отвечающими за имена одного уровня иерархии. Очевидно, что должна существовать одна организация, отвечающая за назначение имен верхнего уровня иерархии.

Совокупность имен, у которых несколько старших частей совпадают, образуют ***домен имен (name domain).*** Например, имена www.zil. mmt.ru, ftp.zil.mmt.ru, yandex.ru и sl.mgu.ru входят в домен ги, так как все они имеют одну общую старшую часть — имя ги. Другим примером является домен mgu.ru. Из представленных на рис. 9.7 имен в него входят имена sl.mgu.ru, s2.mgu.ru и rn.mgu.ru. Этот домен образуют имена, у которых две старшие части равны mgu.ru. Администратор домена mgu.ru несет ответственность за уникальность имен следующего уровня, входящих в домен, т. е. имен si, s2 и m. Образованные домены sl.mgu.ru, s2.mgu.ru и rn.mgu.ru являются ***поддо***

менами домена mgu.ru, так как имеют общую старшую часть имени. Часто поддомены для краткости называют только младшей частью имени, т. е. в нашем случае поддоменами являются si, s2 и m.

Если в каждом домене и поддомене обеспечивается уникальность имен следующего уровня иерархии, то и вся система имен будет состоять из уникальных имен.

Корневой домен управляется центральными органами Интернета, в частности такой организацией, как ICANN.

Домены верхнего уровня назначаются для каждой страны, а также для различных типов организаций. Для обозначения стран используются двухбуквенные аббревиатуры, например ru (Россия), uk (Великобритания), fi (Финляндия), us (Соединенные Штаты), а для различных типов организаций, например, следующие трехбуквенные обозначения:

- com — коммерческие организации (например, microsoft.com);
- edu — образовательные организации (например, mit.edu);
- gov — правительственные организации (например, nsf.gov);
- org — некоммерческие организации (например, fidonet.org);
- net — сетевые организации (например, nsf.net).

Каждый домен администрирует отдельная организация, которая обычно разбивает свой домен на поддомены и передает функции администрирования этих поддоменов другим организациям.

Структура службы DNS также иерархическая. Иерархию образуют серверы DNS, которые обслуживают запросы клиентов DNS, а клиентом DNS является практически каждый узел Интернет, будь то клиентский компьютер, сервер приложений или маршрутизатор.

DNS-серверы поддерживают распределенную базу отображений имен на IP-адреса, она основана на текстовых файлах, состоящих из записей отображений и некоторых служебных записей. Запросы к серверам DNS и их ответы выполняются по протоколу DNS, который использует транспорт UDP с портом сервера 53. Клиентами службы DNS является программное обеспечение ОС, называемое ***резольвером,*** приложения ОС сами не обращаются напрямую к службе DNS, а обращаются к резольверу своей ОС. Для своей работы резольвер должен знать IP-адрес по крайней мере одного сервера DNS, этот адрес конфигурируется вручную или получается по протоколу DHCP.

Вершину иерархии серверов DNS составляют 13 пулов корневых серверов, от a.root-servers.net до m.root-servers.net. Корневые серверы хранят текстовые файлы имен и IP-адресов DNS серверов следующего уровня, называемого верхним (top level DNS, tIDNS или TLD). Серверы верхнего уровня хранят данные о именах и адресах имен, входящих в домены верхнего уровня, таких как com, ru или fm, а также о именах серверов DNS, которые обслуживают домены следующего уровня иерархии — второго, такие как

27 Полномочный — authoritative, что можно также перевести как «официальный», «авторитетный» или «аутентичный».

cisco.com или yandex.ru.

Сервер DNS отвечает на запросы клиентов на основе информации, содержащейся в текстовых файлах отображений имен, хранящихся на данном сервере. В принципе сервер DNS мог бы хранить данные всех отображений, входящих в некоторый домен со всеми его поддоменами; при таком подходе сервер верхнего уровня, отвечающий, например, за домен сот, хранил бы в своих файлах записи всех имен, кончающихся на сот: ibm.com, www.ibm.com,www2.ibm.com, cisco.com, www.cisco.com и т. д. Понятно, что такой подход не масштабируем и не может работать в Интернете. Поэтому пространство доменных имен «разрезают» между серверами DNS, обычно так, чтобы сервер DNS хранил записи только в пределах одного уровня, а для имен своих поддоменов хранил только ссылки на серверы DNS, которые отвечают за эти поддомены. Например, DNS-сервер верхнего уровня, отвечающий за домен сот, хранит только записи листьев своего домена, например, имени www.com, а также имен серверов DNS, которые обслуживают поддомена домена сот, например DNS-сервера поддомена cisco.com.

Часть пространства доменных имен, для которых некоторый сервер DNS имеет полную информацию об их отображениях на основе соответствующего текстового файла, называется ***зоной DNS,*** а сам текстовый файл — файлом зоны. Когда сервер DNS дает ответ о записи, входящей в зону, за которую он отвечает, такой ответ называется *полномочным27* ответом, а сам этот сервер по отношению к запросу — полномочным сервером. Как мы увидим далее, сервер DNS может также давать *неполномочный* ответ, если запрос относится не к его зоне, но он знает его за счет кэширования ответов других серверов. В таких случаях сервер по отношению к запросу является неполномочным. Файл зоны состоит из текстовых записей нескольких типов, таких как:

- A — отображает имя в IPv4-afIpec;
- AAAA — отображает имя в 1Руб-адрес;
- NS — определяет имя DNS-сервера для некоторого домена;
- MS — определяет имя почтового сервера для некоторого домена, а также некоторых других.

Протокол DNS позволяет клиенту делать запросы относительно некоторого доменного имена, задавая тип записи или запрашивая все типы, относящиеся к данному имени. Важнейшим параметром программного обеспечения конечного узла сети является адрес сервера

DNS, которому резольвер должен отправлять свои запросы. Такой DNS-сервер называется ***назначенным*** (designated) для данного клиента.

Для обеспечения надежности и высокой производительности для каждой зоны существует один первичный и несколько вторичных серверов DNS. На первичном сервере находится ***мастер-копия*** файла зоны, которая редактируется администратором сервера; вторичные серверы периодически копируют файл зоны с первичного сервера, для этого может использоваться как протокол DNS, в котором имеется соответствующий тип запроса, или администратор может использовать любой протокол копирования файлов, например ftp или scp. В случае, когда файл зоны передается по протоколу DNS, для повышения надежности используется протокол TCP (порт 53).

Существуют две основные схемы разрешения DNS-имен. В первом варианте работу по поиску IP-адреса координирует DNS-клиент:

- DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного имени;
- DNS-сервер отвечает клиенту, указывая адрес следующего DNS-сервера, обслуживающего домен верхнего уровня, заданный в следующей старшей части запрошенного имени;
- DNS-клиент делает запрос следующего DNS-сервера, который отсылает его к DNS-серверу нужного поддомена, и т. д., пока не будет найден DNS-сервер, в котором хранится соответствие запрошенного имени IP-адресу. Этот сервер дает окончательный ответ клиенту.

Такая процедура разрешения имени, когда клиент сам выполняет последовательность запросов к разным серверам имен, называется ***нерекурсивной.*** Эта схема загружает клиента достаточно сложной работой, и она применяется редко.

Во втором варианте реализуется ***рекурсивная*** процедура, в соответствии с которой клиент перепоручает работу своему назначенному DNS-серверу:

- DNS-клиент обращается с запросом к своему назначенному DNS-серверу.

Далее возможны три варианта действий:

- Если назначенный DNS-сервер знает ответ, а это может произойти в случае, когда запрашиваемое имя относится к зоне его ответственности, то он сразу же возвращает ответ клиенту.
- Если назначенный сервер не знает ответ, но в записях его зоны имеется запись типа NS, указывающая на DNS-сервер, который ведет запрашиваемую зону, то он передает запрос непосредственно указанному серверу и получив ответ перенаправляет его клиенту.
- Если назначенный сервер не знает ответ и в записях его зоны нет записи NS с именем DNS-сервера, который ведет запрашиваемую зону, то он выполняет запрос к корневому DNS-серверу и дальше действует точно так же, как это делает клиент в нерекурсивной процедуре. Получив ответ относительно адреса запрашиваемого имени, DNS-сервер передает его клиенту.

В последних двух случаях, получая окончательный ответ от сервера вышестоящего уровня, назначенный сервер кэширует его для того, чтобы при поступлении аналогичного запроса дать быстрый неполномочный ответ. Чтобы служба DNS могла оперативно отрабатывать изменения, происходящие в сети, ответы кэшируются на относительно короткое время — обычно от нескольких часов до нескольких дней, срок жизни записи задается администратором полномочного сервера.

DNS-серверы работают в сетях провайдеров Интернета или в сетях предприятий. DNS-сервер может быть открытым — в этом случае он отвечает любому клиенту, или закрытым — в этом случае он отвечает только клиентам своего предприятия (в случае корпоративной сети) или только своим подписчикам услуг доступа к Интернету (в случае сети провайдера).

Подавляющее большинство серверов DNS использует программное обеспечение ***BIND*** (Berkeley Internet Name Domain), первоначально разработанное в Калифорнийском университете Беркли.

Атаки на DNS

DNS-спуфинг. В атаке ***DNS-спуфинг*** служба DNS является целью и одновременно средством проведения атаки, направленной на умышленное перенаправление трафика от компьютера пользователя к компьютеру злоумышленника.

Поясним DNS-спуфинг на примере (рис. 9.8). Задача злоумышленника состоит в получении доступа к корпоративному серверу. Для этого ему нужно завладеть именем и паролем авторизованного пользователя корпоративной сети. Эту информацию он решает получить путем ответвления потока данных, которые корпоративный клиент посылает корпоративному серверу, в направлении своего компьютера. Злоумышленник знает, что клиент обращается к серверу, указывая его символьное DNS-имя www.example.com. Ему также известно, что перед тем как отослать пакет серверу, программное обеспечение клиентской машины направляет запрос DNS-серверу, чтобы узнать, какой IP-адрес соответствует этому имени.

Цель злоумышленника — навязать корпоративному клиенту свой вариант ответа, в котором вместо IP-адреса корпоративного сервера (в примере 193.25.34.125) злоумышленник указывает IP-адрес атаку-

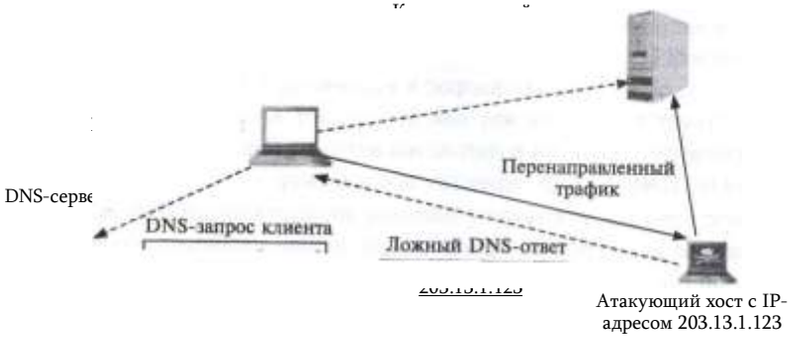


Рис. **9.8. Схема перенаправления трафика путем использования ложных DNS-ответов**

ющего хоста (203.13.1.123). Эта подмена адреса (спуфинг) и дала название данному типу атаки.

На пути реализации этого плана имеется несколько серьезных препятствий. Прежде всего, злоумышленнику необходимо *задержать* ответ DNS-сервера, например, проведя на него DoS-атаку, и, пользуясь этой задержкой, успеть передать свой, ложный ответ на компьютер корпоративного пользователя. Другая проблема связана с *определением номера порта клиента DNS,* который злоумышленник должен поместить в заголовке DNS-ответа, чтобы этот ответ дошел до приложения на машине атакуемого пользователя. Ранее было сказано, что серверная часть службы DNS имеет постоянно закрепленный за ней так называемый «хорошо известный» номер 53. Клиентская же часть DNS получает номер порта от ОС пользователя динамически при запуске, причем операционная система выбирает его из достаточно широкого диапазона. Номер порта, присвоенный клиентской части DNS передается в запросе к серверной части DNS. Однако в описываемом примере злоумышленник не имеет возможности перехватить пакеты корпоративного пользователя. Единственная возможность, которой он располагает — это направлять в адрес пользователя поток пакетов-ложных ответов со *всеми возможными* номерами портов, получаемыми прямым перебором, в надежде, что один из них будет распознан клиентским программным обеспечением как «свой».

Аналогичным образом — перебором всех возможных значений — злоумышленник преодолевает еще одну проблему — *определение идентификаторов DNS-сообщений.* Эти идентификаторы передаются в DNS-сообщениях и служат для того, чтобы клиент службы DNS мог установить соответствие поступающих ответов посланным запросам.

Итак, злоумышленник бомбардирует клиентскую машину ложными DNS-ответами, перебирая все возможные значения идентифицирующих полей так, чтобы клиент, в конце концов, принял один из них за истинный DNS-ответ. Как только это происходит, цель злоумышленника можно считать достигнутой —

вместо корпоративного сервера атакованный компьютер посылает пакеты на адрес атакующего хоста, и злоумышленник получает в свое распоряжение имя и пароль легального пользователя, а с ними и доступ к корпоративному серверу.

Отравление DNS-кэша. ***Атака отравления кэша*** направлена на замену корректной записи отображения имени *в кэше* некоторого DNS- сервера на ту, которая создана злоумышленником, чтобы направить клиента DNS на ложный узел. Эта атака более эффективна, чем DNS-спуфинг, так как, будучи успешной, она действенна в течение длительного времени — времени жизни подложной записи в кэше — и в следствие этого может поразить большое количество клиентов.

Общая схема отравления кэша показана на рис. 9.9. Первым шагом злоумышленника является направление на DNS-сервер ns.victim, org, который злоумышленник хочет отравить, запроса на разрешение несуществующего имени, относящегося к домену злоумышленника hacker.org. Получив такой запрос и не обнаружив искомое имя ни в записях своей зоны, ни в кэше (так как имя не существует), DNS- сервер ns.victim.org начинает рекурсивную процедуру разрешения имени, которая в конце концов приводит его на сервер ns.hacker.org, который ведет зону hacker.org.

Хакер должен заранее подготовить запись, которая должна отравить кэш жертвы. Эту запись он помещает в ответ в качестве дополнительной записи (такие записи разрешаются стандартами DNS для увеличения эффективности работы системы). Содержимое дополнительных записей до некоторого времени не контролировалось программным обеспечением DNS, они просто добавлялись в кэш сервера. Дополнительная запись может относится к различным типам записей DNS. В том случае, когда кэш отравлен записью типа NS, в котором вместо имени легитимного сервера DNS некоторого домена xxx.com будет указано имя DNS-сервера хакера, все запросы клиентов сервера ns.victim.org к хостам домена xxx.com будут направляться

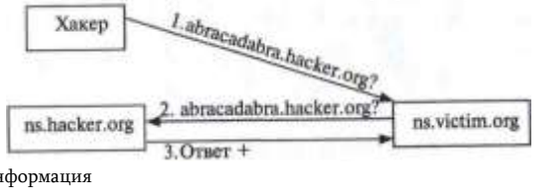


Рис. 9.9. **Общая схема отравления DNS-кэша**

к DNS-серверу хакера, который будет отвечать так, как это задумал злоумышленник.

Атаки отравления DNS-кэша носят исторический характер, так как в современных версиях BIND сервер не кэширует информацию, если она непосредственно не относится к запросу или она выходит за пределы зоны, для которой отвечающий сервер является полномочным. Поэтому современная версия BIND в нашем примере не будет кэшировать дополнительную информацию, которую передает ей сервер хакера.

Атаки на корневые DNS-серверы. Корневые серверы — наиболее уязвимое звено службы DNS, так как разрешение всех запросов, ответы на которые не находятся в кэше или файле зоны какого-либо DNS-сервера нижнего уровня, начинается с обращения к одному из корневых серверов.

Разработчики системы DNS (в начале 80-х годов) понимали это и уже изначально было решено установить 13 корневых серверов, один из которых — сервер A (a.root-servers.net) являлся первичным, а остальные вторичными. Число 13 было выбрано исходя из максимального размера пакета UDP в 512 байтов в то время, сейчас это ограничение снято.

С тех пор число корневых серверов DNS существенно увеличилось — в августе 2013 года их было 376 (рис. 9.10). Это значительно усилило отказоустойчивость и производительность службы DNS, так что даже мощные DDoS-атаки на корневые серверы, два примера которых мы рассмотрим ниже, не смогли привести к ее отказу, а только замедлили работу в некоторых регионах.

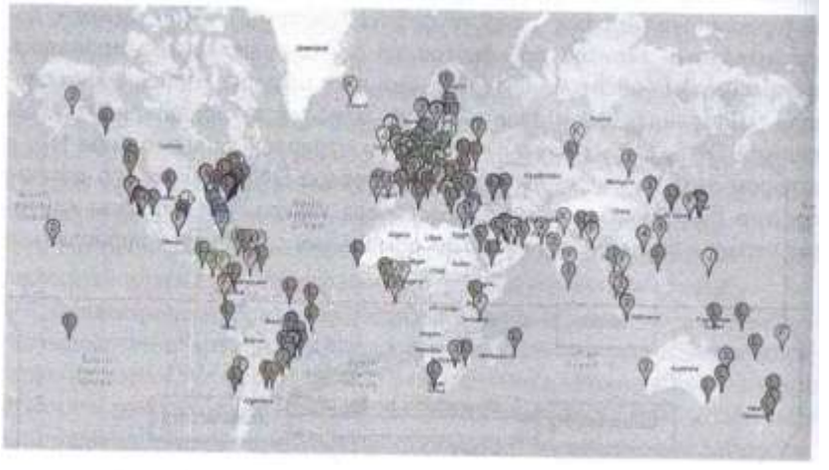


Рис. 9.10. Географическое распределение корневых серверов DNS (источник: root.servers.org)

Несмотря на такое большое количество корневых серверов, они по-прежнему имеют всего 13 имен (от a.root-servers.org до m.root- servers. org), причем каждому имена соответствует один IP-адрес (точнее, один IPv4-aflpec и один IPv6-aflpec).

Каждому имени и адресу (за исключением b.root-servers.net и cl.root-servers.net) соответствует ***кластер*** корневых серверов, например имени

f.root-servers.net соответствует 56 серверов, а имени I.root- servers.net — 146 серверов. Корневые серверы распределены географически, а каждый кластер, соответствующий одному имени, администрируется отдельной организацией.

Доставка запросов к корневым серверам использует ***anycast-***маршрутизацию, то есть доставку пакета любому (any) узлу из группы узлов, имеющей определенный IP-адрес. Техника anycast основана на обычных уникальных IP-адресах, это справедливо как для IPv4, так и для IPv6. При использовании anycast-маршрутизации выбор одного узла из группы обычно осуществляется в соответствии с некоторым правилом предпочтения. Таким правилом может быть расстояние (метрика протокола маршрутизации) от узла-источника до узла-назначения, т. е. пакет доставляется ближайшему к клиенту серверу. Можно также учитывать загрузку серверов группы и направлять паукет самому ненагруженному из них.

Anycast-маршрутизации реализуется за счет специального конфигурирования маршрутизаторов, при этом детали конфигурирования зависят от применяемого правила предпочтения.

Использование техники anycast сулит несколько потенциальных преимуществ при ее применении для серверов некоторого сервиса:

- повышение производительности за счет распараллеливания нагрузки на серверы (баланс нагрузки);
- повышение надежности за счет горячего резервирования серверов, любой сервер может выполнить запрос клиента;
- защиту от DDoS/DoS-атак при рассредоточении серверов группы по различным сетям — чтобы вывести из строя все серверы, атакующему придется проводить одновременную атаку на большое число серверов и сетей, что затруднительно даже для большой армии ботов.

Особенности применения anycast-маршрутизации для службы DNS описаны в RFC 3258 (2002 г.). В нем отмечается, что группа DNS-серверов с одним и тем же anycast-адресом должна находится в одной автономной системе. Информацию о достижимости серверов в таком случае распространяет протокол BGP, в пакетах которого имеются данные о последовательности номеров автономных систем, которые нужно пересечь для достижения некоторого IP-адреса. При наличии альтернативных маршрутов к какому-либо IP-адресу (а к anycast-адресу, принадлежащему группе хостов в разных AS, всегда будет пролегать несколько маршрутов) выбирается тот, у которого число промежуточных AS будет меньше. Таким образом будет выбран тот хост из anycast-группы, который топологически, а часто и географически, ближе к клиенту.

Корневые серверы DNS (и серверы DNS верхнего уровня) широко используют технику anycast для получения выгод всех трех типов, перечисленных выше — производительности, надежности и отражения DDoS-атак.

Наиболее мощными и ощутимыми по своим последствиям были две

DDoS-атаки на корневые серверы, случившиеся 21 октября 2002 года и 6/7 февраля 2007 года. Обычно в этом ряду упоминается еще и опубликованная в феврале 2012 года на сайте pastebin.com угроза группы Anonymous вывести из строя все корневые серверы, но она так и осталась угрозой.

Атаку 21 октября 2002 года характеризуют следующие факты28:

- атака длилась чуть больше часа и была направлена на все 13 адресов корневых серверов:
- атака была комбинированной, использовались атаки ICMP Ping Flood, TCP SYN, фрагментированные IP-пакеты и UDP Flood;
- интенсивность атаки достигала 50-100 Мбит/с на сервер; суммарная интенсивность — 900 Мбит/с;
- в атаке использовался спуфинг IP-адресов источника из различных диапазонов; отследить реальные источники атаки не удалось.

Результаты атаки показали хорошую устойчивость DNS-службы — пользователи замечали только небольшое увеличение времени ожидания при открытии сайта в браузере; все корневые серверы продолжали работать и все ***принятые*** ими запросы были отвечены, но из-за перегрузки входных интерфейсов некоторых серверов не все запросы были приняты.

Еще одним результатом этой атаки было ускорение работ по повышению устойчивости службы DNS — увеличение скорости интерфейсов и линий связи, соединяющих корневые серверы с Интернетом, а также наращивание числа корневых серверов и более равномерное распределение их по автономным системам и географическим регионам.

Атака 6/7 февраля 2007 года длилась 24 часа (поэтому в названии фигурируют две даты). Эта атака была намного мощнее, чем атака 21 октября 2002 года, интенсивность трафика достигала 1 Гбит/с на один пул корневых серверов, но атаковано было только 4 корневых сервера из 13. Подробный анализ атаки вскоре был опубликован ICANN*, но поскольку такие атаки анализировать достаточно сложно, впоследствии в презентации для совещания DNS-OARC (организация, объединяющая специалистов по администрированию DNS) Джон Кристофф внес некоторые коррективы в анализ ICANN.

В атаке было использовано 4500-5000 компьютеров под управлением Microsoft Windows, эти члены ботнета были распределены по сетям нескольких стран. Атака использовала затопление корневых серверов пакетами UDP, направленными на порт 53 (порт DNS), т. е. атака относилась к типу UDP Flood, а использование порта 53 помогало пакетам добраться до серверов, так как у файерволов, защищающих серверы DNS, этот порт всегда открыт, иначе сервер не сможет выполнять свою работу. Атака привела к почти полному исчерпанию пропускной способности двух из четырех атакованных пулов серверов, в то время как остальные два пострадали не так существенно и могли отвечать на большую часть запросов. Fla рис. 9.11 приведены графики последствий атаки для всех 13 корневых серверов;

28 http://d.root-servers.org/october21.txt — отчет специалистов из организаций, администрирующих корневые серверы.

каждая горизонтальная линия соответствует одному из пулов корневых серверов, красные области соответствуют потере сервером более 95 % запросов. Из графика видно, что серверы G (б линия) и L (11 линия) были выведены из строя полностью в течение многих часов, а серверы M(12) и F (5) не могли ответить на 30...40 % запросов в течение примерно двух часов. Необходимо отметить, что на момент атаки серверы G и L не поддерживали anycast, что, вероятно и объясняет выбор их в качестве объектов атаки. Серверы M и F поддерживали anycast, что во многом объясняет их большую устойчивость во время атаки. Всего в начале 2007 года насчитывалось около 100 корневых серверов, т. е. в 3 раза меньше, чем сегодня.

Атака была обнаружена центрами, ответственными за администрирование атакованных пулов корневых серверов, почти немедленно (по алармам самих серверов и данным хостов, выполняющим постоянный мониторинг корневых серверов отправкой на них контрольных запросов). Для уменьшения эффекта атаки были предпринят ряд мер, при этом одной из первых была блокировка любых DNS-запросов, размер которых превысил 300 байтов, так как обычно запрос DNS не превышает 100 байтов, а в атакующих сообщениях размеры поля данных в UDP-потоке доходили до 1023 байтов для усиления эффекта затопления. Однако такая блокировка помогла только частично, так как последующий анализ показал, что размер поля данных трафика атаки менялся случайным образом от 0 до 1023 байтов.

http://www.icann.org/announcements/announcement-08mar07.htm

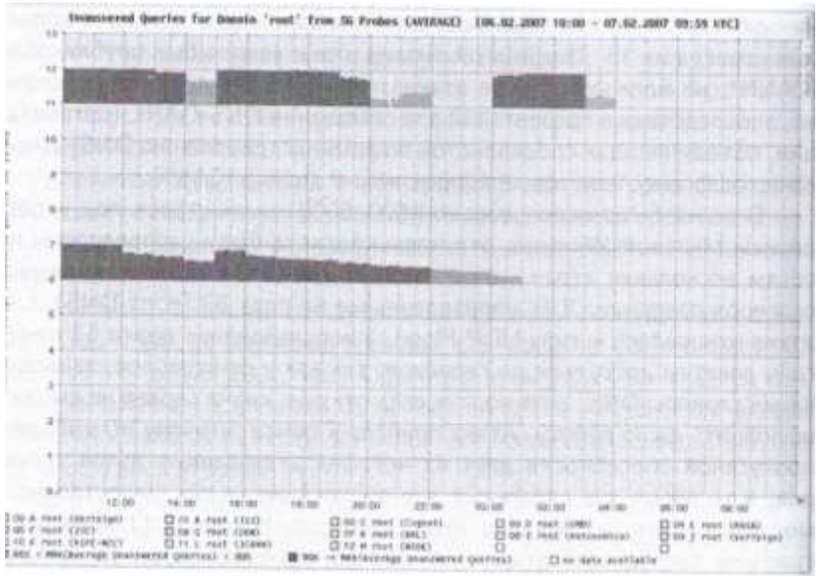


Рис. 9.11. **Количество необслуженных запросов корневыми серверами DNS во время атаки 6/7 февраля 2007 года**

Анализ атаки также показал, что атакующие компьютеры не использовали спуфинг IP-адресов, что дало возможность отследить размещение ботов (в процентном отношении):

- Южная Корея: 65 %;
- США: 19 %;
- Канада: 3,5 %;
- Китай: 2,5 %;
- остальные страны: 10 %.

Хост, координирующий атаку, находился в США, хосты ботнета обращались к нему, используя протокол http.

Причины атаки так и остались неясны; в отчете ICANN предполагается, что это могло быть просто тщеславие хакеров, так как остановить весь Интернет — это звучит красиво, такой «подвиг» представляет вызов для любого хакера.

Мы так подробно разобрали атаку 6/7 февраля 2007 года, потому что она дает хорошее представление о масштабах современных DDoS-атак и о способах защиты от них. Как видно из описания инцидента, защититься от мощной DDoS-атаки очень сложно даже таким опытным специалистам, которые обслуживают корневые серверы DNS. В то же время этот инцидент показал, что такое архитектурное решение, как виртуализация серверов, когда логический сервер представлен большим пулом физических серверов, рассредоточенных по

29 www.blog.cloudflare.com/the-ddos-that-knocked-spamhaus-offline-and-ho

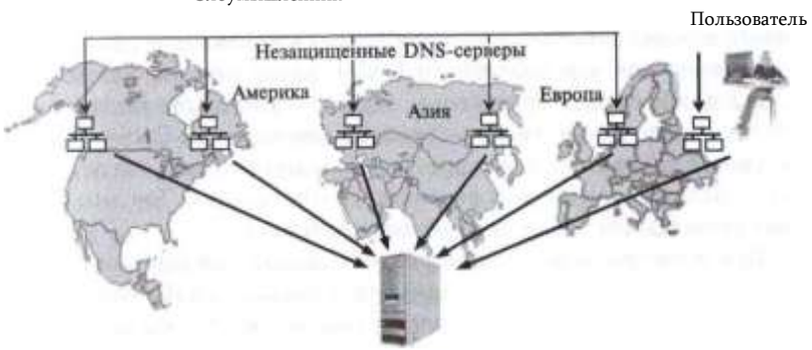


Рис. 9.12. **Атака с использованием отражения от DNS-серверов**

разным сетям и автономным системам, является очень мощным средством защиты, гасящим эффект очень интенсивной DDoS-атаки. Как мы увидим далее, на этом решении построена защита от DDoS-атак провайдеров облачных сервисов, которые широко используют виртуализацию и распределение физических ресурсов по различным сетям.

DDoS-атаки с помощью отражения от DNS-серверов. В Интернете работают миллионы DNS-серверов, и их основной обязанностью является отправка ответов на запросы клиентов. При этом ответ может по объему намного превосходить запрос, например в случае, если запрос относится к передаче файла зоны (запрос типа AXFR) и зона включает большое количество записей.

Очевидно, что инфраструктуру DNS можно не только атаковать, но и использовать как инструмент для атаки типа отражение. И, действительно, такие атаки выполнялись и продолжают выполняться. Рассмотрим схему такой атаки на примере из реальной жизни, благо, компания CloudFlare, провайдер атакуемого клиента, предоставила детальный технический анализ этой атаки на своем сайте29.

Общая схема атаки изображена на рис. 9.12. В марте 2013 года атаке подвергся веб-сервер компании Spamhouse — некоммерческой организации, борющейся со спамом (немного подробнее о ней можно прочитать в главе 21). Для организации атаки было использовано около 30 000 DNS-серверов, работающих в открытом рекурсивном режиме, т. е. отвечающих на запросы любых пользователей и при этом дающих полный (рекурсивный) ответ.

Рекурсивный режим здесь является важным элементов атаки, так как нерекурсивные DNS серверы только перенаправляют запрашивающего на другой DNS-сервер, поэтому их ответ является коротким и не может усилить атаку. Общей практикой является поддержание рекурсивного режима ответов только для «своих» клиентов — сотрудников предприятия для корпоративного DNS-сервера или же подписчиков сервиса для Интернет-провайдера. Однако

http://openresolverproject.org

в Интернете по-прежнему работает около 28 миллионов открытых рекурсивных DNS-серверов, эти данные собрала организация Open Resolver Project*, которая поставила задачу полного искоренения этой практики. Так что найти такие серверы для атаки пока не так уж трудно.

Для усиления атаки использовались запросы на все записи зоны ripe.net (RIPE NCC — региональный информационный Интернет-центр по Европе). Файл зоны ripe.net имеет размер около 3000 байтов, так что при размере запроса в 28 байтов коэффициент усиления составил около 100. Такое мощное усиление позволило создать атаку с общей интенсивностью в 75 Гбит/с, используя поток запросов всего в 2,5 Мбит/с к одному DNS-серверу. Для отдельного DNS-сервера такой поток запросов не является чем-то необычным, так что владельцы этих серверов скорее всего эту атаку не заметили, а вот результирующий поток атаки в 75 Гбит/с вывел веб-сервер компании Spamhouse из строя. Точнее, веб-сервер Spamhouse не работал до тех пор, пока его владельцы не перевели его «под крыло» CloudFlare, провайдера облачных сервисов, к тому же специализирующемуся на защите от DDoS-атак. Перевод помог, так как распределенная виртуальная структура CloudWare, использующая технику anycast и файерволы, смогла абсорбировать большую часть трафика атаки и вебсервер Spamhouse стал вновь доступен пользователям Интернета.

Методы защиты службы DNS

Существует некоторые меры предосторожности, которые повышают защищенность серверов DNS от атак или использования их в качестве инструмента атаки.

Защита ОС хоста. Так как DNS — это приложение ОС, то сама ОС должна быть надежно защищена всеми возможными способами (зашифрованные и сложные для взламывания пароли, доступ с помощью таких протоколов, как SSH, и другие меры, рассматриваемые подробно в части 4).

Разделение пользователей на внутренних и внешних. Предоставление рекурсивных неполномочных ответов только внутренним пользователям, т. е. тем, которые вызывают большее доверие.

Передача файла зоны из первичного сервера только вторичным серверам этой зоны. При передаче необходимо использовать защищенный протокол передачи файлов, например sftp или scp.

Использование DNSSEC. ***DNSSEC*** представляет собой набор стандартов и соответствующих протоколов, обеспечивающих аутентификацию ответов серверов DNS с помощью цифровой подписи и системы публичных ключей. DNSSEC не обеспечивает конфиденциальность данных ответов, они по прежнему передаются в открытом виде, но при использовании DNSSEC клиент может проверить, что полученный ответ действительно пришел от полномочного сервера зоны, а не от сервера, который просто утверждает, что он полномочен, а на самом деле может таковым и не быть. Использование DNSSEC затрудняет злоумышленникам проводить атаки спуфинга и отравления кэша, так как это требует подделки цифровой подписи сервера.

Хотя первые стандарты DNSSEC были разработаны достаточно давно, в 1997 году, его внедрение протекало с большим трудом. Частично это произошло из-за плохой масштабируемости первоначальной версии, а частично — из-за возникновения новых проблем, например проблемы, связанной с возможностью получения полного списка имен зоны с помощью сканирования DNS-сервера, поддерживающего DNSSEC. Тем не менее процесс перехода на DNSSEC идет, с 2010 года все корневые серверы поддерживают DNSSEC, то же происходит и со многими серверами верхнего уровня крупных провайдеров.

Сетевая разведка

Как можно видеть из описания атак на сетевую транспортную инфраструктуру, многие из них требуют предварительных знаний об атакуемой сети и ее хостах. Например, для проведения атаки ICMP Smurf нужно найти промежуточную сеть с большим количеством хостов, отвечающих на ICMP эхо-запросы; кроме того, необходимо убедиться, что эта сеть достижима для пакетов с широковещательным адресом этой сети, посланных из сети злоумышленника, и безусловно, нужно знать IP-адрес атакуемого компьютера. Для атаки ICMP/UDP- затопления нужно знать адреса хостов промежуточной сети, а также номера пассивных портов этих хостов; для атаки ICMP/chargen/echo нужно знать адреса хостов промежуточной сети с активными портами 7 и 19 и так далее.

Если злоумышленник хочет воспользоваться сетью ботов, зараженных вирусом определенного типа, то ему может понадобиться просканировать большое количество компьютеров на отклик по определенному порту, который этот вирус использует для получения команд от контроллера атаки. Дело в том, что вирусы стараются распространиться на возможно большее число компьютеров, но заранее нельзя сказать, будет ли успешно такое внедрение для какого-то определенного хоста или нет, поскольку это зависит от конфигурации

средств защиты и других параметров ОС этого хоста. Поэтому злоумышленник заранее не знает, какие хосты он может использовать в качестве членов сети ботов, а какие нет, даже если это он инициировал распространение этого вируса. А возможно, он просто хочет воспользоваться известным вирусом, распространенным другими лицами, и поэтому ему нужно собрать сведения о зараженных компьютерах.

Администраторы сетей стараются давать минимум информации о сети и ее параметрах — чем меньше у злоумышленника данных о сети, тем сложнее ему ее атаковать. Именно в этом, например, крылась одна из причин медленного внедрения протоколов DNSSEC — они могли использоваться как инструмент получения полного списка имен и адресов хостов некоторого домена, и только с устранением этой возможности внедрение DNSSEC стало массовым.

Поэтому предварительный сбор необходимых для проведения атаки сведений — *сетевая разведка* — является необходимым этапом практически любой атаки. Конкретный набор сведений зависит от типа атаки, но в общем случае «сетевого разведчика» интересуют следующие данные:

- IP-адреса активных (т. е. включенных, отвечающих на сетевой трафик) хостов:
- номера активных TCP-портов хостов;
- номера активных UDP-портов хостов;
- номера пассивных UDP-портов хостов;
- тип и версия ОС;
- тип и версия приложения.

Обнаружение IP-адресов активных хостов сети называют *сканированием сети* (*network scanning*), а активных и пассивных портов — *сканированием портов* (*port scanning).* Человек, устройство или программа, выполняющие сканирование, называются *сканером.* Сам термин «сканирование» говорит о том, что злоумышленник перебирает все возможные IP-адреса некоторой подсети (например, для подсети с маской /24 это 254 значения) или номера портов (65535 для TCP и столько же для UDP). В том случае, когда злоумышленнику нужны узлы, на которых установлено и доступно определенное приложение, например вирус, база данных или сервер DNS, и когда с этой целью он ищет активный хост с открытым определенным портом, говорят о *«прочесывании портов» (port sweep).*

Для сканирования сети и портов злоумышленниками используют более изощренные методы, чем стандартные средства, такие как ping или установление обычного TCP-соединения с каким-либо портом, могут не дать достоверного результата из-за блокировки пакетов файерволом. Например, обычной практикой администраторов корпоративных сетей является запрет ping-запросов к внутренним узлам от любых внешних адресов.

Следующие процедуры могут использоваться для ***сканирования*** *сети.*

Процедура ***TCP SYN ping*** заключается в том, что сканером делается попытка установления TCP-соединения с одним из публично доступных

портов, чаще всего с портом 80 (порт веб-сервиса), который с большой степенью вероятности (но, конечно, не обязательно) открыт для внешнего доступа. Здесь и ниже термин ping в названии процедуры сканирования используется в обобщенном значении, он не связан прямо с утилитой ping, работающей по протоколу ICMP, а только говорит о том, что данная процедура используется с той же целью, что и утилита ping, а именно, тестирует активность хоста с определенным IP адресом. Если в ответ на запрос установления соединения хост отвечает пакетом SYN/ACK, то можно с большой долей уверенности считать, что хост активен.

Процедура ***TCP ACK ping.*** На сканируемый хост посылаются пакеты TCP с случайными значениями номеров портов. В том случае, когда хост активен, он отвечает пакетом TCP с установленным пприз- наком RST (сброс соединения), так как хост считает, что пакет пришел по ошибке — ведь соединение со сканирующим узлом установлено не было. Такой способ позволяет во многих случаях обойти файервол, если тот блокирует выбранный порт, так как обычной практикой конфигурирования файервола является запрет на установление соединения, а вот пакеты уже установленных TCP-соединений, которые отличаются наличием принака АСК и отсутствием признака SYN, пропускаются. В случае, когда тест TCP SYN ping к некоторому порту не проходит, а тест TCP АСК проходит, сканер обычно считает, что данный хост активен, но защищен файерволом, — такая информация также может быть ценной для злоумышленника, так как он может попытаться обойти файервол.

Процедура ***UDP ping.*** На тестируемый хост направляется UDP- пакет с номером порта, который с большой вероятностью является пассивным у активного компьютера. Например, это может быть порт 40125, который не связан ни с одним из популярных приложений и по этой причине используется программой NMAP, предназначенной для аудита сети. В случае, когда компьютер активен, а порт пассивен, сканер получает в ответ ICMP сообщение «Порт недоступен»; если же компьютер отключен, то сканер получает сообщение «Хост недоступен». Если же и компьютер и порт оказываются активными, то никакого ответа сканер не получает и определенного вывода о состоянии хоста он сделать не может; в этом причина выбора пассивного

порта. Эффективность применения UDP объясняется тем, что многие файерволы по умолчанию не блокируют трафик UDP.

Процедура ***ICMP timestamp ping.*** Хост тестируется запросами синхронизации времени протокола ICMP (код 13). Так как администраторы сетей чаще всего блокируют только ICMP эхо-запросы, то этот тип ICMP запросов доходит до хоста и в случае его активного состояния инициирует ответ. Вместо запросов синхронизации времени можно также использовать запросы длины маски IP-адреса (код 17) или информационные запросы (код 15).

Процедура ***IP ping.*** На исследуемый компьютер направляется IP- пакет с кодом вложенного протокола, отличным от кодов протоколов TCP, UDP или ICMP. Скорее всего, такой тип протокола не поддерживается стеком TCP/IP данного компьютера, и активный хост ответит ICMP сообщением «Протокол недостижим».

Очень похожие методы применяются и для сканирования портов. Здесь предпочтение отдается TCP SYN-сканированию, так как это самый быстрый способ. Производительность в данном случае очень важна, так как в отличие от сканирования хостов проверить нужно 65535 TCP-портов и столько же UDP-портов.

Сканирование портов злоумышленниками осуществляется с помощью специальных программных средств, многие из которых используются сетевыми администраторами для инвентаризации сети и аудита ее защищенности. Существует большое количество программных средств аудита средств защиты, как коммерческих, так и с открытым кодом. Среди наиболее популярных продуктов следует отметить уже упомянутую выше систему аудита сети ***NMAP,*** эта программа с открытым кодом позволяет выполнять сканирование сети и портов на основе довольно большого количества различных процедур. Коммерческая программная система сканирования уязвимостей ***Nessus*** является функционально более мощной, чем NMAP, к тому же существует ее бесплатная версия для персонального использования в домашних сетях.

Сканирование сети и портов обычно не проходит незамеченным — очень вероятно, что средства протоколирования событий ОС и файерволов зафиксируют этот процесс и администратор сканируемой сети начнет расследовать инцидент, основываясь на адресе компьютера, который выполнял сканирование. Чтобы избежать раскрытия, злоумышленники часто используют спуфинг IP-адреса при атаках, но на первый взгляд кажется, что в сетевой разведке этот прием не может сработать, так как злоумышленнику нужно получать ответы на свой компьютер, иначе как узнать результаты? Тем не менее сокрытие IP-адреса источника атаки возможно и при сканировании. Одним из приемов является маскировка его среди множества других адресов В этом случае тестовые сканирующие пакеты отправляются с действительного IP-адреса наряду со множеством таких же пакетов, но с поддельными адресами. Расчет здесь на то, что при расследовании факта

сканирования трудно будет установить, кто являлся истинным организатором сканирования, а кого просто использовали как маскировку. Более эффективным является так называемое «пустое» сканирование, когда истинный адрес никогда не используется, а результаты сканирования пытаются понять по реакции третьего компьютера, чей адрес подделывается. Это достаточно тонкая техника, требующая к тому же возможности анализа трафика этого третьего компьютера.

Часто возникает вопрос — является ли сканирование портов уголовно наказуемым? На него нет однозначного ответа, так как он зависит от законодательства конкретной страны, а также обстоятельств инцидента. Например, в США нет закона, который бы явно перечислял сканирование сетей и портов в списке киберпреступлений, тем не менее в 1999 году компания VC3 подала в суд на Скота Моултона за то, что тот сканировал порты веб-сервера, администрируемого этой компанией, и тем самым якобы нарушил закон США о компьютерных преступлениях. Суд принял дело к производству, и судебное разбирательство заняло почти год, что говорит о неоднозначности таких дел. В конце концов Скота Моултона оправдали, так как он сканировал порты с добрыми намерениями — он был администратором компьютерной системы службы 911 графства Чероки и при подключении нового соединения с внешним миром просто хотел проверить, достаточно ли надежно защищена его система; вот только ему не повезло в том, что один из веб-серверов в его сети находился под управлением третьей стороны, которую он не предупредил.

В законодательстве РФ также нет прямого указания на то, что сканирование портов является преступлением, однако его можно квалифицировать как попытку совершения некоторого другого преступления, например нанесения имущественного или репутационного ущерба. В то же время есть страны, где законы, описывающие неправомерный доступ к компьютерным системам, настолько строги, что суду достаточно просто будет посчитать сканирование портов таким видом доступа. К таким странам относятся Великобритания, Израиль и некоторые другие.

Правильнее всего будет рассматривать уместность или легальность сканирования чужой сети с позиций этики. В литературе по сетевой безопасности часто используется такая аналогия — сканирование портов можно сравнить с обходом чужого частного дома, тщательным рассматриванием его внутреннего двора с играющими

детьми через щели в заборе и фотографированием дома с разных точек. С точки зрения законодательства это не преступление, но такое поведение очень похоже на его подготовку или же просто демонстрирует плохие манеры и бесцеремонность случайного любопытного прохожего, т. е. неэтично. Так и при сканировании — если вам нужно его выполнить по долгу службы и вы предполагаете, что вы, возможно случайно, можете затронуть системы, администрируемые не вами, то лучше заранее предупредить администратора этих систем и проводить сканирование, получив согласие.

Вопросы к главе 9

1. Что из перечисленного ниже является элементом транспортной инфраструктуры сети:

а) маршрутизатор:

б) сервер баз данных;

в) сервер DNS;

г) транспортные средства ОС;

д) линия связи.

2. Почему транспортная инфраструктура сети является заманчивой целью для злоумышленников?

а) она открыта для атак, так как провайдера не защищают свои маршрутизаторы;

б) она может усилить атаку во много раз за счет отражения пакетов атакующего;

в) каждый может атаковать каждого;

г) протокол IP передает пароли в открытом виде.

3. Что из перечисленного некорректно описывает протокол IP:

а) IP-адреса уникальны в пределах сети провайдера;

б) IP-адреса распределяются между провайдерами централизованно;

в) узлу сети, имеющему адрес Ethernet, не нужен IP-адрес для работы в Интернет.

4. В чем состоит главная уязвимость протокола IP?

а) заголовок пакета IP переносит адреса источника и назначения в незашифрованном виде;

б) он использует широковещательные адреса назначения;

в) он позволяет каждому узлу Интернет взаимодействовать с каждым без предварительного установления соединения;

г) он поддерживает фрагментацию пакетов.

5. В функции протокола TCP входит следующее:

а) восстановление потерянных пакетов;

б) обеспечение целостности данных;

в) мультиплексирование и демультиплексирование пользовательских данных;

г) аутентификация хостов.

6. С помощью протокола ICMP злоумышленник может:

а) определить, что некоторый хост находится в работоспособном состоянии;

б) организовать DoS-атаку;

в) перенаправить маршрут;

г) узнать, через какие промежуточные маршрутизаторы проходит маршрут до некоторого конечного узла.

7. В том случае, когда сервер DIMS вашей сети по какой-то причине перестал работать, вы можете продолжить работу с сайтами Интернета:

а) задав в качестве адреса сервера DNS в конфигурации стека TCP/IP вашего компьютера адрес любого другого сервера DNS;

б) без проблем, поскольку это обстоятельство не влияет на вашу работу с сайтами Интернета;

в) задав в качестве адреса сервера DNS в конфигурации стека TCP/IP вашего компьютера адрес любого другого открытого сервера DNS;

г) обращаясь к хостам по IP-адресам.

8. Как работает атака SYN Flood:

а) эта атака является взломом протокола TCP;

б) эта атака пользуется протоколом TCP как инструментом;

в) эта атака пытается исчерпать пропускную способность интерфейса.

9. Что является признаком атаки SYN Flood?

а) наличие флага SYN в пакетах, идущих от атакуемого сервера;

б) наличие флага SYN и отсутствие флага ACK в пакетах, идущих к атакуемому серверу;

в) отсутствие флага ACK в пакетах, идущих от атакуемого сервера;

г) наличие флагов SYN и ACK в пакетах, идущих от атакуемого сервера.

10. С какой целью злоумышленник должен подавить отправку ACK-сегментов на атакуемый сервер в ходе атаки SYN Flood?

а) для того чтобы скрыть свой IP-адрес;

б) для того чтобы открытые соединения оставались незавершенными;

в) для того чтобы закрыть открытые соединения.

11. Для чего применяется техника «Проверка обратного пути»?

а) для борьбы со спуфингом IP-адреса отправителя;

б) для борьбы со спуфингом IP-адреса получателя;

в) для борьбы с подделкой IP-адреса отправителя;

г) для отправки блокирующих сообщений на компьютер злоумышленника.

12. В чем заключается идея механизма SYN cookie:

а) браузер злоумышленника прекращает атаку, получив от атакуемого сервера SYN cookie;

б) атакуемый хост не запоминает состояние устанавливаемого соединения, поэтому атака не имеет смысла;

в) компьютер злоумышленника терпит крах, получив в ответ на атакующие пакеты SYN cookie.

13. Соединения какого типа проще использовать для атаки «Подделка TCP сегмента»:

а) длительные соединения протокола TCP;

б) кратковременные соединения протокола TCP;

в) соединения протокола TCP с номерами портов меньше 1024;

г) соединения протокола TCP с номерами портов свыше 1024.

14. Какими средствами можно предотвратить атаки «Повторение сегментов» и «Сброс соединения»?

а) принять все меры по предотвращению прослушиванию трафика;

б) применить технику цифровой подписи сегментов TCP MD5;

в) применить технику цифровой подписи сегментов TCP AO;

г) ни одним из перечисленных.

15. Каким образом можно направить трафик по ложному маршруту?

а) с помощью ложного ARP-ответа;

б) с помощью ICMP-сообщения «Перенаправление маршрута»;

в) с помощью IP-сообщения «Сеть недоступна»;

г) используя DNS-спуфинг.

16. Каким образом можно предотвратить атаку ICMP Smurf?

а) блокировать входящие в вашу сеть эхо-ответы;

б) отключить поддержку широковещания маршрутизаторами вашей сети;

в) запретить компьютерам вашей сети реагировать на широковещательные эхо- запросы;

г) нет нужды предпринимать меры, так как такая атака имеет только исторический интерес.

17. Атака Ping of Death приводит к:

а) краху протокола ICMP на атакуемом компьютере;

б) отказу сетевого интерфейса маршрутизатора;

в) тяжелой болезни администратора сети;

г) краху ОС компьютера.

18. Почему с атаками, использующими протокол UDP, сложнее бороться, чем с атаками, использующими протокол TCP?

а) протокол UDP имеет более сложную логику, а следовательно, он более уязвим;

б) пакеты UDP не имеют ограничения по длине поля данных;

в) протокол UDP не использует механизм обратной связи, поэтому атакуемый компьютер не имеет средств уменьшить интенсивность потока;

г) флаги заголовка UDP могут вызвать переполнение стека атакуемой ОС;

д) это утверждение неверно,

19. Существуют следующие атаки на протокол IP;

а) IP-фрагментация версии IPv4;

б) IP-фрагментация версии IPv6;

в) IP-опции версии IPv4;

г) IP-опции версии IPv6;

д) IP-заголовки IPv4;

е) IP-заголовки IPv6.

20. Чем атака «DNS-спуфинг» отличается от атаки «Отравление DNS-кэша»?

а) ничем, это разные названия одной и той же атаки;

б) в результате атаки «Отравление DNS-кэша» DNS-сервер терпит крах, а атака «DNS-спуфинг» к краху сервера не приводит;

в) в атаке DNS-спуфинг ложный ответ предается клиенту, а в атаке «Отравление DNS-кэша — серверу DNS;

г) в атаке DNS-спуфинг ложный ответ передается серверу DNS. а в атаке «Отравление DNS-кэша» — клиенту.

21. Можно ли использовать систему DNS для атаки затопления?

а) да;

б) нет.

22. Какие из приведенных ниже параметров пытаются определить злоумышленники с помощью сканирования сети:

а) IP-адреса активных хостов;

б) номера активных TCP-портов активных хостов;

в) номера активных UDP-портов хостов;

г) номера пассивных UDP-портов хостов;

д) тип и версию ОС.

10 *ФИЛЬТРАЦИЯ И МОНИТОРИНГ ТРАФИКА*

Фильтрация трафика и файерволы

Типы фильтрации трафика

При описании способов защиты от атак на транспортную инфраструктуру сети мы часто упоминали такой прием, как ***фильтрация трафика.*** Например, атаку Ping flood можно предотвратить, если фильтровать, т. е. отбрасывать, не пропускать, пакеты эхо-запросов ICMP. A DDoS-атаку DNS 6/7 февраля 2007 года удалось смягчить фильтрацией пакетов с размерами более 300 байтов. Условия фильтрации бывают самыми разными, и не всегда удается найти простой признак, по которому одни пакеты нужно пропускать, а другие — отбрасывать. К тому же такое условие почти всегда является компромиссом между предотвращением атаки и основной функциональностью защищаемого узла — чем больше потенциальных атак мы предотвратим, тем больше мы урезаем функции нормальной работы узла TCP/IP. Так, в случае, когда блокируются эхо-запросы ICMP, администратор сети лишается возможности проверить достижимость узла удаленно, извне зоны защиты файервола, фильтрующего эхо-запросы. А в случае упомянутой выше DDoS-атаки администратор не стал для защиты атакуемого сервера DNS блокировать доступ к его порту 53, так как при этом сервер полностью потерял бы свою функциональность, а ограничился полумерами, фильтруя только подозрительно большие пакеты, которые вряд ли могли соответствовать реальным запросам.

Иногда эффективная фильтрация требует анализа не одного, а некоторой последовательности пакетов. Например, для того чтобы распознать атаку TCP SYN, недостаточно принимать во внимание только признаки одного пакета, взятого в отдельности от остальных. В этом случае мы не сможем отличить нормальный запрос на установление TCP-соединения от атаки — на что и рассчитывает злоумышленник. Признаком TCP SYN атаки является большое количество TCP SYN пакетов от некоторого источника без соответствующего количества TCP ACK-пакетов от того же источника, поэтому для

обнаружения этой атаки нужно запоминать и анализировать достаточно длинные последовательности пакетов. В этом случае говорят о *stateful-фильтрации,* т. е. фильтрации на основе *запоминания* состояния трафика. В случае, когда правила фильтрации учитывают только признаки отдельного пакета, говорят о *statless-фильтрации,* или фильтрации *без запоминания* состояния трафика, фильтрации без памяти.

Можно также классифицировать фильтрацию *по уровню стека протоколов,* на котором анализируются заголовки и данные пакетов. В этой главе мы рассматриваем фильтрацию протоколов, на которых основано функционирование транспортной инфраструктуры сети, к ним, прежде всего, относятся протоколы межсетевого и транспортного уровня IP, ICMP, TCP и UDP, а также протоколы прикладного уровня DNS и BGP.

Реализуют фильтрацию трафика устройства, называемые *файерволами,* определение, принципы работы и основные функции которых были рассмотрены в главе 8.

Файерволы на основе маршрутизаторов

Протоколы IP-маршрутизации создают таблицы маршрутизации, на основе которых сообщение любого узла составной сети может быть доставлено любому другому узлу, и каждый пользователь Интернета может получать доступ к любому публичному сайту.

Однако такая всеобщая достижимость узлов и сетей не всегда отражает потребности их владельцев. Поэтому многие маршрутизаторы поддерживают развитые средства фильтрации пользовательского трафика, а также фильтрации объявлений протоколов маршрутизации, что позволяет дифференцированно управлять достижимостью узлов.

Фильтрация пользовательского трафика

Условия (правила) фильтрации маршрутизаторов могут строиться на основе разнообразных признаков, например это могут быть:

- IP-адреса источника и приемника;
- MAC-адреса источника и приемника;
- идентификатор интерфейса, с которого поступил пакет;
- тип протокола, сообщение которого несет IP-пакет (т. е. TCP, UDP, ICMP или OSPF);
- номер порта TCP/UDP (т. е. тип протокола прикладного уровня).

При наличии фильтра маршрутизатор сначала проверяет совпадение условия, описанного этими фильтром, с признаками пакета и при положительной проверке выполняет над пакетом ряд действий — как стандартную передачу на некоторый выходной интерфейс в соответствии с записью таблицы маршрутизации, так и одно из нестандартных действий. Например, пакет может быть отброшен (drop); направлен к | ледующему маршрутизатору, отличающемуся от того, который ука- >Зн в таблице маршрутизации; помечен как вероятный кандидат на отбрасывание при возникновении перегрузки.

Рассмотрим примеры фильтров, написанных на командном языке маршрутизаторов Cisco. Эти фильтры, называемые ***списками доступа*** *(access list),* являются очень распространенным средством ограничения пользовательского трафика в 1Р-маршрутизаторах.

Наиболее простым является ***стандартный список доступа; в*** нем в качестве условия фильтрации указывается только IP-адрес источника. Общая форма такого условия выглядит следующим образом:

access-list номер_списка_доступа {deny | permit}
(адрес.источника [метасимволы.источника] | any}

Стандартный список доступа определяет два действия с пакетом, который удовлетворяет описанному в фильтре условию: отбросить (deny) или передать для стандартной обработки в соответствии с таблицей маршрутизации (permit). Условием выбора того или иного действия в стандартном списке доступа является совпадение IP- адреса источника пакета с адресом источника, заданным в списке. Совпадение проверяется в том же стиле, что и при проверке таблицы маршрутизации, при этом ***метасимволы*** являются аналогом маски, но в несколько модифицированном виде. Двоичный нуль в поле метасимволов источника означает, что требуется совпадение значения этого разряда в адресе пришедшего пакета и в адресе, заданном в списке доступа. Двоичная единица означает, что совпадения в этом разряде не требуется. Практически, если вы хотите задать условие для всех адресов некоторой подсети, то должны использовать инвертированное значение маски этой подсети. Параметр any означает любое значение адреса — это просто более понятная и краткая форма записи значения 255.255.255.255 в поле метасимволов источника.

Пример стандартного списка доступа: access-list 1 deny 192.78.46.0 0.0.0.255 Здесь 1 — номер списка доступа; ***deny —*** действие с пакетом, который удовлетворяет условию данного списка доступа; 192.78.46.0 — адрес источника; 0.0.0.255 — метасимволы источника. Этот фильтр запрещает передачу пакетов, у которых в старших трех байтах адреса источника имеется значение 192.78.46.0.

Список доступа может включать более одного условия. В этом случае он состоит из нескольких строк с ключевым словом ***access-list*** и одним и тем же номером списка доступа. Так, если мы хотим разрешить прохождение через маршрутизатор пакетов хоста 192.78.46.12,

запрещая передачу пакетов одному из хостов сети 192.78.46.0/24, то список доступа будет выглядеть следующим образом:

access-list 1 permit 192.78.46.12 0.0.0.0

access-list 1 deny 192.78.46.0 0.0.0.255

access-list 1 permit any

Условия списка доступа проверяются по очереди, если какое-либо из них дает совпадение, то выполняется действие ***permit*** или ***deny,*** определенное в этом условии. После этого остальные условия списка уже не проверяются. Считается по умолчанию, что в конце каждого списка имеется неявное условие вида:

[access-list 1 deny any]

Однако, если все же требуется пропускать все пакеты, не определенные явно в условиях, необходимо добавить в последней строке условие

access-list 1 permit *any*

Список доступа можно применять к любому интерфейсу маршрутизатора и в любом направлении: если список применяется с ключевым словом in, то он действует на входящие в интерфейс пакеты. В этом случае говорят, что выполняется ***входная фильтрация*** (ingress filtering).

Например, написанный нами список доступа 1 можно применить к некоторому интерфейсу для обработки входящего трафика, используя следующую команду:

access-group 1 in

Если же применить список доступа с ключевым словом out, то он будет воздействовать на пакеты, выходящие из интерфейса, в этом случае будет выполняться ***выходная фильтрация (egress filtering).***

Существует также и более мощные типы списков доступа для маршрутизаторов Cisco, например ***расширенные списки доступа.***

С помощью расширенной фильтрации можно блокировать или разрешать прохождение пакетов не только по адресу источника, но и по адресу назначения, а также по определенным номерам портов TCP/UDP источника и назначения, а следовательно, управлять доступом к определённым приложениям, которые используют эти порты. Обычно серверная часть приложения использует фиксированный и известный порт (например, за веб-сервером закреплен TCP порт 80), именно он и используется в списке доступа. Узнать номер порта приложения можно из данных IANA, опубликованных в RFC 6335.

Общий формат расширенного списка доступа Cisco следующий: access-list номер.спискадцоступа { deny | permit }
{ protocol | ключевое_слово_протокола }
{ адрес „источника [метасимволы_источника]] [портлс- точника] | any }
[адрес„приемника [метасимволылриемника]] [портлри- емника]

Пользуясь расширенными списками доступа, можно запретить прохождение во внутреннюю сеть предприятия FTP-пакетов. Как известно, служба FTP использует для приема запросов от клиентов протокол TCP с хорошо известным портом 21. Для этого в список доступа нужно включить условие

access-list 102 deny TCP any any 21 Затем можно применить его к интерфейсу маршрутизатора, к которому подключена внутренняя сеть, с ключевым словом out.

Заметим, что клиентскую часть приложения защищать сложнее, так как она использует номер порта из диапазона 1024-65535, причем этот номер не фиксирован, он назначается операционной системой динамически при старте приложения. Для клиентской части приложений, использующих TCP, маршрутизаторы Cisco обеспечивают защиту при использовании ключевого слова established. Это слово говорит о том, что имеются в виду сегменты TCP уже установленного (established) соединения. Признаком таких сегментов является наличие бита ACK= 1 в пакете.

Рассмотрим пример записи списка доступа:
access-list 105 permit tcp any eq 80 any gtl023
established

Если это условие применить во входном направлении (т. е. от внешней сети ко внутренней), то оно приведет к тому, что все клиенты внутренней сети (они выделены параметром gtl023, задающим порты с номерами, большими 1023) могут получать ответы от внешних веб-серверов (выделенных параметром eq 80), но извне невозможно установить соединение с этими клиентами, потому что запрос на установление соединения не содержит бит ACK= 1 (а только SYN= 1).

Администраторы корпоративных сетей из соображений безопасности часто запрещают возможность сканирования извне внутренних хостов с помощью утилиты ping. Это можно сделать с помощью условия

access-list 101 deny ICMP any 192.78.46.8 0.0.0.0 eq 8

Как видно из условия, его синтаксис для протокола ICMP несколько отличается от общего синтаксиса расширенных списков доступа. Параметр eq 8 означает, что запрещается передача 1CMP-со-

общений типа 8, соответствующего эхо-запросам, с помощью которых функционирует утилита ping.

Протоколирование событий, связанных с фильтрацией пакетов, необходимо для обеспечения такого важного свойства безопасной системы, как подотчетность.

Как уже было сказано, часто важнее определить сам факт атаки, чем заблокировать ее. Маршрутизаторы Cisco могут помещать сообщения об обработке пакетов, удовлетворяющих условию некоторой записи списка доступа в системный журнал маршрутизатора ***syslog.*** По умолчанию такая опция для каждой записи списка доступа неактивна, это сделано для уменьшения нагрузки на маршрутизатор. Для активизации протоколирования необходимо добавить к записи ключевое слово log, например:

access-list 102 permit TCP any 21 any log

После этого каждое событие, связанное с тем, что маршрутизатор обработал пакет, удовлетворяющий записи 102, вызовет появление в системном журнале syslog нового сообщения следующего типа:

*May 1 22:12:13.243: "/.SEC-6-IPACCESSL0GP:

list ACL-IPv4-E0/0-IN permitted tcp 192.168.1.3(1024)

-> 192.168.2.1(22), 1 packet

Отметим, что приведенный здесь пример языка списка доступа маршрутизаторов Cisco является хотя и фирменной, но достаточно типичной реализацией функций файервола в IP-маршрутизаторе. Отсутствие в перечне этих функций ***фильтрации с запоминанием состояния*** связано со стремлением не создавать слишком большую нагрузку на маршрутизатор и «не отвлекать» его от выполнения основных обязанностей. Это ограничение является главным отличием маршрутизаторов от программных и программно-аппаратных файерволов. Конечно существуют и исключения из этого правила, например модели Juniper SRX, в полном объеме поддерживающие все функции «нормального» маршрутизатора, способны выполнять фильтрацию с запоминанием состояния,

Фильтрация маршрутных объявлений. Фильтрация маршрутных объявлений помогает защитить IP-маршрутизацию от непреднамеренных ошибок администраторов и атак злоумышленников. Особенно важна такая фильтрация для протокола BGP, который распространяет данные о достижимости той или иной сети через последовательность автономных систем Интернета и тем самым обеспечивает его связность.

Посмотрим на примере маршрутизаторов Cisco, каким образом выполняется фильтрация маршрутных объявлений BGP.

30 www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml

Фильтровать маршрутные объявления BGP можно как на основе префиксов IP-адресов, так и на основе номеров автономных систем.

Синтаксис списков доступа на основе префиксов весьма прост, например запись

ip prefix-list abcl deny 10.0.0.0/8

запрещает объявления о префиксе 10.0.0.0/8, который относится к диапазонам адресов, зарезервированных IANA в качестве частных (это определено в RFC1918), поэтому хорошей практикой является фильтрация их в публичном адресном пространстве Интернета. Для того чтобы список доступа начал работать, его нужно применить к конфигурации BGP для определенного соседа и указать направление фильтрации, например:

switch# configure terminal

switch(config)# ip prefix-list allowprefix 10 permit 192.0.2.0 **eq** 24 switchCconfig)# ip prefix-list allowprefix 20 permit 209.165.201.0 **eq** 27 switchCconfig) router bgp 65536:20 switchCconfig-router)# neighbor 192.0.2.1/16 remote-as 65536:20 switchCconfig-router-neighbor)# address-family ipv4 unicast switchCconfig-router-neighbor-af)# prefix-list allowprefix in switchCconfig-router-neighbor-af)#

В этом примере в списке доступа allowprefix созданы две записи, 10 и 20, разрешающие префиксы 192.0.2.0.24 и 209.165.201.0/27, а затем этот список применен к BGP-соседу 192.0.2.1 во входном направлении, т. е. от этого соседа разрешено принимать только эти префиксы.

В словаре администраторов Интернет-провайдеров закрепился такой термин, как ***«богоны»*** (***bogons***). Он обозначает префиксы, которые не должны появляться в публичном адресном пространстве Интернета. Богоны включают адреса, зарезервированные в RFC 1918 как частные, но не только — в них входят также адреса, еще не распределенные между региональными Интернет-центрами (такими, как RIPE NCC, ARIN и т. д.) или же зарезервированные для специальных целей. IANA публикует список распределенных адресов на своем сайте30, этот список периодически обновляется, так что провайдер должен отслеживать изменения и корректировать свой список богонов, использующихся в списках доступа.

Синтаксис списков доступа на основе номеров автономных систем более гибок, так как он позволяет использовать регулярные выражения в стиле Unix. Рассмотрим следующий пример:

```
router bgp 200
 neighbor 193.1.12.10 remote-as 100 neighbor
193.1.12.10 filter-list 1 out neighbor 193.1.12.10 filter-
list 2 in ip as-path access-list 1 permit .109. ip as-path
access-list 1 permit *117. ip as-path access-list 2 permit
.200$ ip as-path access-list 2 permit “100$
```

Здесь список доступа 1 разрешает маршрутные объявления, содержащее

номер AS 109 в любом месте списка номеров объявления (символ «_» определяет это), а также объявления от соседа с номером 117 (символ «"» обозначает начало выражения). Список доступа 2 разрешает объявления, в которых исходной автономной системой является AS200 (символ «$» относится к концу строки), а также объявления от непосредственного соседа AS100, в которых он объявляет себя исходной AS (так как номер 100 должен быть первым и последним в последовательности AS, а значит — единственным).

Влияние динамического назначения адресов (DHCP) на работу файервола. Для нормальной работы сети каждому сетевому интерфейсу компьютера и маршрутизатора должен быть назначен 1P-адрес.

Процедура присвоения адресов происходит в ходе ***конфигурирования*** компьютеров и маршрутизаторов. Назначение IP-адресов может происходить вручную в результате выполнения процедуры конфигурирования интерфейса либо автоматически с помощью ***протокола динамического конфигурирования хостов (Dynamic Host Configuration Protocol, DHCP).*** Этот протокол работает в локальных сетях, так как основан на широковещательных запросах клиентов к DHCP-серверам.

В сети, где адреса назначаются динамически, нельзя быть уверенным в адресе, который в данный момент имеет тот или иной узел. И такое непостоянство IP-адресов влечет за собой некоторые проблемы, прежде всего в работе DNS-серверов и файерволов. Как мы видели в предыдущих разделах, фильтрация трафика во многих случаях происходит на основе IP-адреса защищаемого хоста, а при использовании DHCP этот адрес выдается хосту временно, например на сутки, а затем процедура назначения адреса повторяется, и нет никакой гарантии, что этот адрес останется тем же. Поэтому в локальной сети, защищенной файерволом и использующей протокол DHCP, возможны два подхода:

- применение DHCP только для конфигурирования клиентских компьютеров, для которых обычно не требуется отображение доменного имени в IP-адрес, и конфигурирование серверов с постоянными, ***статическими 1P-адресами***
- применение DHCP и для-клиентов, и для серверов, но настройка сервера DHCP таким образом, чтобы он для серверов создавал

статические записи DHCP, в которых MAC-адрес запроса связывается с IP-адресом ответа сервера. При поступлении запроса с MAC-адресом, имеющимся в какой-либо статической записи сервера DHCP, последний помещает в ответ IP-адрес из этой статической записи, а не из динамического пула IP-адресов. Преимущество статических записей DHCP перед статическим конфигурированием IP-адресов у серверов, в первом случае, заключается только в централизации базы данных таких адресов, когда они собраны в одном месте и их можно редактировать с помощью удобного интерфейса сервера DHCP.

Файерволы с функцией NAT

Одной из функций файервола является ***трансляция сетевых адресов (Network Address Translation, NAT).*** В этом случае фильтрация трафика заключается не в пропуске или отбрасывании пакетов, а в замене внешнего IP-адреса пакета, который использовался при маршрутизации пакета через Интернет, на внутренний, который используется для маршрутизации во внутренней сети, корпоративной или персональной.

Сегодня существуют две причины использования технологии NAT: одна из них — дефицит IPv4 адресов, а другая — скрытие адресов хостов для повышения безопасности сети. И в том, и в другом случаях внутренняя сеть использует частные адреса (RFC 1918), которые заменяются на один или несколько публичных адресов при отправке пакетов во внешние сети.

Применение NAT позволяет скрыть адреса узлов своей сети, чтобы не дать возможности злоумышленникам составить представление о структуре и масштабах корпоративной сети, а также о структуре и интенсивности исходящего и входящего трафиков.

Технология NAT стояла у истоков зарождения файерволов как отдельного класса продуктов. В начале 90-х годов, когда дефицит IPv4 адресов еще мало ощущался, несколько специалистов основали компанию Network Translation и разработали программный продукт PIX, который позволял транслировать сетевые адреса. Позднее эту компанию приобрела компания Cisco, а программный продукт стал знаменитым Cisco PIX Firewall, одним из флагманов этого класса средств защиты.

Традиционная технология NAT. Технология трансляции сетевых адресов имеет несколько разновидностей, наиболее популярная из которых — ***традиционная технология трансляции сетевых адресов*** — позволяет узлам из частной сети прозрачным для пользователей образом получать доступ к узлам внешних сетей. Подчеркнем, что в

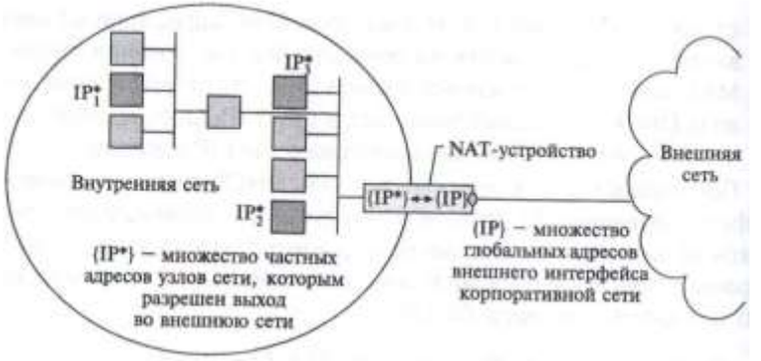


Рис. 10.1. Схема действия традиционной технологии NAT

данном варианте NAT решает проблему организации только тех сеансов связи, которые *исходят* из частной сети. Направление сеанса в данном случае определяется положением инициатора: если обмен данными инициируется приложением, работающем на узле внутренней сети, то сеанс называется исходящим несмотря на то, что в его рамках в сеть могут поступать данные извне31.

Идея технологии NAT состоит в следующем. Пусть сеть предприятия образует тупиковый домен, узлам которого присвоены частные адреса (рис. 10.1). На маршрутизаторе, связывающем сеть предприятия с внешней сетью, установлено программное обеспечение NAT. Это NAT-устройство динамически отображает набор частных адресов {IP*} на набор глобальных адресов {IP}, полученных предприятием от поставщика услуг и присвоенных внешнему интерфейсу маршрутизатора предприятия.

Важным для работы NAT-устройства является правило распространения маршрутных объявлений через границы частных сетей. Объявления протоколов маршрутизации о внешних сетях «пропускаются» пограничными маршрутизаторами во внутренние сети и обрабатываются внутренними маршрутизаторами. Обратное утверждение неверно — маршрутизаторы внешних сетей не получают объявлений о внутренних сетях, объявления о них отфильтровываются при передаче на внешние интерфейсы. Поэтому внутренние маршрутизаторы «знают» маршруты ко всем внешним сетям, а внешним маршрутизаторам ничего не известно о существовании частных сетей.

Традиционная технология NAT подразделяется на две технологии:

- *базовая трансляция сетевых адресов (Basic Network Address Translation, Basic NAT),* в которой для отображения используются только IP-адреса;
- *трансляция сетевых адресов и портов (Network Address Port Translation, NAPT),* в которой для отображения наряду с IP-адресами используются

31 Традиционная технология NAT в виде исключения допускает сеансы обратного направления, заранее выполняя статическое взаимно однозначное отображение внутренних и внешних адресов для некоторого ограниченного набора узлов.

еще и так называемые *транспортные идентификаторы,* в качестве которых чаще всего выступают TCP- и UDP-порты.

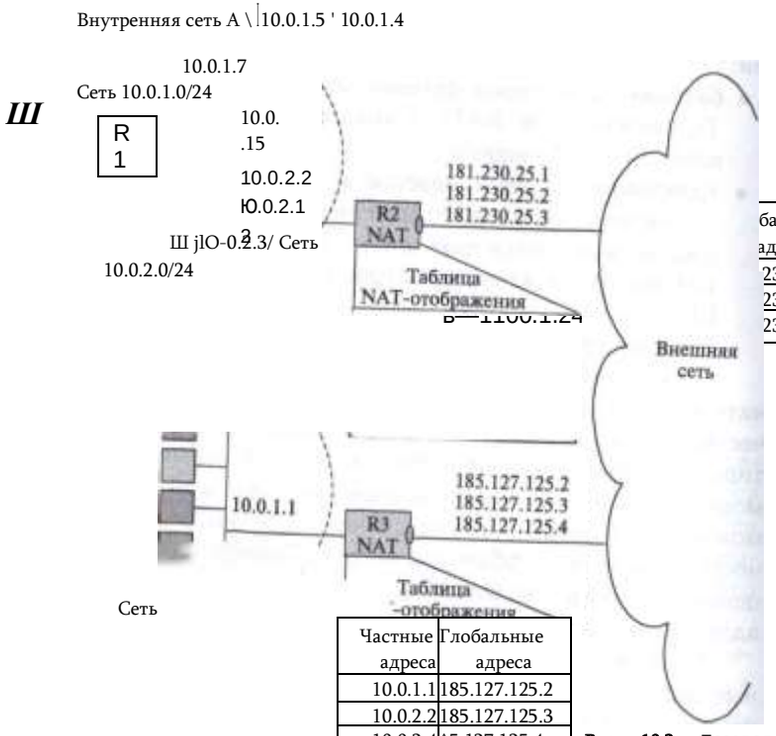
Базовая трансляция сетевых адресов

Если количество локальных узлов, которым необходимо обеспечить выход во внешнюю сеть, меньше или равно имеющегося количества глобальных адресов, то для каждого частного адреса гарантировано однозначное отображение на глобальный адрес. В каждый момент времени количество внутренних узлов, которые получают возможность взаимодействовать с внешней сетью, ограничивается количеством адресов в глобальном наборе. Понятно, что в такой ситуации целью трансляции является не столько решение проблемы дефицита адресов, сколько обеспечение безопасности.

Частные адреса некоторых узлов могут отображаться на глобальные адреса *статически.* К таким узлам можно обращаться извне, используя закрепленные за ними глобальные адреса. Соответствие внутренних адресов внешним задается таблицей, поддерживаемой маршрутизатором или другим устройством (например, брандмауэром), на котором установлено программное обеспечение NAT

В нескольких тупиковых доменах могут быть совпадающие частные адреса. Например, в сетях А и В на рис. 10.2 для внутренней адресации применяется один и тот же блок адресов 10.0.1.0/24. В то же время адреса внешних интерфейсов обеих сетей (181.230.25.1/24, 181.230.25.2/24 и 181.230.25.3/24 в сети А и 185.127.125.2/24, 185.127.125.3/24, 185.127.125.4/24 в сети В) уникальны глобально, т. е. никакие другие узлы в составной сети их не используют. В данном примере в каждой из сетей только три узла имеют возможность «выхода» за пределы сети своего предприятия. Статическое соответствие частных адресов этих узлов глобальным адресам задано в таблицах пограничных устройств обеих сетей.

Когда узел 10.0.1.4 сети А посылает пакет хосту 10.0.1.2 сети В, то он помещает в заголовок пакета в качестве адреса назначения глобальный адрес 185.127.125.3/24. Узел-источник направляет пакет своему маршрутизатору R1 по умолчанию, которому известен маршрут к сети 185.127.125.0/24. Маршрутизатор передает пакет на пограничный маршрутизатор R2, которому также известен маршрут



Частные адреса	Глобальные адреса
10.0.1.1	185.127.125.2
10.0.2.2	185.127.125.3
10.0.2.4	^5.127.125.4

Рис. 10.2. Базовая трансляция сетевых адресов для исходящих сеансов

к сети 185.127.125.0/24. Перед отправкой пакета модуль NAT, работающий на данном пограничном маршрутизаторе, используя таблицу отображения, заменяет в поле адреса источника частный адрес 10.0.1.4 соответствующим ему глобальным адресом 181.230.25.1/24. Когда пакет после путешествия по внешней сети поступает на внешний интерфейс NAT-устройства сети B, глобальный адрес назначения 185.127.125.3/24 преобразуется в частный адрес 10.0.1.2. Пакеты, передаваемые в обратном направлении, проходят аналогичную процедуру трансляции адресов.

Заметим, что в описанной операции не требуется участия узлов отправителя и получателя, т. е. она прозрачна для пользователей.

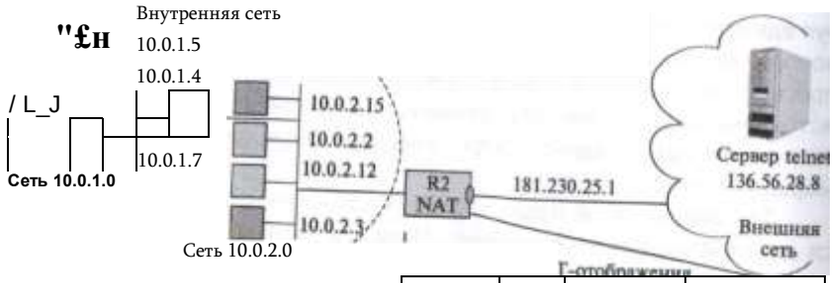
Трансляция сетевых адресов и портов. Пусть некоторая организация имеет частную IP-сеть и глобальную связь с поставщиком услуг Интернета. Внешнему интерфейсу пограничного маршрутизатора R2 назначен глобальный адрес, а остальным узлам сети организации назначены частные адреса. NAPT позволяет ***всем*** узлам внутренней сети одновременно взаимодействовать с внешними сетями, исполь-

»уя единственный зарегистрированный IP-адрес. Возникает законный вопрос, каким образом внешние пакеты, поступающие в ***ответ*** на запросы из частной сети, находят узел-отправитель, ведь в поле адреса источника всех пакетов, отправляющихся во внешнюю сеть, помещается один и тот же адрес — адрес внешнего интерфейса пограничного маршрутизатора?

Для однозначной идентификации узла отправителя привлекается дополнительная информация. Если в IP-пакете находятся данные протокола UDP или TCP, то в качестве такой информации выступают номер UDP- или TCP-порта соответственно. Но и это не вносит полной ясности, поскольку из внутренней сети может исходить несколько запросов с совпадающими номерами портов отправителя, а значит, опять возникает вопрос об однозначности отображения единственного глобального адреса на набор внутренних адресов. Решение состоит в том, что при прохождении пакета из внутренней во внешнюю сеть каждой паре {внутренний частный адрес; номер TCP- или UDP-порта отправителя} ставится в соответствие пара {глобальный IP-адрес внешнего интерфейса; назначенный номер TCP- или UDP- порта}. Назначенный номер порта выбирается произвольно, однако должно быть выполнено условие его уникальности в пределах всех узлов, получающих выход во внешнюю сеть. Соответствие фиксируется в таблице.

Эта модель при наличии единственного зарегистрированного IP- адреса, полученного от поставщика услуг, удовлетворяет требованиям по доступу к внешним сетям большинства сетей средних размеров.

На рис. 10.3 приведен пример, когда в тупиковой сети ***A*** используются внутренние адреса из блока 10.0.0.0. Внешнему интерфейсу маршрутизатора этой сети поставщиком услуг назначен адрес 181.230.25.1. Когда хост 10.0.1.4 внутренней сети посылает во внешнюю сеть пакет серверу telnet, то он в качестве адреса назначения использует его глобальный адрес 136.56.28.8. Пакет поступает маршрутизатору R1, который знает, что путь к сети 136.56.0.0/16 идет через пограничный маршрутизатор R2. Модуль NAPT маршрутизатора R2 транслирует адрес 10.0.1.4 и порт TCP 1245 источника в глобально уникальный адрес 181.230.25.1 и уникально назначенный TCP-порт, в приведенном примере — 3451. В таком виде пакет отправляется во внешнюю сеть и достигает сервера telnet. Когда получатель генерирует ответное сообщение, то он в качестве адреса назначения указывает единственный зарегистрированный глобальный адрес внутренней сети, являющийся адресом внешнего интерфейса NAPT-устройства. В поле номера порта получателя сервер помещает назначенный номер TCP-порта, взятый из поля порта отправителя пришедшего пакета. При поступлении ответного пакета на NAPT-устройство внутренней



Частные адреса	Порт	Глобальные адреса	Назначенный порт
10.0.1.4	1245	181.230.25.1	3451
10.0.2.15	4245	181.230.25.1	3452
10.0.2.3	1045	181.230.25.1	3455

UDP-сеансов

сети именно по номеру порта в таблице трансляции выбирается нужная строка. По ней определяется внутренний IP-адрес соответствующего узла и действительный номер порта. Эта процедура трансляции полностью прозрачна для конечных узлов.

Заметьте, что в таблице имеется еще одна запись с номером порта 1245. такая ситуация вполне возможна: операционные системы на разных компьютерах независимо присваивают номера портов клиентским программам. Именно для разрешения такой неоднозначности и привлекаются уникальные назначенные номера портов.

В технологии NAPT разрешаются только исходящие из частной сети TCP- и UDP-сеансы. Однако возникают ситуации, когда нужно обеспечить доступ к некоторому узлу внутренней сети извне. В простейшем случае, когда служба зарегистрирована, т. е. ей присвоен хорошо известный номер порта (например, WWW или DNS) и, кроме того, эта служба представлена во внутренней сети в единственном экземпляре, задача решается достаточно просто. Служба и узел, на котором она работает, однозначно определяются хорошо известным зарегистрированным номером порта службы.

Завершая рассмотрение технологии NAT, заметим, что помимо традиционной технологии NAT существуют и другие ее варианты, например технология двойной трансляции сетевых адресов, когда модифицируются оба адреса — и источника, и приемника (в отличие от традиционной технологии NAT, когда модифицируется только один адрес). Двойная трансляция сетевых адресов необходима, когда частные и внешние адресные пространства имеют коллизии. Наиболее часто это происходит, когда внутренний домен имеет некорректно назначенные публичные адреса, которые принадлежат другой организации. Подобная ситуация может возникнуть из-за того, что сеть орга

низации была изначально изолированной и адреса назначались произиольно, причем из глобального пространства. Или же такая коллизия может быть следствием смены поставщика услуг, причем организация хотела бы сохранить старые адреса для узлов внутренней сети.

Мониторинг сети

Файервол может защитить внутреннюю сеть от многих атак, не пропуская пакеты в тех случаях, когда его фильтры оказались правильно сконфигурированы. Однако фильтры файервола конфигурируются статически, и для их эффективной работы нужно предвидеть всевозможные атаки на сеть заранее, а это в принципе невозможно. Конфигурирование фильтров файервола происходит на основе *накопленного опыта* как администратора конкретной сети, так и коллективного опыта нескольких поколений администраторов Интернета, обобщенного в многочисленных RFC, статьях и дискуссиях на форумах. Поэтому *новыйтт* атаки имеет все шансы «просочиться» через файервол и достичь внутренних серверов защищаемой сети.

Для укрепления защиты сети с помощью файервола или файерволов необходимо дополнительно выполнять ***мониторинг трафика*** сети. Анализ трафика позволяет обнаружить следы атак, которые смогли преодолеть барьер файервола, а обнаружение атаки, как мы уже не раз подчеркивали, часто не менее важно, чем ее предотвращение. Обнаружение атаки позволяет оценить ее последствия, по возможности ликвидировать их, а также разработать меры по дальнейшему предотвращению подобных атак. Пропущенная атака может серьёзно подорвать авторитет компании в будущем, когда «тайное станет явным».

Мониторинг сети можно выполнять различными способами и с помощью средств разного типа. Сетевые сниферы позволяют захватывать трафик локальных сетей, представлять его в удобном для анализа виде, но сам анализ оставляют в основном на откуп администратору. Системы обнаружения вторжений (Intrusion Detection Systems, IDS) специализируются на автоматическом распознавании вторжений и угроз в прослушиваемом трафике локальных сетей. И, наконец, маршрутизаторы, поддерживающие протокол NetFlow, собирают обобщённые данные о трафике, передаваемом глобальными сетями, а программные системы анализа данных NetFlow автоматизируют поиск атак и угроз в этих данных.

Сетевые сниферы

Сетевые сниферы, которые также называются ***анализаторами протоколов,*** представляют собой программные или аппаратно-про

граммные системы, которые выполняют функции мониторинга и анализа трафика в сетях. Хороший анализатор протоколов может захватывать и декодировать пакеты большого количества протоколов, применяемых в сетях, — обычно несколько десятков.

Анализаторы протоколов позволяют установить некоторые логические условия для захвата отдельных пакетов и выполняют полное декодирование захваченных пакетов, т. е. показывают в удобной для специалиста форме вложенность пакетов протоколов разных уровней друг в друга с расшифровкой содержания отдельных полей каждого пакета.

Процесс анализа протоколов включает захват циркулирующих в сети пакетов, реализующих тот или иной сетевой протокол, и изучение содержимого этих пакетов. Основываясь на результатах анализа, можно осуществлять обоснованное и взвешенное изменение каких- либо компонент сети, оптимизацию ее производительности, поиск и устранение неполадок. Очевидно, что для того чтобы можно было сделать какие-либо выводы о влиянии некоторого изменения на сеть, необходимо выполнить анализ протоколов и до, и после внесения изменения.

Анализатор протоколов представляет собой либо самостоятельное специализированное устройство, либо персональный компьютер, обычно переносной, класса Notebook, оснащенный специальной сетевой картой и соответствующим программным обеспечением. Применяемые сетевая карта и программное обеспечение должны соответствовать топологии сети (кольцо, шина, звезда). Анализатор подключается к сети точно так же, как и обычный узел. Отличие состоит в том, что анализатор может принимать все пакеты данных, передаваемые по сети, в то время как обычная станция — только адресованные ей. Программное обеспечение анализатора состоит из ядра, поддерживающего работу сетевого адаптера и декодирующего получаемые данные, и дополнительного программного кода, зависящего от типа топологии исследуемой сети. Кроме того, поставляется ряд процедур декодирования, ориентированных на определенный протокол, например HTTP. В состав некоторых анализаторов может входить также экспертная система, которая может выдавать пользователю рекомендации о том, какие эксперименты следует проводить в данной ситуации, что могут означать те или иные результаты измерений, как устранить некоторые виды неисправности сети.

Несмотря на относительное многообразие анализаторов протоколов, представленных на рынке, можно назвать некоторые черты, в той или иной мере присущие всем им.

Пользовательский интерфейс. Большинство анализаторов имеют развитый дружественный интерфейс, базирующийся, как правило, на Windows или Linux. Этот интерфейс позволяет пользователю: выводить результаты анализа интенсивности трафика, получать мгновенную и усредненную статистическую оценку производительности сети, задавать определенные события и критические ситуации для отслеживания их возникновения, производить декодирование протоколов разного уровня и представлять в понятной форме содержимое пакетов.

Буфер захвата. Буферы различных анализаторов отличаются по объему. Буфер может располагаться на устанавливаемой сетевой карте, либо для него может быть отведено место в оперативной памяти одного из компьютеров сети. Если буфер расположен на сетевой карте, то управление им осуществляется аппаратно и за счет этого скорость ввода повышается. Однако это приводит к удорожанию анализатора. В случае недостаточной производительности процедуры захвата часть информации будет теряться и анализ будет невозможен. Размер буфера определяет возможности анализа по более или менее представительным выборкам захватываемых данных. Но каким бы большим ни был буфер захвата, рано или поздно он заполнится. В этом случае либо прекращается захват, либо заполнение начинается с начала буфера.

Фильтры. Фильтры позволяют управлять процессом захвата данных и тем самым позволяют экономить пространство буфера. В зависимости от значения определенных полей пакета, заданных в виде условия фильтрации, пакет либо игнорируется, либо записывается в буфер захвата. Использование фильтров значительно ускоряет и упрощает анализ, так как исключает просмотр ненужных в данный момент пакетов.

Переключатели — это задаваемые оператором некоторые условия начала и прекращения процесса захвата данных из сети. Такими условиями могут быть выполнение ручных команд запуска и остановки процесса захвата, время суток, продолжительность процесса захвата, появление определенных значений в кадрах данных. Переключатели могут использоваться совместно с фильтрами, позволяя более детально и тонко проводить анализ, а также продуктивнее использовать ограниченный объем буфера захвата.

Поиск. Некоторые анализаторы протоколов позволяют автоматизировать просмотр информации, находящейся в буфере, и находить в ней данные по заданным критериям. В то время, как фильтры проверяют входной поток на предмет соответствия условиям фильтрации, функции поиска применяются к уже накопленным в буфере данным.

Рассмотрим два популярных, свободно распространяемых программных сетевых снифера: Tcpdump и Wireshark.

Tcpdump — это утилита Unix, которая появилась в конце 80-х годов. Она была разработана сотрудниками Lawrence Berkeley Laboratory. Tcpdump поддерживает интерфейс командной строки в классическом Unix-стиле; результаты также выдаются на экран в виде текстовых строк, а также могут записываться в файл. Существует и версия WinDump, которая работает под Windows. Разработчики tcpdump создали открытую библиотеку программ ***pcap*** (packet capture), с помощью которой приложение, занимающееся мониторингом трафика сети, может получить доступ к сетевому адаптеру и выполнить захват кадров. Кроме того, определен формат файла с расширением '.pcap', в который помещаются захваченные кадры; этот формат сегодня поддерживают практически все программные и программно-аппаратные средства мониторинга трафика.

Tcpdump переводит сетевой адаптер компьютера в *режим неразборчивого захвата пакетов (promiscuous mode).* В стандартном режиме адаптер Ethernet буферизует и передает для дальнейшей обработки сетевым модулям операционной системы только кадры, у которых MAC-адрес назначения совпадает с MAC-адресом адаптера. В неразборчивом режиме адаптер захватывает все пакеты, биты которых появляются на его входе. В сетях Ethernet на разделяемой среде (которые практически уже не встречаются) это означает, что захватываются вообще все пакеты, циркулирующие в сети. В локальных коммутируемых сетях ситуация другая — здесь трафик сегментируется и на интерфейс компьютера попадают только те кадры, которые предназначены только ему. Поэтому для захвата трафика, направляемого к серверу или группе серверов, применяется *зеркализация портов коммутатора,* иллюстрируемая примером сети на рис. 10.4.

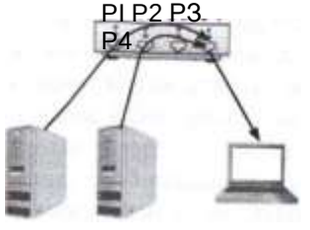


Рис. 10.4. Зеркализация трафика в коммутируемых локальных сетях

Здесь коммутатор сконфигурирован так, что весь трафик, поступающий и покидающий порты P1 и P2, зеркализуется на порт P4, к которому подключен компьютер с утилитой Tcpdump. В результате сниферы трафика Tcpdump получает возможность захватывать весь трафик серверов, подключенных к портам P1 и P2.

Результат захвата трафика утилитой tcpdump выглядит следующим образом:

```
11:24:25.061403 IP 193.62.83.186.66590 > ganymede.dev.ja.net.ssh:
Flags [.], ack 391232, win 8168, options [nop,nop,TS val 308850349
ecr 2275231285], length 0
11:24:25.061733 IP ganymede.dev.ja.net.ssh > 193.62.83.186.56590:
Flags [P.], seq 391232:391616, ack 337, win 194, options [nop,nop,TS
```



```
val 2275231286 ecr 308850349], length 384
11:24:25.062483 IP 193.62.83.186.56590 > ganymede.dev.ja.net.ssh:
Flags [.], ack 391616, win 8168, options [nop,nop,TS val 308850350
ecr 2275231286], length 0
11:24:25.062734 IP ganymede.dev.ja.net.ssh > 193.62.83.186.56590:
Flags [P.], seq 391616:392000, ack 337, win 194, options [nop,nop,TS
val 2275231287 ecr 308850350], length 384
```

Утилита tcpdump позволяет достаточно гибко задавать отображения захваченных пакетов (эти условия иногда называют *фильтрами отображения).* Например, для того чтобы на экране появились только начальный и конечный TCP-сегменты каждого TCP-соединения с хостом, не принадлежащим локальной сети, нужно вызвать утилиту со следующими условиями:

```
tcpdump 'tcp[tcpflags] & (tcp-syn | tcp-fin) != 0 and
not src and dst net localnet'
```

Снифер ***Wireshark*** (прежнее название — ***Ethereal)*** представляет собой популярную программу мониторинга трафика с развитым графическим интерфейсом. Существуют версии Wireshark для всех основных типов ОС: Unix/Linux, MS Windows, Mac OS.

Для удобства пользователя Wireshark представляет захваченные данные в виде иерархии полей пакетов (рис. 10.5).

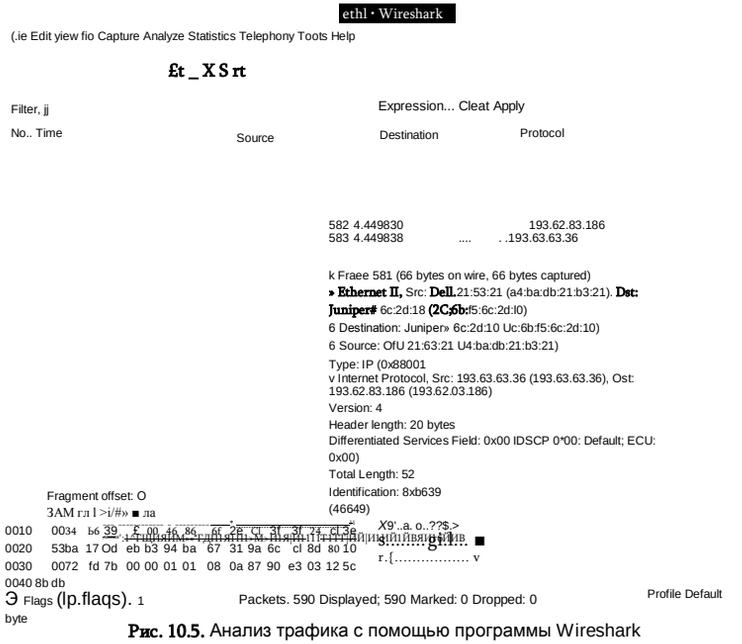


Рис. 10.5. Анализ трафика с помощью программы Wireshark

Верхняя панель окна результатов Wireshark показывает основные параметры каждого захваченного пакета укрупненно, примерно в том же стиле, что и tcpdump.

Нижняя панель позволяет рассмотреть один из пакетов более детально, при этом те поля, которые состоят из нескольких подполей, можно раскрывать рекурсивно, добираясь до самого дна иерархии признаков пакета, например до признаков заголовка TCP. Wireshark поддерживает весьма длинный список протоколов, начиная от канального (Ethernet) и кончая прикладным уровнем — на практике это означает возможность раскрытия заголовков протоколов с пояснениями назначения каждого поля.

Wireshark позволяет задавать фильтры двух типов: фильтр захвата пакетов и фильтр отображения пакетов; условия задания фильтров весьма гибкие, практически любое поле любого протокола может быть использовано в качестве условия этих фильтров.

Применение сетевых сниферов для обнаружения атак требует значительного опыта, так как они часто предоставляют слишком детальную картину, в которой среди десятков устанавливаемых соединений различных протоколов не так-то просто выделить подозрительную активность. Поэтому очень желательно автоматизировать эту деятельность, поручив анализ данных, собранных снифером в файле формата *pcap,* какой-нибудь имеющейся ***программе анализа трафика32.*** Такая программа работает с данными, собранными снифером, в режиме офф-лайн; распознование атак основано на выявлении типичных образцов атак, имеющихся в базе данных программы. Преимуществом сетевых сниферов по отношению к программам анализа трафика является то, что они дают полную картину проходящего через сеть трафика, так что у специалиста по безопасности всегда есть шанс разобраться в ситуации и обнаружить атаку даже в том случае, когда это новая атака и ее типичное поведение пока неизвестно имеющемуся программному обеспечению анализа трафика.

Система мониторинга NetFlow

Принципы работы и стандарты протокола NetFlow. ***NetFlow*** сегодня является основным средством учета и анализа трафика, проходящего через маршрутизаторы и коммутаторы сети. Поддерживающие протокол NetFlow сетевые узлы не только выполняют свою основную работу — передачу пакетов в соответствии с адресом назначения, но и собирают статистику о проходящих через них потоках данных и периодически отправляют их в *коллекторы* для хранения и обработки такой информации.

Не путать с анализаторами протоколов.

Практически все ведущие производители сетевого оборудования поддерживают протокол NetFlow, так что для того чтобы превратить маршрутизатор в источник информации о проходящем трафике, достаточно активизировать на нем NetFlow и указать, куда нужно передавать статистику.

32 Такой набор параметров потока определен в 5-й версии протокола NetFlow, разработанного компанией Cisco. Данная версия протокола на момент написания этого текста является наиболее распространенной реализацией NetFlow.

NetFlow собирает статистику не о каждом пакете, а о потоке пакетов, отсюда и название протокола (net — сеть, flow — поток). Под *потоком* понимается последовательность пакетов, принадлежащих одному и тому же соединению между определенными приложениями двух определенных компьютеров, например Skype-сессия между двумя пользователями, передача файла с сервера на клиентский компьютер, чтение данных веб-страницы с сервера браузером клиентского компьютера. Аналогом потока можно считать данные телефонного разговора между двумя абонентами, но так как компьютер, в отличие от телефона, может вести сразу несколько «разговоров» с различными собеседниками, то и потоков от него может исходить несколько. Поэтому для определения потока нужно использовать не только адреса участвующих в сессии компьютеров, но и дополнительные признаки. Собранную статистику о прохождении потоков можно использовать для различных целей и одна из важнейших — это распознавание сетевых атак.

Формально NetFlow определяет поток как набор нескольких признаков, в число которых наиболее часто* входят:

- IP адрес источника;
- IP адрес назначения;
- TCP/UDP порт источника;
- TCP/UDP порт назначения;
- тип протокола, переносимого пакетом IP (полезно в тех случаях, когда это не TCP или UDP, например это может быть ICMP или OSPF);
- индекс интерфейса, на которой получен пакет;
- качество обслуживания — значения байта ToS/DiffServ. Существуют другие версии NetFlow. которые включают больше

признаков, а также позволяют пользователям добавлять свои признаки для выделения потока из последовательности пакетов.

NetFlow собирает разнообразную статистику о потоке, такую как времена начала и окончания потока, объем данных, переданных с момента начала потока, ну и, естественно, все параметры, определяю

щие поток, т. е. адреса, порты и так далее. Кроме того, передается агрегированная информация о флагах заголовка TCP (SYN и другие).

Такая статистика передается в один или несколько коллекторов при окончании потока или же по истечении определенного периода времени, если поток еще не окончился. Для передачи статистики NetFlow чаще всего использует дейтаграммы UDP, для более надежной передачи применяется протокол *Stream Control Transmission Protocol* (SCTP).

Маршрутизатор может собирать данные NetFlow в двух режимах — непрерывном, когда обрабатывается каждый пакет, поступающий в маршрутизатор, и выборочном, когда обрабатывается только каждый п-й пакет. Выборочный режим менее надежен для распознавания атак, но зато он создает гораздо меньше дополнительной нагрузки на маршрутизатор, а для магистрального маршрутизатора, через который проходят десятки, а иногда и сотни тысяч потоков, это существенно.

Важно подчеркнуть, что NetFlow собирает только *метаданные о* трафике, не заглядывая в *поля данных* пакетов. Этим он отличается от анализаторов трафика и систем обнаружения атак. Если пользоваться аналогией с телефонными разговорами, то NetFlow может знать, кто разговаривал с кем и как долго, но не может знать, о чем они разговаривали (здесь аналогия немного хромает, так как порт TCP или UDP может сказать, какое приложение использовалось в соединении, так что предмет разговора немного прорисовывается). Часто статистику NetFlow сравнивают с телефонным счетом, который показывает, с кем и сколько разговаривал данный абонент, но не показывает о чем он говорил.

Долгое время NetFlow оставался фирменным протоколом компании Cisco, хотя другие производители также поддерживали его под собственными названиями, например JFlow (Juniper) или Cflowd (Alcatel-Lucent). Наиболее популярной версией является NetFlow v5, хотя существует и более поздняя версия NetFlow v9. Основным отличием версии 9 является большая гибкость при определении набора параметров собираемой статистики. В версии 5 перечень измеряемых параметров потока был ограничен семью параметрами, а в версии 9 администратор имеет возможность сделать выбор из более чем 50 параметров. В 2008 году протокол NetFlow был стандартизирован IETF и получил название IPFIX. IPFIX по своим свойствам очень близок к NetFlow версии 9 компании Cisco.

Выявление угроз и атак с помощью **NetFlow.** Данные NetFlow используются для оптимизации трафика, планирования сети и распознавания атак. *Распознавание сетевых атак* — это область применения NetFlow, наиболее интересующая читателей этой книги. Е1есмотря на то что NetFlow собирает только метаданные о потоках, такой информации часто бывает достаточно для того, чтобы распознать атаку.

Для этого применяется общий принцип мониторинга сети — сравнение ее текущего поведения с «нормальным», т. е. таким, которое устойчиво повторялось в прошлом и при этом мы знаем, что при этом атак в сети не наблюдалось. Другими словами, данные NetFlow используются для поиска аномалий в характере метаданных. Этот же прием используют службы безопасности банков — если вы обычно снимаете деньги в банкоматах Москвы, то снятие денег в Рейкьявике является для вас аномалией и ее нужно проверить, может быть вы просто полетели посмотреть на гейзеры и водопады, а может у вас украли данные вашей карточки.

Атака обычно генерирует не совсем обычный образец трафика, и для данных, собираемых NetFlow, существуют рекомендации для того, чтобы такие аномалии распознать.

Top **N & Baseline.** В соответствии с этим подходом внимание уделяется тем потокам, некоторые характеристики которых имеют «слишком большие значения» по сравнению с устоявшимся ***базовым уровнем*** *(baseline).* Этот метод анализа имеет две разновидности.

Top N Sessions. Это случай, когда один узел посылает в сеть (одному адресату или блоку адресатов) необычно большое число запросов на установление соединения. Такая активность характерна для DoS/DDoS-атак, узлов, зараженных червями, сканирования портов и некоторых других видов злоумышленной деятельности. Так, компьютер, зараженный червем, обычно пытается заразить таким кодом как можно больше других компьютеров и поэтому пытается с ними соединиться. Хост, рассылающий спам, будет пытаться отослать как можно больше писем и поэтому будет устанавливать большое количество соединений в единицу времени с портом 25 (SMTP порт, на который отправляется почта).

Top N Data. В этом случае хост, который обычно не входил в число 10 самых активных, начинает посылать или получать необычно высокое количество данных в единицу времени, т. е. генерировать высокий трафик. Это может быть DoS-атака или же активность червя, пытающегося заразить другие хосты.

Совпадение с известными образцами (Pattern Matching). Многие вредоносные программы связаны с определенными портами TCP и UDP, а также IP-адресами. Например, червь SQL Slammer использует порт 1434, а червь W32/Netsky.c worm всегда использует DNS-сервер с адресом из данного списка:

145.253.2.171, 151.189.13.35, 193.141.40.42, 193.189.244.205, 193.193.144.12, 193.193.158.10, 194.25.2.129, 194.25.2.129, 194.25.2.130, 194.25.2.131, 194.25.2.132, 194.25.2.133, 194.25.2.134, 195.185.185.195, 195.20.224.234, 212.185.252.136, 212.185.252.73, 212.185.253.70, 212.44.160.8, 212.7.128.162, 212.7.128.165, 213.191.74.19, 217.5.97.137, 62.155.255.16.

Поэтому обнаружение запроса DNS к одному из таких серверов должно служить индикатором возможной атаки (хотя и не со 100%-ной вероятностью), так что необходим дальнейший анализ поведения подозрительного узла.

Кроме того, полезно проверять, насколько текущее использование IP-адресов соответствует их «нормальному» использованию. Например, исходящий трафик для корпоративной сети должен содержать в качестве IP-адреса отправителя адреса из диапазонов, выделенных данному предприятию. И наоборот, входной трафик корпоративной сети не должен включать адреса из диапазонов данного предприятия. Подозрительным также будет появление частных IP-адресов в трафике из публичных сетей.

Анализ SYN и других флагов заголовка TCP. Атаки не всегда сопровождаются повышенной интенсивностью трафика, так что анализ списков TopN не всегда позволяет обнаружить атаку. Более детальное представление может дать анализ флагов заголовка TCP, в первую очередь флага SYN — мы уже рассмотрели детали этого анализа выше.

Анализ сообщений ICMP. Полезную информацию об атаках могут дать сообщения ICMP, такие как «Порт/хост/сеть недоступен». Большое количество такие сообщений может свидетельствовать о сканировании злоумышленником или вирусом хостов и портов, причем не только в случае использования TCP, но и UDP. Данные NetFlow не дают прямой информации о сообщениях ICMP в потоке пакетов, но такие данные легко извлечь из значений полей заголовка IP.

В целом, подход к анализу данных NetFlow должен быть адаптивным, основанным на постоянном обновлении и пополнении базы признаков атак, т. е. аналитик должен стараться «идти в ногу» с разработчиками вирусов, ботов и другого вредоносного программного обеспечения.

Программные системы анализа данных NetFlow. Когда система мониторинга получает каждую секунду данные о тысячах, а иногда и о десятках и даже сотнях тысяч потоков, то для обработки таких данных естественно использовать специальное программное обеспечение. Эти программные системы аналогичны программам анализа данных, собираемых сниферами.

Программные системы анализа данных NetFlow автоматизируют процедуры выявления аномальной активности в сети, проверяя потоки на соответствии многочисленным образцам разнообразных атак, в первую очередь таких атак, как атаки «Отказ в обслуживании» и сканирование сети и портов. Данные, отнесенные системой к подозрительной активности, выделяются в особую группу и предоставляются администратору сети в компактной форме. Кроме того, администратор может создавать собственные правила выявления подозрительной активности.

Система анализа данных NetFlow имеет панель, отражающую статистику потоков в сети в реальном времени, такая статистика может включать TopN-сессии, сессии, сгруппированные по отдельным протоколам (web, ssh, ftp, smtp и т. д.), по группам хостов и другим признакам.

Данные об отдельных подозрительных сессиях могут быть показаны администратору в более детальном виде для того, чтобы он мог провести более тонкий анализ и возможно выявить ранее не встречающийся тип атаки.

Типовые архитектуры сетей, защищаемых файерволами

Мы рассмотрели функциональные возможности файерволов по защите одной сети от возможных атак, исходящих от другой сети. В простейшем случае первая сеть — это единственная внутренняя сеть предприятия, а внешняя представлена всеми сетями Интернета, соединенными с внутренней сетью через единственную линию связи предприятия с провайдером Интернета. Чаще же ситуация оказывается сложнее — сеть предприятия может состоять из нескольких сетей, при этом серверы и хосты этих сетей нуждаются в защите различного типа. Р1апример, если в одной сети находится почтовый сервер и веб-сервер предприятия, а другой — сервер базы данных клиентов предприятия, то доступ к ним должен регулироваться в соответствии с разными правилами. Если добавить к этому, что многие предприятия соединяют свои сети с Интернетом несколькими линиями связи и, возможно, через несколько провайдеров, то защита сети предприятия приобретает еще одно измерение — защиту всего периметра сети с помощью нескольких файерволов, обеспечивающих дифференцированную и скоординированную защиту. Под периметром сети понимаеится граница между частной сетью, которой обладает и управляет некоторая организация, и публичной сетью, которая обычно представляет собой сеть или несколтько сетей провайдера Интернет.

Поэтому для надежной и эффективной защиты корпоративной сети необходимо ее логически сегментировать таким образом, чтобы ресурсы каждой подсети были подобными в отношении мер защиты.

Демилитаризованная зона

Ресурсы корпоративной сети, к которым обращаются внешние пользователи — почтовый сервер, веб-сервер, сервер DNS, — безусловно составляют в отношении мер безопасности отдельную группу.

Рис. 10.6. **Сетевой экран на базе двух маршрутизаторов**

Повсеместной практикой является размещение таких ресурсов в особой подсети, которая получила название ***(demilitarized zone, DMZ).***

Рассмотрим особенности организации защиты корпоративной сети на примере (рис. 10.6). В этой сети на рубеже защиты установлено два маршрутизатора, между которыми располагается демилитаризованная зона. Маршрутизаторы здесь выполняют функции файерволов сетевого уровня.

В сети DMZ расположены два общедоступных сервера — внешний сервер DNS и внешний веб-сервер предприятия. В этой зоне могут быть размещены также прокси-серверы. Учитывая, что само назначение этих компьютеров предполагает практически никак не ограничиваемый доступ к ним внешних пользователей (а значит, и злоумышленников), их необходимо защищать особенно тщательно. Главными задачами при защите этих компьютеров, называемых иногда ***компьютерами-бастионами,*** является обеспечение целостности и доступности размещенных на них данных для пользователей внешней сети. Эту задачу решают «индивидуальные» средства защиты, устанавливаемые на компьютерах-бастионах, такие, например, как антивирусные программы или фильтры спама. Кроме того, каждый сервер, к которому разрешено обращение внешних пользователей, должен быть сконфигурирован на поддержку только минимально необходимой функциональности. Например, публичный DNS-сервер предприятия не должен быть открытым для любых типов DNS-запросов, так какой может стать инструментом DDoS-атаки (например, должны быть запрещены рекурсивные запросы для пользователей, не являющихся сотрудниками предприятия; об этом смотрите более подробно в разделе «DNS-атаки»),

Чтобы пояснить, каким образом сеть DMZ усиливает защиту внутренней сети, давайте посмотрим, что произойдет, если какой-либо

злоумышленник сможет «взломать» первый рубеж защиты — внешний маршрутизатор — и начнет прослушивать трафик подключенной к нему сети DMZ. Очевидно, что он получит доступ только к трафику общедоступных серверов, который не является секретным.

Внешний маршрутизатор призван фильтровать трафик с целью защиты сети DMZ и внутренней сети. Однако строгая фильтрация в этом случае оказывается невостребованной. Общедоступные серверы по своей сути предназначены для практически неограниченного доступа. Что касается защиты внутренней сети, правила фильтрации для доступа к ее узлам и сервисам являются одними и теми же для обоих маршрутизаторов, поэтому внешний маршрутизатор может просто положиться в этом деле на внутренний маршрутизатор.

Обычно внешний маршрутизатор находится в зоне ведения провайдера, и администраторы корпоративной сети ограничены в возможностях его оперативного реконфигурирования. Это является еще одной причиной, по которой функциональная нагрузка на внешний маршрутизатор обычно невелика.

Основная работа по обеспечению безопасности внутренней сети возлагается на ***внутренний маршрутизатор,*** который защищает ее как от внешней сети, так и от сети DMZ. Правила, определенные для узлов-бастионов демилитаризованной зоны по доступу к ресурсам внутренней сети, часто бывают более строгими, чем правила, регламентирующие доступ внешних пользователей к этим ресурсам. Это делается для того, чтобы в случае взлома какого-либо компьютера- бастиона уменьшить число узлов и сервисов, которые впоследствии могут быть атакованы с этого ставшего опасным компьютера. Именно поэтому внутренний маршрутизатор должен отбрасывать все пакеты, следующие во внутреннюю сеть из сети DMZ, исключая пакеты нескольких протоколов (например, HTTP, SMTP, DNS), абсолютно необходимых пользователям внутренней сети для обращения к внешним серверам, установленным в демилитаризованной зоне или же за пределами корпоративной сети.

Для исключения возможности обращения внешних пользователей к серверам внутренней сети предприятия можно разрешить пропуск только тех TCP-пакетов, которые относятся к TCP соединениям, установленным по инициативе внутренних пользователей. Например, для маршрутизаторов Cisco это можно сделать с помощью такой строки списка доступа:

```
access-list 200 permit tcp any 201.15.0.0 0.0.255.255
established,
```

где 201.15.0.0/16 — диапазон адресов внутренней сети, а список доступа применяется во входном направлении к интерфейсу внутреннего маршрутизатора, к которому подключена сеть DMZ.

Защиту внутренней сети можно усилить, если во внутренней сети имеются аналоги внешних серверов, т. е. в нашем примере это веб-сервер и сервер DNS. В такой конфигурации с серверами DMZ разрешается взаимодействовать только внутренним ***серверам,*** а внутренним ***пользователям*** напрямую разрешено работать только с внутренними серверами. Например, назначенным по умолчанию DNS- сервером для пользователей должен быть внутренний DNS-сервер и только он может изнутри обращаться к внешнему DNS-серверу в том случае, когда он не может разрешить запрос самостоятельно.

Защиту внутренних серверов можно усилить за счет использования частных IP-адресов во внутренней сети. В этом случае внутренний маршрутизатор должен поддерживать NAPT трансляцию частных адресов в публичный адрес на своем интерфейсе, связывающем его с сетью DMZ.

Обобщенная архитектура сети с защитой периметра и разделением внутренних зон

Демилитаризованная зона является практически обязательным элементом защищенной архитектуры любой корпоративной сети. Кроме того, типичным является разбиение внутренней сети на большее число сегментов, характеризуемых сходными требованиями к защите на основе фильтрации и анализа трафика. Рис. 10.7 иллюстрирует этот подход.

Здесь достаточно крупная корпоративная сеть разделена на б сегментов, каждый из которых представляет собой отдельную зону безопасности. Сети зон соединены друг с другом через ***корпоративный файервол,*** непосредственной связи между этими сетями нет — такая архитектура позволяет надежно реализовывать правила политики безопасности для каждой зоны. Корпоративный файервол выполнен в виде двух устройств, работающих в режиме горячего резервирования, когда каждое из устройств отрабатывает одни и те же правила фильтрации трафика, и в случае отказа одного из устройств работоспособное устройство может продолжать обслуживать трафик, проходивший через отказавшее устройство без разрыва имеющихся соединений.

Корпоративный файервол также контролирует Интернет-трафик предприятия, который проходит через две линии связи с различными Интернет-провайдерами — такое соединение достаточно типично для крупных корпоративных сетей, так как оно обеспечивает высокую надежность связи с Интернетом.

Посмотрим на состав и требования к защите каждой из зон.

Зона 1 представляет собой демилитаризованную зону предприятия с открытыми для публичного доступа серверами. Ее особенности

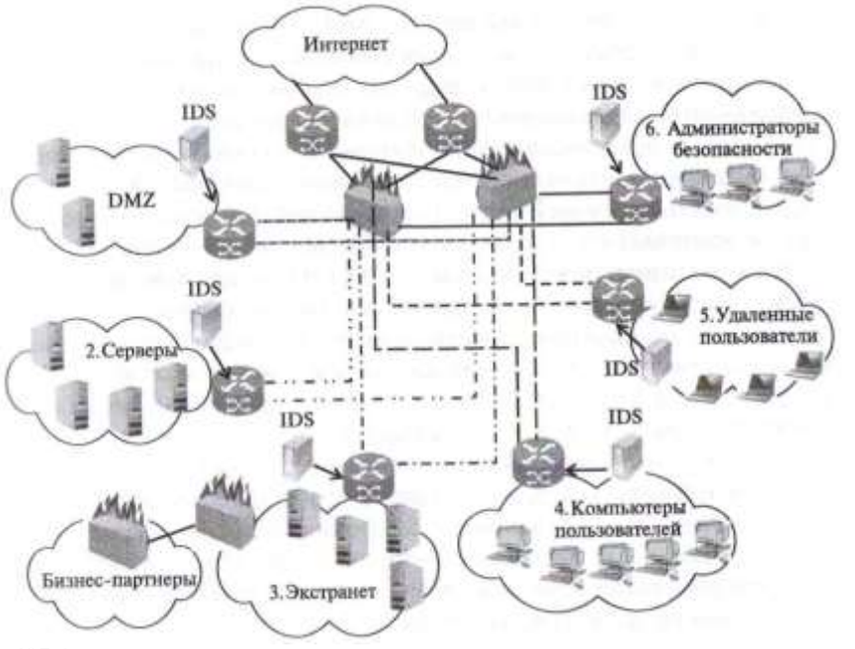


Рис. 10.7. Архитектура сети с несколькими зонами защиты

мы уже рассмотрели. Иногда в DMZ выделяют отдельную подзону, в которой размещают все внешние веб-серверы, так как специфика веб-сервисов весьма велика и их защита на прикладном уровне существенно отличается от защиты почтового сервера или DNS-сервера.

Зона 2 — это зона внутрикорпоративных серверов, иногда называемая ***корпоративным порталом.*** Здесь сосредоточены все информационные ресурсы предприятия, к которым обращаются сотрудники в свое работе, — внутренний веб-сервер, серверы баз данных, внутренний почтовый сервер, серверы приложений управления предприятием и т. п. К этой зоне должны иметь доступ только пользователи предприятия, доступ внешних пользователей должен быть заблокирован. Но и для внутренних пользователей доступ к ресурсам этой зоны должен быть ограничен файерволом в соответствии с ***принципом минимальных привилегий.*** На уровне файервола он означает, что пользователям должен быть разрешен доступ только к портам тех приложений, которые им нужны в работе, а все остальные порты должны быть файерволом заблокированы. Так если какому-то пользователю нужен доступ только к веб-сервису, то ему должен быть разрешен доступ к определенному серверу зоны 1 по порту 80 и/или 8080, а все остальные порты этого сервера должны быть заблокированы.

Зона 3— это зона ***экстранет*** *(extranet),* где сосредоточены ресурсы, доступ к которым должен предоставляться сотрудникам предпри- ятий-партнеров. Эти ресурсы представлены базами данных или вебприложениями, содержащими *конфиденциальные* данные, поэтому доступ к ним из публичного домена Интернета должен быть запрещен. Зона экстранет в нашем примере соединена с предприятиями-парт- нерами отдельной линией связи, возможно через отдельного провайдера, и контролируется отдельным файерволом. В других случаях доступ к экстранет может проходить через общие для всех зон линии связи с Интернетом и контролироваться тем же файерволом. Для обеспечения конфиденциальности данных сети экстранет здесь может быть применена технология ***виртуальных частных сетей (VPN),*** в этом случае файервол выполняет также функции VPN-шлюза (подробнее об этом — в следующем разделе).

Зона 4 **—** это внутренняя зона предприятия, в ней находятся клиентские компьютеры сотрудников предприятия. Для этой зоны разрешается установление соединений с серверами зоны 2 (корпоративный портал) и внешними серверами Интернета. Установление соединений с компьютерами этой зоны извне, т. е. из Интернета и из любой другой зоны предприятия, запрещен.

Зона 5 объединяет сотрудников предприятия, пользующихся удаленным доступом, т. е. работающих из дома или из сетей других предприятий и публичных провайдеров Интернета, например, из зон Wi-Fi вокзалов, аэропортов, кафе и т. п. Таких пользователей называют *мобильными пользователями.* Обычно для хостов этой зоны устанавливаются те же правила доступа, что и для пользователей зоны 4, т. е. они имеют доступ к корпоративному порталу и могут устанавливать соединения с ресурсами Интернета, которые проходят через тот же корпоративный файервол, что и соединения внутренних пользователей. Для обеспечения конфиденциальности, как и в случае экстранета, доступ мобильных пользователей осуществляется через защищенные каналы VPN.

Зона 6 объединяет серверы и клиентские компьютеры, используемые для администрирования средств безопасности предприятия. Здесь сосредоточены серверы политики файерволов, серверы антивирусной защиты, приложений обеспечения безопасности, таких, например, как анализаторы трафика NetFlow.

К сети каждой зоны подключен сервер системы IDS, который выполняет анализ трафика этой сети (или, по крайней мере, ее наиболее критичных сегментов) и предупреждает оператора систем безопасности о подозрительной активности.

Файерволы корпоративной сети должны быть сконфигурированы так, чтобы их правила отражали *политику безопасности предприя-*
тия. Собственно, эта политика и должна определять структуризацию ресурсов сети на зоны, приведенный пример — это только один из вариантов этой политики, хотя и достаточно типичный. Возможно и другое разбиение ресурсов на зоны — как более детальное, так и более укрупненное. Например, зона 5, объединяющая в нашем примере всех пользователей предприятия, работающих в локальной сети предприятия (т. е. не удаленно), может быть достаточно просто разбита на несколько зон — если принять во внимание

организационную структуру предприятия. В частности, финансовый отдел может быть выделен в отдельную зону безопасности в виду особой чувствительности информации, обрабатываемой сотрудниками этого отдела.

После определения количества зон для каждой из них должны быть определены критичные ресурсы, доступ к которым нужно ограничивать, и для каждого ресурса (а это, как правило, серверная часть некоторого сетевого сервиса) должны быть определены группы пользователей, которым обеспечивается доступ к ресурсу, остальным пользователям доступ должен быть заблокирован.

Средства, используемые для определения групп пользователей, зависят от типа файервола и его функциональных возможностей. Для файерволов на основе маршрутизаторов эти группы определяются диапазонами IP-адресов — по сути пользователи здесь отождествляются с компьютерами, которые они используют, более дифференцированной и надежной идентификации этот тип файервола не обеспечивает.

В то же время файерволы верхних классов, основанные на специальном программном и аппаратном обеспечении, могут использовать те же идентификаторы пользователей и групп пользователей, что и системы аутентификации и авторизации операционных систем или централизованных справочных служб типа Microsoft Active Directory. В этом случае правила файервола будут более избирательными и эффективными.

Еще раз подчеркнем, что представленная на рис. 10.8 архитектура не является единственным вариантом разбиения корпоративной сети на зоны безопасности и контроля взаимодействия зон между собой и с внешним миром. Она служит только для иллюстрации общего подхода к такому разбиению, а конкретные воплощения этого подхода могут быть самыми различными.

Вопросы к главе 10

1. Списки доступа маршрутизаторов выполняют функции:
а) файервола прикладного уровня с запоминанием состояния:
б) программного файервола:
в) файервола сетевого уровня без запоминания состояния:
г) прокси-сервера.

2. Какие признаки в пакете сможет учитывать файервол на основе маршрутизатора?

а) IP-адрес источника;

б) IP-адрес назначения;

в) MAC-адрес;

г) номер интерфейса, на который поступил пакет;

д) номер интерфейса, на который передается пакет;

е) номер порта TCP или UDP;

ж) тип протокола прикладного уровня;

з) тип команды протокола HTTP (GET, POST, PUT и т. д.).

3. Какое условие подразумевается неявным образом в конце каждого списка доступа маршрутизатора Cisco:

а) access-list nn deny any;

б) access-list nn permit any;

в) access-list ip deny any;

г) access-list any any.

4. Можно ли фильтровать трафик по адресу назначения в стандартных списках доступа маршрутизаторов Cisco?

а) да;
б) нет;

в) только в том случае, когда задан индивидуальный 1P-адрес.

5. Вставьте пропущенные параметры в строку расширенного списка доступа маршрутизатора Cisco, разрешающие прохождение эхо-запросов к хостам подсети 10.12.13.0/24 от любых хостов:

access-list 101 ... icmp any 10.12.13.0 ... eq ...

из перечисленных ниже. В ответе перечислите буквенные обозначения параметров в той последовательности, в которой они идут в строке списка доступа:

а) deny;
б) 255.0.0.0;
в) permit;
г) 10;
д) 8;
е) tcp;
ж) 0.0.0.0;
з) any;
и) 0.0.0.255;
к) udp.

6. Список доступа

ip as-path access-list 1 permit "117- выполняет

следующую фильтрацию:

а) разрешает объявления, которые содержат автономную систему 117 в любом месте списка AS;
б) запрещает объявления, которые содержат автономную систему 117 в любом месте списка AS;
в) разрешает объявления, которые содержат автономную систему 117 в начале списка AS;
г) разрешает объявления, список AS которых состоит из единственной автономной системы 117.

7. Провайдеры используют фильтрацию маршрутных объявлений BGP для:
а) уменьшения нагрузки на маршрутизаторы;
б) ускорения маршрутизации;
в) защиты от ошибок конфигурации маршрутизаторов.

8. Для того чтобы сервер мог быть защищен файерволом в сети с назначением IP адресов по протоколу DHCP, он должен:

а) не получать IP-адрес от сервера DHCP, а иметь статический IP-адрес, назначенный вручную администратором;
б) получать от сервера DHCP статический IP-адрес, связанный с его MAC-адресом;
в) запрашивать у сервера DHCP статический адрес;
г) запрашивать IP-адрес у файервола.

9. Технология NAT используется для:
а) назначения статических адресов внутренним хостам сети;
б) скрытия адресов хостов внутренней сети;
в) экономии адресов IPv4 в условиях их дефицита;
г) облегчает работу администратора по назначению адресов хостам сети.

10. Можно ли использовать традиционную технологию NAT для доступа из Интернета к внутреннему серверу, имеющему частный 1P-адрес?

а) нет;
б) да;
в) да, в том случае, когда сервер поддерживает сервис с известным номером порта и во внутренней сети имеется только один сервер этого сервиса.

11. Какой параметр пакета использует технология NAPT для различения внутренних хостов при использовании только одного публичного IP-адреса?

а) MAC-адрес хоста;
б) номер TCP- или UDP-порта;
в) доменное имя хоста.

12. Какие утверждения правильно описывают возможности сетевого анализатора:

а) сетевой адаптер анализатора протоколов может захватывать любые пакеты, а не только те, которые адресованы ему;

б) сетевой анализатор может захватывать пакеты, удовлетворяющие условиям фильтрации;
в) условия захвата пакетов могут включать переключатели, описывающие условия начала и окончания процесса захвата;
г) анализатор протоколов может декодировать захваченные пакеты и выводить их содержание в понятной для пользователя форме;

13. Сетевой анализатор подключен к порту 4 коммутатора Ethernet, имеющего всего 8 портов. Трафик каких портов сможет захватывать анализатор, если порт 5 коммутатора зеркализирован на порт б:

а) никаких, так как у коммутатора нет портов, зеркально отображенных на порт 4;
б) широковещательный трафик всех портов;
в) трафик порта 5.

14. Протокол NetFlow используется для:
а) сбора данных о потоках трафика для сглаживания пульсаций;
б) сбора данных о потоках трафика для оптимизации сети;
в) сбора данных о потоках трафика для планирования сети;
г) сбора данных о потоках трафика для распознавания атак.

15. Поток трафика характеризуется в версии NetFlow v5 следующими параметрами:
а) IP-адрес источника;
б) IP-адрес назначения;

в) MAC-адрес;
г) TCP/UDP-порт источника;
д) TCP/UDP-порт назначения;
е) индекс интерфейса, на которой пакет должен быть передан;
ж) тип протокола, переносимого пакетом IP;
з) индекс интерфейса, на которой получен пакет;
к) качество обслуживания — значения байта ToS/DiffServ.

16. На чем основан метод распознавания атак «Top N Sessions»?
а) в методе «Top N Sessions» выявляются потоки, передающие большое количество данных;
б) в методе «Top N Sessions» выявляются потоки, посылающие наибольшее число запросов на установление соединений;
в) в методе «Top N Sessions» выявляются потоки, передающие наибольшее количество данных по соединениям TCP.

17. Какие компьютеры вы бы включили в демилитаризованную зону корпоративной сети:
а) компьютеры удаленных пользователей предприятия;
б) все компьютеры предприятия;
в) серверы, к которым разрешен публичный доступ;
г) компьютеры администраторов безопасности предприятия;
д) внутренние серверы предприятия.

18. По каким признакам компьютеры должны объединяться в зоны, корпоративной сети, защищаемой файерволом:
а) компьютеры, которым нужен доступ к Интернет;
б) компьютеры, к которым осуществляется доступ из Интернет;
в) компьютеры, к которым предъявляются одинаковые требования политики безопасности предприятия.

11 Безопасность маршрутизации на основе BGP

Протокол ***BGP (Border Gateway Protocol)*** в версии 4 является сегодня основным протоколом обмена маршрутной информацией между автономными системами Интернета. Автономная система (***Autonomous System, AS) —*** это совокупность сетей под единым административным управлением, в которой существует единая политика маршрутизации. Каждый провайдер Интернета управляет собственной автономной системой или несколькими автономными системами; крупные корпоративные сети также часто представляют собой отдельную автономную систему, более мелкие являются частью автономной системы провайдера. Автономные системы имеют номера от 1 до 65 534, при этом номера от 65 512 до 65 534 являются частными, они зарезервированы для внутреннего использования провайдером и не должны использоваться для организации маршрутизации между сетями различных провайдеров Интернет.

Разбиение Интернета на автономные системы позволяет декомпозировать маршрутизацию, внутри каждой автономной системы маршрутизация выполняется любым известным способом — с помощью статических маршрутов, т. е. полностью вручную, или же на основе любого динамического протокола маршрутизации. Выбор внутреннего способа маршрутизации является делом организации, администрирующей данную автономную систему, главное — чтобы маршруты между узлами сетей автономной системы существовали и отвечали политике маршрутизации данной организации. Однако между автономными системами маршрутизация подчиняется общим правилам, которые реализованы в протоколе BGP.

Принципы работы протокола маршрутизации BGP

Поясним основные принципы работы BGP на примере (рис. 11.1). В каждой из трех автономных систем (AS 1021, AS 363 и AS 520) имеется несколько маршрутизаторов, исполняющих роль внешних шлюзов. На каждом из них работает протокол BGP, с помощью которого они общаются между собой.

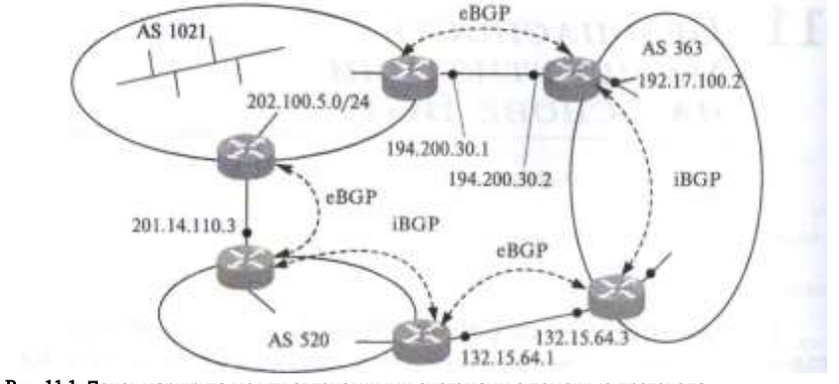


Рис. 11.1. Поиск маршрута между автономными системами с помощью протокола BGP

Маршрутизатор взаимодействует с другими маршрутизаторами по протоколу BGP только в том случае, если администратор ***явно*** указывает при конфигурировании, что эти маршрутизаторы являются его ***соседями.*** Например, маршрутизатор EG1 в рассматриваемом примере будет взаимодействовать по протоколу BGP с маршрутизатором EG2 не потому, что эти маршрутизаторы соединены двухточечным каналом, а потому, что при конфигурировании маршрутизатора EG1 в качестве соседа ему был указан маршрутизатор EG2 (с адресом 194.200.30.2). Аналогично, при конфигурировании маршрутизатора EG2 его соседом был назначен маршрутизатор EG1 (с адресом 194.200.30.1).

Такой способ взаимодействия удобен в ситуации, когда маршрутизаторы, обменивающиеся маршрутной информацией, принадлежат разным провайдерам. Администратор сети провайдера может решать, с какими автономными системами он будет обмениваться трафиком, а с какими нет, задавая список соседей для своих внешних шлюзов. Протоколы RIP и OSPF, разработанные для применения внутри автономной системы, обмениваются маршрутной информацией со всеми маршрутизаторами, находящимися в пределах их непосредственной досягаемости (по локальной сети или через двухточечный канал). Это означает, что информация обо всех сетях появляется в таблице маршрутизации каждого маршрутизатора, так что каждая сеть оказывается достижимой для каждой. В корпоративной сети это нормальная ситуация, а в ISP-сетях нет, поэтому протокол BGP исполняет здесь особую роль, поддерживая разнообразные и гибкие политики маршрутизации, говорящие о том, каким соседям передавать маршрутные объявления и о каких сетях им в этих объявлениях сообщать, а также от каких соседей и о каких сетях можно принимать маршрутные

объявления. Политика маршрутизации провайдера отражает условия по взаимной передаче трафика, имеющиеся в ***соглашениях об уровне обслуживания*** (***Service Level Agreement, SLA),*** которые провайдер заключает с другими провайдерами, т. е. так называемые ***пиринговые соглашения*** (от ***peering*** — отношения равных субъектов).

Для установления сеанса с указанными соседями BGP-маршру- гизаторы используют протокол TCP (порт 179). При установлении BGP-сеанса могут применяться разнообразные способы аутентификации маршрутизаторов, повышающие безопасность работы автономных систем.

Основным сообщением протокола BGP является сообщение UPDATE (обновить), с помощью которого маршрутизатор сообщает маршрутизатору соседней автономной системы о достижимости сетей, относящихся к его собственной автономной системе. Само название этого сообщения говорит о том, что это триггерное объявление, которое посылается соседу только тогда, когда в автономной системе что-нибудь резко меняется: появляются новые сети или новые пути к сетям или же, напротив, исчезают существовавшие сети или пути.

В одном сообщении UPDATE можно объявить об одном новом маршруте или аннулировать несколько переставших существовать маршрутов. Под маршрутом в BGP понимается последовательность автономных систем, которую нужно пройти на пути к указанной в адресе сети. Более формально информация о маршруте (BGP Route) к сети (Network/MaskJength) выглядит так:

BGP Route = AS_Path; NextHop; Network/Mask_length Здесь AS.Path — набор номеров автономных систем, NextHop — IP- адрес маршрутизатора, через который нужно передавать пакеты в сеть Network/MaskJength. Например, если маршрутизатор EG1 хочет объявить маршрутизатору EG2 о том, что в AS 1021 появилась новая сеть 202.100.5.0/24, то он формирует такое сообщение:

AS 1021; 194.200.30.1; 202.100.5.0/24 и передает его маршрутизатору EG2 автономной системы AS 363 (с которым у него, конечно, должен быть установлен BGP-сеанс).

Маршрутизатор EG2, получив сообщение UPDATE, запоминает в своей таблице маршрутизации информацию о сети 202.100.5.0/24 вместе с адресом следующего маршрутизатора 194.200.30.1 и отметкой о том, что эта информация была получена от протокола BGP. Маршрутизатор EG2 обменивается маршрутной информацией с внутренними шлюзами системы AS 363 по какому-либо протоколу группы IGP, например OSPF. Если у EG2 установлен режим перераспределения маршрутов BGP в маршруты OSPF, то все внутренние шлюзы AS 363 узнают о существовании сети 202.100.5.0/24 с помощью

объявления OSPF, которое будет иметь внешним. В качестве адреса следующего маршрутизатора маршрутизатор EG2 будет теперь объявлять адрес своего внутреннего интерфейса, например 192.17.100.2 (для IG1).

Однако для распространения сообщения о сети 202.100.5.0/24 в другие автономные системы, например в AS 520, протокол OSPF использоваться не может. Маршрутизатор EG3, связанный с маршрутизатором EG4 автономной системы 520, должен пользоваться протоколом BGP, генерируя сообщение UPDATE нужного формата. Для решения этой задачи он не может использовать информацию о сети 202.100.5.0/24, полученную от протокола OSPF через один из своих внутренних интерфейсов, так как она имеет другой формат и не содержит, например, сведений о номере автономной системы, в которой находится эта сеть.

Проблема решается за счет того, что маршрутизаторы EG2 и EG3 также устанавливают между собой BGP-сеанс, хотя они и принадлежат одной и той же автономной системе. Такая реализация протокола BGP называется внутренней ***(Interior BGP, iBGP),*** в отличие от основной, внешней ***(Exterior BGP***, ***eBGP***). В результате маршрутизатор EG3 получает нужную информацию от маршрутизатора EG2 и передает ее внешнему соседу — маршрутизатору EG4. При формировании нового сообщения UPDATE маршрутизатор EG3 трансформирует сообщение, полученное от маршрутизатора EG2 за счет того, что добавляет в список автономных систем собственную автономную систему AS 363, а полученный адрес следующего маршрутизатора заменяет адресом собственного интерфейса:

AS 363, AS 1021; 132.15.64.3; 202.100.5.0/24

Номера автономных систем позволяют исключать зацикливание сообщений UPDATE. Например, когда маршрутизатор EG5 передаст сообщение о сети 202.100.5.0/24 маршрутизатору EG6, то последний не будет его использовать, так как оно будет иметь вид

AS 520, AS 363, AS 1021; 201.14.110.3; 202.100.5.0/24

Так как в списке автономных систем уже есть номер собственной автономной системы, очевидно, что сообщение зациклилось.

Уязвимости и инциденты BGP

Маршрутизация между автономными системами на основе протокола BGP является (наряду со службой DNS) одним из наиболее уязвимых элементов Интернета. Это объясняется, во-первых, тяжелыми последствиями неверной работы BGP-маршрутизаторов провайдеров, когда маршруты ко многим частям Интернета вдруг исчезают или оказываются ложными для значительной части пользователей. Во- вторых, причиной повышенной уязвимости протокола BGP по сравнению с внутренними протоколами маршрутизации OSPF или IS-IS является то, что «собеседники» BGP-маршрутизатора находятся за пределами административной ответственности его организации и поэтому возможностей для проверки достоверности маршрутных объявлений BGP намного меньше, чем в случае внутренних протоколов, когда администратор сети всегда может

проверить конфигурацию любого маршрутизатора и понять причины некорректного или подозрительного поведения.

О значительной части крупных инцидентов, произошедших в Интернете по «вине» протокола BGP, трудно сказать, произошел ли этот инцидент из-за ошибки конфигурирования маршрутизатора персоналом провайдера или же это была спланированная и осуществленная атака. Провайдеры не любят раскрывать детали инцидентов по разным причинам, в том числе и заботясь о безопасности сети. Во многих статьях и документах, описывающих уязвимости BGP-маршрутизации, появляется новое действующее лицо — провайдер-злоумышленник (malicious ISP), который вольно или невольно создает проблемы для остальных провайдеров.

Первый широко известный и достаточно масштабный инцидент с BGP-маршрутизацией оказался весьма характерным, он очень ярко выявил уязвимости BGP, которые затем много раз проявили себя в аналогичных ситуациях. Этот инцидент произошел 25 апреля 1997 года, когда многие провайдеры обнаружили, что в их маршрутизаторах исчезли маршруты, описывающие путь к сетям Интернета. Стив Май- зел, администратор одного из крупных провайдеров, одним из первых обнаружил это неприятное обстоятельство и довольно быстро нашел, что причиной являются объявления AS ***7007,*** которые говорят, что путь ко многим сетям Интернета должен вести к их сети. Звонок провайдеру AS7007 помог устранить причину; оказалось, что виновником был единственный маршрутизатор одного из клиентов провайдера AS7007, который после реконфигурирования начал генерировать некорректные объявления не о своих сетях, причем с более специфическим адресом, чем адреса этих сетей в маршрутизаторах своего провайдера и большинства других провайдеров Интернета, в результате чего их записи были вытеснены из таблиц маршрутизации этой более специфической записью. После отключения виновного маршрутизатора таблицы маршрутизации провайдеров быстро восстановились. Шок от того, как просто оказалось вывести Интернет из строя, оказался таким, что Стив Майзел написал: «Завтра мы прочитаем об этом в каждом компьютерном журнале, газете и по услышим на телевидении как о конце Интернета».

Рис. 11.2. Эффект деагрегирования адресов

Детали этого и подобных инцидентов иллюстрирует рис. 11.2.

В этом примере диапазон адресов 12.24.0.0/16 принадлежит AS1. BGP-маршрутизатор этой AS объявляет о достижимости этих адресов через свою

сеть. Маршрутизаторы автономной системы AS5 принимают это объявление, помещают запись о достижимости адресов 12.24.0.0/16 в свои таблицы маршрутизации и передают его далее маршрутизаторам автономных систем AS4 и AS7. Теперь посмотрим, что произойдет, если маршрутизаторы автономной системы AS1 начнут по ошибке или умышленно распространять маршрутные объявления к адресам диапазона 12.24.128.0/17. Этот диапазон является поддиапазоном диапазона 12.24.0.0/16, но его маска и, соответственно, префикс, длиннее, а это и называется более специфическим адресом и ему маршрутизатор должен отдавать предпочтение перед более коротким адресом. Поэтому маршрутизаторы AS7 поместят запись о том, что трафик к хостам с адресами из диапазона 12.24.128.0/17 должен направляться в AS1, а не AS1. Эта информация распространится далее по всем AS Интернет, в том числе и по AS6, что приведет к потере связи с хостами 12.24.128.0/17, так как он находятся в AS6. Если AS1 будет распространять только объявление 12.24.128.0/17, то остальные хосты из диапазона 12.24.0.0/16 будут достижимы в AS6, но если AS1 будет распространять подобные объявления для всех поддиапазонов этого диапазона с маской 17, такие как 12.24.0.0/17, 12.24.192.0/17, 12.24.224.0/17 и так далее, то все хосты сети 12.24.0.0/16 станут недостижимы. Этот эффект называется деагрегированием адресов.

Инцидент с AS7007 был первой масштабной демонстрацией уязвимости маршрутизации на основе протокола BGP, который был разработан, как и другие протокола стека TCP/IP, в расчете на добрую волю всех пользователей Интернета и не имел никакой защиты от ошибок или злого умысла. В дальнейшем такие инциденты повторялись достаточно регулярно, один из них привлек большое внимание, потому что привел к временной недоступности такого популяр- кого сервиса, как Youtube. Это случилось в 2008 году, когда Pakistan lelecom пытался заблокировать доступ к Youtube для пользователей Пакистана (из-за отказа Youtube удалить некоторые материалы), а вместо этого допустил утечку специфических маршрутов к Youtube и Интернет, что привело к направлению трафика к Youtube со всего мира в сеть Pakistan Telecom в течение двух часов.

Подобные инциденты являются следствием того, что маршрутное объявление BGP формируется шаг за шагом многими провайдерами, при этом достоверность информации каждого шага проверить невозможно, так как у провайдера имеется полная свобода действий при обработке маршрутного объявления и передаче его соседним провайдерам. Например, вместо простого добавления номера своей AS к уже имеющейся последовательности номеров AS, он может выполнить такие манипуляции с полученным маршрутом:

1.Поместить адрес чужой сети с номером своей AS в качестве исходной автономной системы для того, чтобы направить трафик в свою AS. Такая атака называется «захват префикса», и именно она произошла в инцидентах с AS7007 и Pakistan Telecom. Для того чтобы его объявление выглядело предпочтительнее, адрес должен был более специфическим, чем у объявлений истинной исходной AS.

2.Выбросить из последовательности какую-то определенную AS. Это может быть сделано для того, чтобы обойти политику некоторой третьей AS,

которая по финансовым или иным соображениям блокирует все маршруты, которые проходят через удаленную AS.

3.Добавить номер соседней AS перед передачей его объявления. В этом случае соседняя AS, получив объявление и увидев в нем свой номер, отбросит его, так как решит, что объявление зациклилось.

4.Добавить номер своей AS несколько раз, чтобы объявление стало непривлекательным (из-за длины последовательности AS) для других провайдеров.

5.Составить ложную последовательность AS, но поместить в качестве исходного правильный (но не свой) номер AS, чтобы вызвать доверие к маршруту.

Как видим, незащищенность маршрутного объявления дает большой простор для злонамеренных искажений и просто ошибок при его обработке. Вероятность ошибки усугубляется тем, что в отличие от внутренних протоколов маршрутизации, которые обрабатывают сообщения с минимальным вмешательством администратора, работа протокола BGP обычно регулируется большим количеством правил фильтрации, которые задаются вручную администратором AS. Эти фильтры определяют политику маршрутизации данной AS, которая отражает взаимоотношения данного провайдера с каждым из провайдеров, с которым у него есть пиринговые соглашения о передаче тра-

фика. Вместе с тем, фильтры политики BGP представляют собой мощный и популярный способ защиты маршрутизации BGP от ошибок и атак. Но для этого их нужно правильно применять.

Защита BGP-сессии между соседними маршрутизаторами

Первым заслоном на пути ошибок и атак типа «злоумышленник посередине» должна быть защита BGP-сессии между соседними маршрутизаторами, особенно защита сессии eBGP, так как маршрутизаторы в этом случае принадлежат разным провайдерам и между ними могут находится промежуточные коммутаторы и другие, не BGP, маршрутизаторы.

По умолчанию сессия BGP использует протокол TCP, поэтому атаки на TCP, описанные выше, подвергают риску и работу BGP. Результатом атаки на TCP может стать удаление всех BGP-маршрутов из таблицы маршрутизации, так как все они могли быть получены в результате одной и той же длительной TCP-сессии между BGP- маршрутизаторами. Подделка TCP-сегмента может привести к появлению ложного маршрута в таблице маршрутизации или удалению корректного маршрута из нее.

Для защиты TCP-сессии между BGP-маршрутизаторами рекомендуется использовать режим работы TCP с аутентификацией сегментов, т е. по стандартам TCP MD5 или TCP AO. Название первого стандарта «Защита сессий BGP с помощью подписи TCP MD5» говорит о том, что причиной его появления было стремление обезопасить BGP-маршрутизаторы от TCP-атак, и это объяснимо, так как TCP-соединения между ними носят долговременный характер, т. е. обладают повышенной уязвимостью по сравнению, например, с кратковременными веб-сессиями. На самом деле стандарт TCP MD5 защищает соединения TCP вне зависимости от того, какой протокол его использует — BGP, HTTP или FTP, название просто отражает главную по мнению разработчиков область его применения. Так как TCP AO обеспечивает более надежную защиту BGP, то его применение сегодня более предпочтительно.

Защита BGP на основе данных региональных информационных центров Интернета

С середины 90-х годов региональные информационные центры Интернета, которые распределяют IP-адреса и номера AS среди провайдеров своих регионов (т. е. RIPE NCC, ARIN, APNIC, LACNIC и

AlriNIC), начали вести ***базу данных маршрутов Интернета*** (***Internet Houting Registry, IRR).*** Каждый провайдер, зарегистрированный в базе IKR. характеризуется номерами AS, которые он администрирует. Для мждой AS указываются условия политики маршрутизации, которую провайдер поддерживает по отношению ко всем соседним AS, с которыми у него установлены пиринговые отношения. Пусть, например, провайдер ISP1 администрирует AS1 и у него имеются пиринговые | оглашения с AS2, AS3 и AS4, тогда в базе IRR будет существовать объект типа aut-num с параметрами такого вида: aut-num: AS1

```
aut-name:   ISP1
import:     from AS2 action pref=50; accept AS2
export:     to AS2 announce AS1
import:     from AS3 action pref=50:        accept any
export:     to AS3 announce AS1
import:     from AS4 action pref=50:        accept AS3
export:     to AS4 announce AS1
address:      XX XXXXX XXXX
phone:      YY-YYYYY-YYYYY
```

Из данных объекта видно, что политика маршрутизации AS1 состоит в том, что ее маршрутизаторы объявляют каждой из пиринговых AS только о тех маршрутах, которые ведут к адресам ее собственных сетей (атрибут 'announce AS1' говорит об этом). В свою очередь AS1 также готова принимать от AS2 и AS4 только те маршруты, которые исходят от адресов их собственных сетей, но от AS3 она готова принимать любые маршруты. Скорее всего, AS2 и AS4 являются клиентами AS1, а AS3 — это магистральный провайдер, через которого происходит связь AS1 с остальными автономными системами Интернета. В параметрах атрибутов 'export' и 'import' можно использовать не только номера автономных систем, но и префиксы IP-адресов.

Кроме объекта 'aut-num' в базе IRR существует также объект 'route', который говорит о том, какие адреса будет объявлять данная автономная система в своих маршрутных объявлениях. Эти адреса должны быть выделены данной AS, но нужно отметить, что AS не обязана объявлять все выделенные ей адреса, некоторые могут быть еще не назначены реальным сетям, а некоторые могут быть предназначены для внутренней маршрутизации. Таким образом адреса, указанные в объектах 'route', являются подмножеством адресов, выделенных провайдеру.

Регистрация в базе IRR не является обязательной для провайдеров, но она желательна, а иногда и необходима, так как некоторые провайдеры отказываются устанавливать пиринговые отношения

с провайдерами, не зарегистрированными в базе IRR. Нужно отметить, что базы IRR всех пяти региональных центров RIR идентичны, центры поддерживают зеркализацию данных в базах независимо от того, в каком центре провайдер зарегистрирован.

База IRR является открытой, любой пользователь Интернет может запросить сведения о любой автономной системе с помощью команды *whols.* Обычной практикой провайдера является построение фильтров политики протокола BGP на своих маршрутизаторах на основании данных о политике соседей, полученных из базы IRR. Существует также утилита IRRToolSet, которая автоматизирует этот процесс и транслирует правила политики, описанные в базе IRR, в язык фильтров BGP определенного типа маршрутизаторов.

Очевидно, что администратор автономной системы AS7007 мог бы легко предотвратить утечку специфических префиксов из маршрутизатора своего клиента, если бы он создал простой фильтр, принимающий от маршрутизатора клиента только префиксы адресов, которые были назначены данному клиенту провайдером AS7007. Пострадавшие провайдеры соседних с AS7007 автономных систем также могли бы построить свои фильтры соответствующим образом, если бы провайдер AS7007 зарегистрировал свои объекты в базе IRR — однако он это не сделал.

Несмотря на то что база IRR существует уже много лет и большинство провайдеров регистрируют в ней свои правила политики маршрутизации, инциденты с захватом префиксов по-прежнему регулярно случаются. Причин этому несколько: во-первых, построить надежную систему фильтров, защищающую от всех возможных атак, сложно, а для больших транзитных автономных систем практически невозможно. Во-вторых, провайдеры, являющиеся клиентами какого-то определенного регионального Интернет-центра, не склонны доверять записям IRR, сделанным другим региональным центром; т. е. европейские провайдеры не вполне доверяют записям из баз ARIN или APNIC, так как эта информация никем не аутентифицирована и провайдерам остается только полагаться на хорошую работу региональных центров, а практику чужих центров они не знают так досконально, как своего собственного. Поэтому провайдеры используют в основном данные IRR для построения фильтров, защищающих от ошибок своих клиентов, а взаимоотношения с другими провайдерами фильтруются очень примитивно, например с помощью ограничения числа различных префиксов, которые принимаются от некоторого провайдера.

Поэтому уже достаточно давно стало очевидно, что проблему повышения безопасности BGP-маршрутизации нужно решать на принципиально новом уровне. Попыток создать безопасную версию протокола BGP было предпринято немало (например, предложения S-BGP,

••oBGP, psBGP), однако они не смогли завоевать расположение провайдеров, так как не обеспечивали хороший баланс между безопасностью и масштабируемостью.

Для решения этой задачи в 2006 году в IETF была создана рабочая группа SIDR (Secure Inter-Domain Routing), которая предложила новый подход, основанный на публичной инфраструктуре сертификатов ресурсов.

Сертификаты ресурсов и их использование для защиты BGP

Как мы видели из описаний атак и инцидентов, одной из причин неверной работы BGP является ошибочное или злонамеренное указание некорректного исходного номера AS для некоторого префикса сети, в результате чего маршрут к данному префиксу искажается. Поэтому в качестве первого шага защиты маршрутизации Интернет было решено создать систему, которая бы позволяла надежно проверять, имеет ли право автономная система, номер которой указан как исходный в объявлении BGP, объявлять данный префикс.

В качестве основы масштабируемой системы защиты исходного номера AS группа SIDR решила создать публичную систему сертификатов ***Resource Public Key Infrastructure, RPKI.*** Главным назначением системы RPKI является удостоверение владения некоторыми номерами AS и префиксами IP-адресов. Например, если провайдер ISP1 имеет сертификат RPKI, то этот сертификат показывает, что провайдеру в установленном порядке были выделены номера автономных систем AS1, AS2....ASn и префиксы IP1, IP2, IP3..... IPm. Установлен
ный порядок означает, что номера и адреса были выданы либо IANA (корневая организация, выделяющая номера и адреса в Интернете), либо пятью региональными Интернет-центрами, либо провайдерами, получившими их от региональных центров. Провайдеров в этой иерархии обычно называют ***локальными Интернет-центрами*** (***Local*** *Internet Register, UR).*

Система RPKI состоит, как и любая система PKI, из центров сертификации (Certificate Authority, CA), при этом каждая организация из иерархии IANA -» RIRs -> LIRs имеет свой CA, который выдает сертификаты по запросу нижестоящей организации. Сертификаты RPKI соответствуют стандарту X.509 с расширением RFC 3779, которое описывает дополнительные поля номеров автономных систем и префиксов адресов. В остальном это обычный сертификат, в котором содержится открытый ключ владельца сертификата и имя владельца. Сертификат называется сертификатом ресурса, потому что он не предназначен для аутентификации владельца сертификата, имя

владельца в сертификате может быть произвольным, оно используется для поиска в иерархической системе CA в процедуре проверки подлинности сертификата (для этого имя должно быть уникальным в этой системе). Сертификаты RPKI служат не для аутентификации владельца, а для авторизации — они свидетельствуют, что владелец имеет законное право распоряжаться номерами автономных систем и префиксов адресов, например передавать или продавать префиксы, указывать их в маршрутных объявлениях как исходные и тому подобное. Сертификаты здесь используются как масштабируемое решение, не требующее хранения множества паролей для проверки законности владения номером или номерами автономных систем и префиксов адресов некоторой организацией, вместо этого используется проверка предъявленного сертификата вдоль не очень длинной иерархии сертификационных центров.

Однако сам по себе сертификат RPKI не может свидетельствовать о достоверности номера исходной AS в объявлении маршрута BGP, так как обладатель некоторого префикса может делегировать право на объявление этого префикса как маршрута другой AS, отличной от той, которой он владеет. Это может быть вышестоящий провайдер или клиент провайдера; некоторые адреса провайдер может не объявлять по BGP вовсе, оставив их для внутреннего использования или же пока совсем не используя.

Поэтому для проверки законности объявления некоторой AS как исходной для определенного префикса в маршрутном объявлении рабочая группа SIDR предложила использовать новый тип объекта — ***Route Origination Authorisation, ROA,*** название которого можно перевести как объект ***авторизации источника маршрута.*** ROA содержит в качестве атрибута номер автономной системы и несколько атрибутов — префиксов IP-адресов, которые эта автономная система имеет право объявлять в маршрутах BGP. Объект ROA создается и подписывается владельцем префиксов, указанных в ROA. Для этого используется специальный одноразовый сертификат типа EE (End Entity — конечный субъект сертификации). Сертификат EE содержит публичный ключ и соответствующий закрытый ключ, с помощью которого владелец префикса подписал ROA. Объекты ROA и соответствующие им одноразовые сертификаты хранятся в распределенной базе данных RPKI, доступной провайдерам. Использование одноразового сертификата упрощает управление объектами ROA — легче организовать процедуру отзыва сертификата при изменении номера AS, связанного с какими-либо префиксами, не нужно хранить закрытые ключи, примененные для подписи ROA.

Провайдеры могут использовать базу данных объектов ROA двумя способами. В первом данные этих объектов используются для построения фильтров маршрутизаторов, так же, как и данные объектов ‘aut-num’ и 'route' из базы IRR. Отличие состоит в том, что объекты ROA обладают цифровой подписью, которую можно проверить, а объекты базы IRR — нет, поэтому доверия к объектам ROA больше, даже тогда, когда они созданы не «своим» региональным Интернет- центром. Во втором способе предлагается фильтры не строить, а автоматизировать процесс проверки достоверности источника маршрута. Для этого каждому провайдеру необходимо создать свой локальный

кэш базы RPKI, который маршрутизаторы могут опрашивать с помощью нового протокола RPKI-router. Этот протокол может работать поверх TCP, ssh или SSL, с помощью него маршрутизатор получает копии объектов ROA, которые он использует при проверке каждого маршрутного объявления BGP.

База данных RPKI и протокол RPKI-router являются новыми средствами защиты BGP-маршрутизации, но они уже реализованы на практике. Такие ведущие производители маршрутизаторов, как Cisco и Juniper, реализовали протокол RPKI-router в своих операционных системах IOS и Junos. Распределенную базу RPKI поддерживают все региональные Интернет-центры, но так как хранящиеся в ней объекты ROA покрывают не все префиксы Интернета, то результат проверки источника нового маршрута BGP-маршрутизатором может иметь три значения:

- «достоверный источник» — если для префикса из объявления имеется ROA с таким же префиксом и номера исходной AS в объявлении и ROA совпадают;
- «недостоверный источник» — если для префикса из объявления имеется ROA и его номер исходной AS не совпадает с номером AS объявления;
- «неизвестный источник» — если для префикса из объявления нет соответствующего ROA в базе RPKI.

Варианты действий маршрутизатора по результатам проверки маршрута могут быть разными и они определяются администратором при конфигурировании маршрутизатора. Например, маршруты с недостоверным источником могут отбрасываться, а маршрутам с достоверным источником может отдаваться предпочтение перед маршрутам с неизвестным источником (если возникает несколько маршрутов к одной и той же сети). Можно поступить и более осторожно, предписав маршрутизатору отбрасывать все маршруты с неизвестным источником, но это чревато потерей достижимости тех сетей, для которых не был созданы объекты ROA.

Понятно, что неопределенность в случае «неизвестного источника» оставляет возможность атаки, если маршрутизатор сконфигурирован так, что он принимает такие маршруты, но это неизбежное зло единственно возможного варианта процесса внедрения любого нового стандарта в масштабах Интернета — постепенного внедрения шаг за шагом.

Защита полного маршрута BGP с помощью сертификатов RPKI

Возможность проверки достоверности номера исходной автономной системы для префикса сети является важным шагом в повышении защищенности протокола BGP от ошибок и атак. Однако это только первый шаг в нужном направлении, так как он не исключает манипуляций с маршрутом при передаче от провайдера к провайдеру. Даже тот факт, что указанная в маршруте исходная AS имела право объявить маршрут к данному префиксу (факт, проверенный с помощью ROA), не гарантирует того, что маршрут был сгенерирован данной AS, — его вполне мог скомпоновать и провайдер-злоумышленник.

Поэтому следующим шагом должно быть появление средств, которые

позволят маршрутизаторам на лету проверять достоверность всех звеньев маршрута. Таким протоколом должен стать, по мнению специалистов из группы SIDR, протокол ***BGPSEC.*** Этот протокол должен заменить текущую версию BGP в маршрутизаторах провайдеров, и это гораздо более сложный шаг, чем предыдущий — и в силу самой сложности задачи проверки всех звеньев маршрута, а не только одного, и в силу того, что применение ROI не требует модификации BGP, а BGPSEC — требует. Эта работа еще далека от завершения, так что мы только коротко обрисуем основную идею этого протокола.

BGPSEC основан на цифровых подписях каждого провайдера, участвующего в пошаговом формировании маршрутного объявления BGP. Получив объявление, маршрутизатор провайдера проверяет цифровые подписи предыдущих провайдеров, указанных как номера AS в объявлении, а затем добавляет свою подпись, которая удостоверяет предыдущую версию объявления плюс свой номер AS и номер AS следующего шага. Добавление номера AS следующего шага исключает перехват и незаконную передачу маршрутного объявления не по назначению злоумышленником, т. е. атаку «злоумышленник посередине». Это также гарантирует, что объявление прошло тот же путь через последовательность автономных систем, который указан в самом объявлении. Для проверок цифровой подписи используются сертификаты типа EE, выпущенные в рамках системы RPKI.

Вопросы к главе 11

1. Какой формат маршрутного объявления протокола BGP является корректным:
а) (IP-адрес сети, расстояние).....(IP-адрес сети, расстояние);
б) (1P-адрес1, 1P-адрес2)..... (1P-адрес1, 1P-адрес2);
в) AS1, AS2.....ASN, IP-адрес сети.

2. Что из перечисленного является уязвимостью протокола BGP:

а) отсутствие возможности проверить право автономной системы объявлять маршрут к некоторой подсети;

б) возможность подделки последовательности AS в маршрутном объявлении;

в) возможность проведения атаки «человек посередине» на сессию BGP между соседними AS;

г) возможность зацикливания маршрута из-за невозможности проверить наличие петель в маршрутном объявлении.

3. Будет ли принято или отвергнуто маршрутное объявление «AS 13999, AS 688, AS 376, AS 10388, AS 542, 195.47.108.0/24», полученное BGP-маршрутизатором AS 376 от соседнего BGP-маршрутизатора AS 13999?

а) будет принято;

б) будет отвергнуто.

4. Причиной инцидента AS7007, когда из таблиц маршрутизации многих провайдеров исчезли записи, ведущие к крупным сетям Интернета, было:

а) генерация провайдером-злоумышленником ложного объявления;

б) ошибка администратора AS7007, который разослал маршрутное объявление, отзывающее маршруты к исчезнувшим сетям;

в) ошибка конфигурирования маршрутизатора клиента AS7007, объявившего, что путь к исчезнувшим сетям должен вести в AS7007.

5. Каким образом можно «подделать» маршрутное объявление BGP, которое вы передаете вашему соседу, если ваша AS является транзитной для этого маршрута, а вы хотите, чтобы сосед не использовал этот маршрут для передачи трафика:

а) добавить в объявление номер AS вашего соседа;

б) выбросить из последовательности определенную AS;

в) добавить номер своей AS несколько раз;

г) заменить адрес сети и номер исходной AS на свои.

6. Какие меры предпринимают провайдеры при фильтрации маршрутных объявлений BGP от своих соседей:

а) запрещают своим соседям-клиентам объявлять маршруты с исходным номером AS, не равным номеру их собственной AS;

б) запрещают своим соседям-клиентам объявлять маршруты с длиной маски, меньшей определенного порога (слишком специфические маршруты);

в) принимают от своих соседей по пирингу не более определенного количества префиксов;

г) запрещают соседям по пирингу объявлять маршруты с исходным номером AS, не равным номеру их собственной AS.

7. Какие типы объектов содержит база данных маршрутов Интернета IRR:
а) все маршруты Интернета к каждой из подсетей с префиксом 16 битов и короче;
б) объект aut-num, описывающий политику экспорта и импорта маршрутов провайдера;
в) объект route, описывающий префиксы, которые объявляет данный провайдер.

8. По каким причинам провайдеры используют базу данных маршрутов Интернета IRR не эффективно:

а) она не содержит нужной информации;

б) провайдеры не доверяют информации, полученной из третьих рук;

в) аутентичность и целостность информации базы IRR нельзя проверить.

9. Для чего используются сертификаты ресурсов RPKI:
а) для аутентификации его владельца-провайдера;

б) для авторизации права распоряжаться номерами автономных систем и префиксов IP-адресов владельцем сертификата;

в) для цифровой подписи маршрутных объявлений BGP.

10. Что удостоверяет объект ROA:

а) тот факт, что указанная в нем автономная система имеет право объявлять указанные в нем префиксы;

б) право его владельца распоряжаться указанными в нем префиксами;

в) тот факт, что указанные в нем префиксы находятся в указанной в нем автономной системе.

11. На чем основан протокол BGPSEC?

а) на шифровании маршрутного объявления BGP;

б) на цифровой подписи маршрутного объявления BGP;

в) на установлении сессии IPSec между соседними маршрутизаторами BGP.

L 2 *Виртуальные частные сети*

Определение виртуальной частной сети

Виртуальной частной сетью (***Virtual Private Network, VPN***) называется инфраструктура логических соединений в публичной сети, которая имитирует свойства частной сети, такие как безопасная передача данных, независимая адресация узлов и, возможно, гарантированная пропускная способность соединений.

Наиболее популярным типом VPN сегодня являются так называемые ***VPN доступа,*** которые шифруют трафик удаленных пользователей на пути между их домашним компьютером и корпоративной сетью. Существуют и другие типы VPN, которые могут и не шифровать трафик. VPN различаются прежде всего тем, какие свойства частной сети они имитируют и в какой степени.

Свойства частной сети, имитируемые VPN

В течение довольно длительного начального периода своего существования (до Интернет-революции, т. е. до начала 90-х годов) корпоративные компьютерные сети представляли собой ***частные сети.*** Это значит, что сеть предприятия была полностью или почти полностью изолирована от сетей других предприятий, при этом все локальные сети предприятия, расположенные в разных городах (эти сети часто называют сайтами, подчеркивая их территориальную рассредоточенность), соединялись физическими каналами только между собой. Такие физические каналы типа «точка-точка» либо принадлежали самому предприятию (довольно дорогой и поэтому редко встречавшийся вариант), либо брались в аренду у операторов связи и назывались ***выделенными*** или ***арендуемыми каналами.*** Первое название подчеркивает тот факт, что канал постоянно коммутируется так, что вся его фиксированная пропускная способность выделяется клиенту. В начальный период создания глобальных компьютерных сетей

Рис. 12.1. **Сервис выделенных каналов**

выделенные линии представляли собой постоянно скоммутированные аналоговые телефонные соединения.

По мере роста популярности компьютерных сетей услуги выделенных каналов стали более востребованными, и такой сервис стали предоставлять в более широком масштабе на основе новых технологий первичных сетей: PDH, SDH, OTN и DWDM.

На рис. 12.1 показан пример построения корпоративной сети клиента ***А*** с помощью сервиса выделенных каналов. Сети 2 и 3 этого клиента соединены двумя выделенными каналами с сетью 1 того же клиента, образуя корпоративную сеть со звездообразной топологией. Выделенные каналы проложены через сети операторов 1 и 2.

Хотя выделенные каналы чаще всего не принадлежат предприятию, которое владеет IP-сетью, построенной на основе этих каналов, тем не менее такую сеть называют частной, так как частной является вся инфраструктура сети с коммутацией пакетов — уровни Ethernet и IP с их оборудованием, коммутаторами, маршрутизаторами и хостами. Принадлежность каналов передачи данных в данном случае не влияет существенно на свойства IP/Ethernet-сети, так как эти каналы обладают фиксированной пропускной способностью и достаточно хорошей защищенностью от воздействий других клиентов оператора связи, который предоставляет услугу выделенных каналов, — просто в силу принципа своей работы такие каналы трудно атаковать.

Главным отличием частной сети от общедоступной сети или сети, совместно используемой несколькими предприятиями, является ее *изолированность.* Перечислим, в чем выражается эта изолированность.

Независимый выбор сетевых технологий. Выбор ограничивается только возможностями производителей оборудования. То есть это может быть сеть IP поверх каналов OTN, проложенных в спектральных каналах DWDM, а может быть сеть I P/Carrier Ethernet, работающая на основе каналов SDH. Единственное, что здесь фиксировано, это стек IP на верхних уровнях транспортной инфраструктуры сети.

Независимая система адресации. В частных сетях нет ограничений на выбор адресов — они могут быть любыми, так как сеть не связана с другими публичными сетями, т. е. не входит в Интернет. Это могут быть частные или публичные IP-адреса, а также Ethernet-адреса.

Предсказуемая производительность. Собственные линии связи гарантируют заранее известную пропускную способность между сайтами предприятия (для глобальных соединений) или коммуникационными устройствами (для локальных соединений).

Максимально возможная безопасность. Отсутствие связей с внешним миром ограждает сеть от атак извне и существенно снижает вероятность «прослушивания» трафика по пути следования. Частные сети обеспечивают безопасность данных в соответствии с тремя главными ее критериями:

- аутентичность данных следует из того, что выделенная линия связи соединяет сеть-получатель только с определенной сетью-отправителем, третья сторона с высокой степенью вероятности не может подключиться к этой линии связи с топологией «точка- точка» и послать по ней данные;
- целостность данных определяется высоким качеством и надежностью выделенной линии связи, а также ее защищенностью от воздействия третьей стороны, так как только провайдер или провайдеры имеют к ней доступ (помимо пользователей);
- конфиденциальности данных определяется невозможностью доступа к выделенной связи третьей стороны (и обязательствами провайдера по обеспечению конфиденциальности данных своих клиентов).

Типы VPN

Обобщенная структура VPN показана на рис. 12.2. Как видно из этого рисунка, она очень близка к структуре частной корпоративной сети за исключением типа каналов — они являются логическими (название «виртуальные» также часто используется в этом контексте) каналами в сетях в коммутацией пакетов, пунктир выбран для их изображения на рисунке для подчеркивания этой виртуальности.

Классификация VPN может быть проведена по различным признакам.

По типу используемых логических каналов:

- VPN на основе защищенных каналов с шифрацией и аутентификацией (например, каналов IPSec);
- VPN на основе логических (виртуальных) каналов транспортных технологий с установлением соединений (например, технологии MPLS).

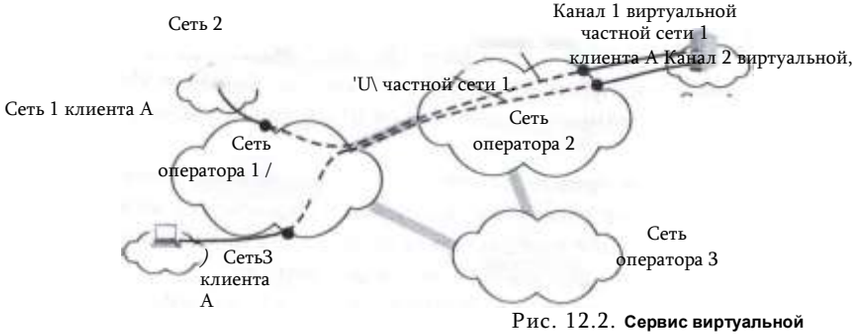


Рис. 12.2. **Сервис виртуальной частной сети**

В зависимости от того, кто реализует сети VPN, они подразделяются на два вида:

- *поддерживаемая клиентом виртуальная частная сеть* (Customer Provided Virtual Private Network, ***CPVPN***) отражает тот факт, что все тяготы по поддержки сети VPN ложатся на плечи потребителя. Поставщик предоставляет только «простые» традиционные услуги Интернет-доступа, а специалисты предприятия самостоятельно конфигурируют средства VPN и управляют ими. Большинство сетей VPN предприятий относится сегодня к этому типу;
- в случае *поддерживаемой провайдером виртуальной частной сети* (Provider Provisioned Virtual Private Network, ***PPVPN***) провайдер на основе собственной сети строит частную сеть для каждого своего клиента, изолируя и защищая ее от остальных. Такой способ организации VPN сравнительно нов и не столь широко распространен, как первый. Пример такого типа VPN — MPLS VPN.

В последнее время популярность сетей PPVPN растет — заботы по созданию и управлению VPN довольно обременительны и специфичны, поэтому многие предприятия предпочитают переложить их на плечи надежного поставщика. Реализация услуг VPN позволяет поставщику оказывать и ряд дополнительных услуг, включая контроль за работой клиентской сети, веб-хостинг и хостинг почтовых служб, хостинг специализированных приложений клиентов.

В зависимости от места расположения устройств, выполняющих функции VPN, виртуальная частная сеть может строиться:

- *на базе оборудования, установленного на территории потребителя* (Customer Premises Equipment based VPN, CPE-based VPN, или Customer Edge based VPN, CE-based VPN);
- *на базе собственной инфраструктуры провайдера* (Network-based VPN, или Provider Edge based VPN, PE-based VPN).

В любом случае основную часть функций (или даже все) по поддержанию VPN выполняют пограничные устройства сети — либо потребителя, либо поставщика.

Сети, поддерживаемые провайдером, могут строиться как на базе инфраструктуры поставщика, так и на базе оборудования, установленного на территории потребителя. Первый вариант наиболее понятен: провайдер управляет расположенным в его сети оборудованием. Во втором случае оборудование VPN расположено на территории клиента, но поставщик управляет им удаленно, что освобождает специалистов предприятия-клиента от достаточно сложных и специфических обязанностей.

Когда VPN поддерживается клиентом (CPVPN), оборудование всегда находится в его сети, т. е. VPN строится на базе устройств клиента (CE-based).

В зависимости от топологии связей VPN может быть:

- VPN топологией «звезда» (hub-and-spoke);
- VPN с топологией «каждый с каждым» (any to any);
- VPN с топологией «точка-точка» (point to point).

VPN с топологией «звезда» используется в основном для доступа сотрудников предприятия, работающих удаленно, к сети своего центрального офиса; поэтому VPN с такой топологией часто называется VPN доступа. Этот же тип топологии VPN может использоваться для связи сетей небольших офисов с сетью центрального офиса.

VPN с топологией «каждый с каждым» и «точка-точка» чаще всего используются для связи сетей офисов некоторого предприятия. В первом случае VPN обеспечивает взаимодействие в стиле локальной сети.

MPLS VPN

Этот тип VPN использует разграничение трафика клиентов сети MPLS, обеспечиваемое техникой виртуальных каналов. ***MPLS VPN*** относятся к типу поддерживаемых провайдером VPN и строится на основе собственной инфраструктуры провайдера (то есть относится к типам PPVPN и PE-Based VPN). MPLS VPN может иметь любую топологию соединений, в основном этот тип VPN используется для соединения сайтов предприятия. Соединяемые сайты должны подключаться к сети одного и того же провайдера, так как MPLS-сети различных провайдеров (в отличие от их IP-сетей), как правило, не взаимодействуют между собой (хотя теоретически такая возможность есть, на практике каждый провайдер использует свою сеть MPLS только для поддержки своих сервисов). MPLS — не единственная

технология, использующая технику виртуальных каналов, на ней также были основаны такие технологии, как X.25, Frame Relay и ATM, но сегодня они представляют только исторический интерес. MPLS вытеснила все остальные технологии виртуальных каналов из сектора сетей провайдеров связи, в основном благодаря своей тесной интеграции с протоколами стека TCP/IP, в то время как другие технологии этого типа были самостоятельными транспортными технологиями.

Виртуальный канал MPLS, называемый ***путем коммутации меток (Label Switched Path, LSP),*** устанавливается между двумя интерфейсами узлов сети оператором сети путем конфигурирования IP/MPLS-маршрутизаторов сети. Только после установления виртуального канала соединённые им узлы могут обмениваться данными.

Продвижение пакетов в сети MPLS происходит на основе ***меток (labels),*** которые имеют локальное значение для каждого MPLS-маршрутизатора. Значения метки менятся на каждом промежуточном маршрутизаторе вдоль пути LSP (говорят, что метка коммутируется, что отражено в названии технологии), правила этой замены отражены в таблице продвижения пакетов каждого MPLS-маршрутизатора. Конфигурирование таблиц продвижения пакетов вдоль пути LSP и является процедурой его установления.

Пограничный маршрутизатор MPLS принимает пакеты от клиентов и на основании адресной информации этих пакетов, например IP-адресов или MAC-адресов, посылает пакет вдоль пути LSP, давая метке ее начальное значение. Внутри сети MPLS ни IP-адреса, ни MAC-адреса не используются, продвижение пакетов выполняется только на основе меток, что делает сеть MPLS хорошо защищенной от атак из Интернета.

Таким образом, никакой узел на может посылать данные некоторому другому узлу сети, если между ними нет установленного оператором виртуального канала — пути LSP. Это свойство сетей MPLS, принципиально отличающееся от демократичных сетей IP с доступностью каждого узла, позволяет организовать изолированный набор клиентов сети, которые могут обмениваться данными только между собой, т. е. организовать VPN.

На рис. 12.3 показан пример сети IP/MPLS, предоставляющей услуги VPN для офисов двух предприятий. Сайты 1, 2 и 3 первого предприятия объединены в VPN1, а сайты 1, 2 и 3 второго предприятия объединены в VPN2.

Существует два типа MPLS VPN: ***MPLS VPN второго уровня (L2 MPLS VPN)*** и ***MPLS VPN третьего уровня (L3 MPLS VPN).*** Разница заключается в том, сети L3 MPLS VPN взаимодействует с сетями клиентов на основе IP-адресов, а L2 MPLS VPN — на основе адресной информации второго уровня, т. е. Ethernet MAC-адресов и тэгов

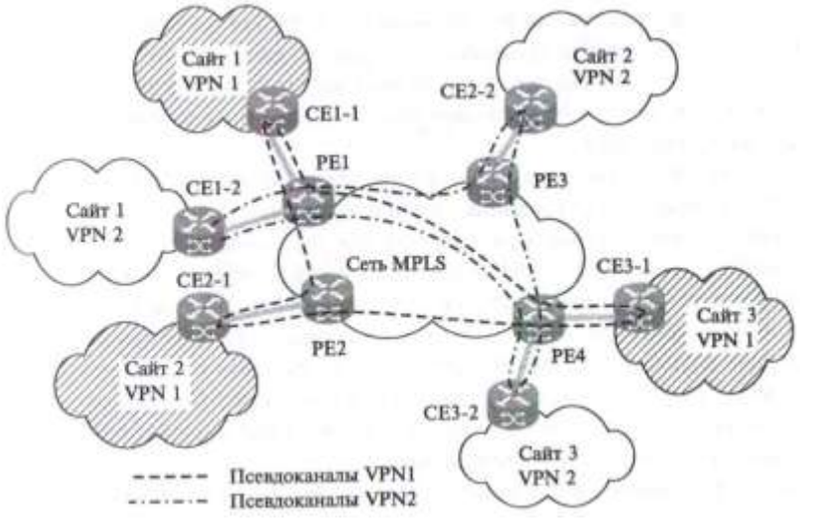


Рис. 12.3. MPLS VPN. Соединения MPLS VPN

VLAN. В том случае, когда MPLS VPN второго уровня предоставляют услуги в топологии «каждый с каждый», такие услуги называют также услугами VPLS (Virtual Private LAN Services), так как имитируют услуги локальной сети.

Как L2 MPLS VPN, так и L3 MPLS VPN обеспечивают защиту передаваемого трафика за счет разграничения его потоков на основе технологии коммутации меток. Шифрование трафика и проверка целостности данных в пакетах не входит в функции MPLS VPN, но эти функции могут выполняться другими технологиями поверх соединений MPLS VPN.

VPN на основе шифрования

Виртуальная частная сеть может быть определена как совокупность защищенных каналов, созданных предприятием в открытой публичной сети для объединения своих филиалов.

Основной публичной сетью является сегодня Интернет, и большинство типов ***защищённых каналов,*** стандартизованных сегодня (см. главу 7 «Защищенные каналы»), работают в Интернете «из конца в конец», используя стандартный IP-протокол. Защищенный канал может быть образован силами клиента Интернета, и от провайдеров обоих окончаний канала требуется только предоставление стандартного доступа в Интернет. В этом состоит основное преимущество VPN на основе шифрования от VPN на основе разделения трафика — первые

работают в пределах всего Интернета, в то время как вторые — в пределах сети одного провайдера, поддерживающего MPLS.

Сети VPN на основе шифрования могут быть организованы как силами клиентов, так и силами провайдеров, но последний вариант распространен мало.

Сеть VPN на основе шифрования представляет собой своего рода «сеть в сети», то есть сервис, создающий у пользователей иллюзию существования их частной сети внутри публичной сети. Одним из важнейших свойств такой «частной сети» является защищенность трафика от атак пользователей публичной сети. Сетям VPN доступна не только способность имитации частной сети; они дают пользователю возможность иметь собственное адресное пространство (например, частные IP-адреса, такие как адреса сети 10.0.0.0).

Технологии VPN на основе шифрования включают шифрование, аутентификацию и туннелирование:

- шифрование гарантирует конфиденциальность корпоративных данных при передаче через открытую сеть;
- аутентификация отвечает за то, чтобы взаимодействующие системы (пользователи) на обоих концах VPN были уверены в идентичности друг друга;
- туннелирование предоставляет возможность передавать зашифрованные пакеты по открытой публичной сети.

Для повышения уровня защищенности виртуальных частных сетей технологии VPN на основе шифрования можно применять ***совместно*** с технологиями VPN на основе разграничения трафика. Технологии VPN на основе разделения трафика иногда критикуют за недостаточный уровень безопасности, считая, что без шифрования трафика персонал поставщика услуг может получить несанкционированный доступ к данным. Действительно, такая вероятность существует, поэтому клиент услуг VPN на основе разграничения трафика, например MPLS VPN, может самостоятельно повысить защищенность своего трафика, прибегнув, скажем, к шифрованию передаваемых данных.

Сейчас наиболее широко используются сети VPN на основе протоколов IPSec и SSL.

Стандарты IPSec обеспечивают высокую степень гибкости, позволяя выбрать нужный режим защиты (с шифрованием или только с обеспечением аутентичности и целостности данных), а также использовать различные алгоритмы аутентификации и шифрования. Режим инкапсуляции IPSec позволяет изолировать адресные пространства получателя (клиента) и поставщика услуг за счет применения двух IP-адресов — внешнего и внутреннего.

Сети VPN на основе IPsec, как правило, строятся по типу CPVPN, т. е. как виртуальные частные сети, в которых клиент самостоятельно

Рис. 12.4. VPN доступа на основе шифрования

создает туннели IPSec через IP-сеть поставщика услуг. Конфигурирование сетей VPN на основе IPSec довольно трудоемко, поскольку туннели IPSec двухточечные, т. е. при полносвязной топологии их количество пропорционально $N(N-1)$, где N — число соединений. Необходимо учесть еще и непростую задачу поддержания инфраструктуры ключей. Протокол IPSec может применяться также для создания виртуальных частных сетей, поддерживаемых провайдером (PPVPN), — туннели в них также строятся на базе устройств клиента (CE-based), но эти устройства удаленно конфигурируются и администрируются поставщиком услуг.

На рис. 12.4 показан пример организации VPN на основе шифрования, которая обслуживает сотрудников предприятия, работающих удаленно. В корпоративной сети установлен УРИшлюз, который объединен с корпоративным файерволом (это необязательное объединение, но часто используемое, так как многие файерволы поддерживают функции УРЫшлюза). На компьютерах удаленных пользователей установлена программа-клиент VPN, которая обращается к шлюзу и устанавливает с ним защищенный канал. Шлюз VPN должен обладать высокой производительностью для того, чтобы поддерживать одновременно достаточное количество сессий с удаленными пользователями. Программное обеспечение шлюза должно также позволять администратору VPN управлять учетными записями удаленных пользователей, а также ключами, используемыми для аутентификации и шифрации. Учитывая высокий риск ошибки аутентификации удаленного пользователя, который в случае успешного ее исхода получает доступ к внутренним ресурсам предприятия, эта процедура должна быть максимально надежной, например с использованием токена доступа с двухфакторной проверкой например по паролю и псевдослучайному числу, генерируемому клиентом и шлюзом.

В VPN на основе шифрования возможно также использование защищенного канала с использованием протокола SSL. Напомним, что этот протокол работает на уровне представления, непосредственно под уровнем приложений, так что приложения должны явным способом его вызывать, чтобы создать защищенный канал для своего трафика.

Наиболее популярным приложением, использующим защищенные каналы SSL, является веб-браузер. В этом случае защищенные каналы SSL создаются для передачи данных по прикладному протоколу HTTP который в этом режиме работы часто называют протоколом HTTPS (что может создать ложное впечатление о появление нового единого протокола HTTPS, на самом деле это не так, просто этой аббревиатурой обозначают совместную работу двух протоколов — HTTP и SSL). Пользователи Интернета хорошо знают этот режим, так как браузер прибегает к нему во всех случаях, когда необходимо обеспечить конфиденциальность передаваемой информации: при покупках в Интернет-магазинах, при Интернет-банкинге и т. п.

Служба VPN на основе SSL функционирует на основе веб-портала, развернутого в локальной сети организации. Пользователи такой защищенной службы VPN получают удаленный доступ к ресурсам этой локальной сети, обращаясь к веб-порталу посредством обычного браузера через порт 443 (TCP-порт протокола HTTPS). Отсутствие специального клиентского программного обеспечения, требующего настройки, является значительным преимуществом VPN на основе SSL.

Вопросы к главе 12

1. Какие свойства частной сети имитирует виртуальная частная сеть?

а) безопасная передача данных;

б) независимая система адресации;

в) принадлежность одному предприятию;

г) предсказуемая производительность (опционально).

2. Может ли некоторая VPN быть одновременно классифицирована как VPN на основе виртуальных каналов, VPN, поддерживаемая провайдером, и VPN с топологией «звезда»:

а) да;

б) нет.

3. Может ли MPLS VPN быть классифицирована как VPN, поддерживаемая клиентом?

а) да;

б) нет.

4. Каким способом MPLS VPN обеспечивают безопасность передачи данных?

а) за счет цифровой подписи данных;

б) за счет шифрования данных;

в) за счет логического разграничения потоков данных.

5. Технология MPLS VPN поддерживает следующие топологии соединений пользователей:

а) каждый с каждым;
б) звезда;
в) точка-точка.

6. Технология L3 MPLS VPN называется технологией третьего уровня, потому что:

а) она реализуется маршрутизаторами, которые являются устройствами третьего уровня модели OSI;

б) она взаимодействует с клиентами на основе IP-адресов, которые относятся к третьему уровню модели OSI;

в) она реализуется тремя уровнями протоколов.

7. С какой целью VPN на основе шифрования используют туннелирование?

а) для обеспечения конфиденциальности данных;

б) для обеспечения целостности данных;

в) для передачи зашифрованных пакетов по публичной сети.

8. Что из перечисленного ниже не характеризует VPN на основе каналов SSL:

а) этот тип VPN использует веб-портал для доступа клиентов;

б) клиенты услуг сервиса VPN на основе каналов SSL используют веб-браузеры для доступа к ресурсам корпоративной сети;

в) клиент услуг сервиса VPN на основе каналов SSL должен использовать специальное клиентское программное обеспечение.

13 *Безопасность локальных беспроводных сетей*

Уязвимости локальных беспроводных сетей

Локальные беспроводные сети семейства 802.IIb/g/n, называемые также сетями Wi-Fi, используются сегодня повсеместно. На предприятиях они позволяют сотрудникам быть «подключенными» не только за своим рабочим столом, где они используют проводное соединение с корпоративной сетью, но и в комнатах совещаний и холлах. Аэропорты, вокзалы, кафе, рестораны и гостиницы оснащены точками доступа к беспроводным сетям, так что скучающий пассажир, посетитель или постоялец может прочитать последние новости на своих любимых Интернет-сайтах или пообщаться с друзьями. И, наконец, многие домашние сети построены на беспроводной связи, так как это гораздо удобнее и обеспечивает доступом к Интернету все компьютеры, планшеты и телефоны членов семьи.

Но за удобство приходится расплачиваться повышенным риском, который связан с локальными беспроводными сетями. Причина этого риска понятна — радиосигнал таких сетей распространяется на несколько десятков метров, а при использовании излучающих антенн- мачт — и на сотни метров, поэтому каждый, попавший в зону покрытия такой сети, может попытаться ей воспользоваться, даже не будучи авторизованным пользователем такой сети. Прослушивать сигналы беспроводной сети можно совершенно незаметно для окружающих — для этого не нужно подключать свой ноутбук или телефон к розетке проводной сети, достаточно иметь его в сумке включенным. Возможно также прослушивать сигналы беспроводной сети в скрытном режиме, когда прослушивающее устройство не посылает сигналов в сеть, а только получает их — этот режим не

является нормальным для стандартного программного обеспечения ноутбуков и других устройств, но существует большое количество программ, которые поддерживают такой «скрытый» режим работы. Это делает сетевую разведку особенно простой процедурой, при этом прослушивание может быть не только разведкой перед атакой, но и непосредственным актом кражи данных.

Самым известным и масштабным случаем несанкционированного сбора данных, передаваемых по беспроводным локальным сетям, является, наверно, сбор таких данных в ходе проекта Street View компании Google. В автомобилях, которые ездили по улицам для съемки изображений для Google Map, находились регистраторы активности беспроводных сетей, и в результате в распоряжении Google оказались имена частных сетей, MAC-адреса домашних компьютеров и их точек доступа, пароли, а в некоторых случаях — даже электронные письма.

Другим известным случаем утечки чувствительных данных через беспроводную сеть является атака на крупную торговую компанию TJX, имеющую около 3000 торговых центров и магазинов в разных странах мира. В 2007 году эта компания объявила об утечке финансовых и личных данных — злоумышленники получили доступ к отчетам обо всех торговых транзакциях компании за период с 2003 по 2006 годы, в результате чего были украдены номера от 45 миллионов до 200 миллионов кредитных карточек, а также 455 000 записей с именами и адресами покупателей. Атака началась со взлома беспроводной сети одного из магазинов, где такая сеть использовалась для связи торговых автоматов и кассовых аппаратов с серверами торговой системы. После получения доступа к сети магазина злоумышленники смогли использовать полученные данные для доступа к центральному серверу всей компании, создали там свою учетную запись и выкачивали данные из сервера предприятия через Интернет.

Несанкционированное использование беспроводных локальных сетей может преследовать и такую простую цель, как получение бесплатного доступа к Интернет. Зона покрытия такой сети может распространяться за пределы вашей квартиры или дома, и уровень вашего сигнала может оказаться вполне достаточным для устойчивого доступа к Интернет. Сама зона покрытия всегда имеет неправильную форму из-за наличия таких препятствий, как мебель, стены и т. п. Для точного контроля зоны покрытия необходимо проводить измерения уровня сигнала в различных точках помещения и за его пределами с помощью специальных устройств, так что заранее определить область, где ваш сигнал может «протечь», практически невозможно.

Две схемы организации беспроводной сети

В виду всего сказанного, локальные беспроводные сети нужно тщательно защищать, и методы такой защиты мы рассмотрим. Но перед этим давайте посмотрим на две основные схемы организации такой сети, изображенные на рис. 13.1. При одноранговой организации

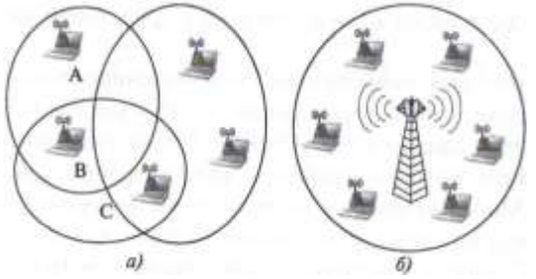


Рис. 13.1. **Беспроводные локальные сети семейства 802.11: *а* — одноранговая беспроводная сеть; 6 — беспроводная сеть с базовой станцией**

сети все компьютеры равноправны и могут взаимодействовать друг с другом (вариант а). В другом случае (вариант ***б)*** в сети имеется базовая станция, называемая также точкой доступа (access point). Все компьютеры в этом варианте взаимодействуют только с точкой доступа, а через нее — с другими сетями, так как обычно точка доступа имеет также проводное соединение и является также маршрутизатором (эти функции не являются обязательными для точки доступа, но для удобства организации домашних сетей они обычно добавляются).

Практически все локальные беспроводные сети предприятий, а также домашние сети используют точку доступа, которую необходимо максимально возможно защитить, включив режимы аутентификации пользователей и шифрации трафика. Тем не менее, и в такой защищенной сети может существовать лазейка для злоумышленника — ей может быть компьютер пользователя этой сети. Сегодня все операционные системы MS Windows, Apple Mac, различные версии Unix/Linux, позволяют пользователю превратить свой компьютер в точку доступа к беспроводной сети. Для этого нужно активизировать режим Internet Connection Sharing. И если такой режим сконфигурирован без использования аутентификации и шифрования, то компьютер пользователя становится открытой точкой входа в защищенную сеть. Если учесть, что беспроводной порт ноутбуков по умолчанию чаще всего включен, а пользователь ноутбука не всегда имеет хорошие знания о конфигурировании программного обеспечения, то вероятность существования таких лазеек на большом предприятии оказывается весьма высокой.

Методы защиты локальных беспроводных сетей

Протоколы шифрования трафика и аутентификации пользователей являются основными механизмами защиты локальных беспроводных сетей. К ним относятся протоколы WEP, WPA и WPA2, а также протокол 802.1x, который обеспечивает аутентификацию с помощью обращения к серверу RADIUS.

Протокол WEP

Этот протокол был разработан одновременно с первыми спецификациями локальных беспроводных сетей 802.11a и 802.11b. Разработчики WEP поставили перед собой цель — обеспечить такую безопасность передачи данных по беспроводной локальной сети, которая была бы эквивалентна безопасности передачи данных по проводной локальной сети, например Ethernet, без применения в последней каких-либо средств защиты. Эту цель отражает название протокола — ***Wired Equivalent Privacy —*** секретность, эквивалентная проводной.

Протокол WEP предоставляет возможность шифровать данные, передаваемые через беспроводную среду, и тем самым обеспечивает их конфиденциальность. Технология 802.11 предлагает еще один механизм безопасности — аутентификацию пользователя, подключающегося к сети. WEP использует шифрование на основе стандарта RC4 с использованием статического ключа 64 или 128 битов. Ключ длиной 64 бита состоит из 40-битового секретного слова пользователя и 24-битового вектора инициализации (вектор инициализации выбирается как псевдослучайное число для задания неповторяющегося результирующего ключа). Секретное слово обычно задается при конфигурировании точки доступа и клиентского компьютера с помощью 10 шестнадцатеричных цифр: 0-9 и A-F. При использовании 128- битового ключа пользовательское секретное слово составляет 104 бита (26 шестнадцатеричных цифр) при том же 24-битовом векторе инициализации.

Стандарт WEP предусматривает два режима аутентификации пользователя: режим открытой системы и режим разделяемого ключа. В обоих случаях при аутентификации пользователь должен указать идентификатор сети SS/D ***(Service Set IDentifier),*** называемый также именем беспроводной сети,. Точка доступа может широковещательно рассылать свой идентификатор SSID, и в этом случае программное обеспечение клиентского компьютера может показать пользователю список беспроводных сетей, находящихся в зоне приема компьютера. Точку доступа можно сконфигурировать и так, что она не будет широковещательно рассылать свой SSID — такой режим иногда рекомендуют как более защищенный, однако это не совсем так, потому что клиент может запросить у точки доступа ее SSID и она должна ответить.

В режиме открытой системы WEP фактически не выполняет аутентификацию и присоединяет любого клиента, правильно указавшего SSID.

В режиме разделяемого ключа выполняется 4-шаговая процедура аутентификации:

- клиент посылает запрос на аутентификацию (указывая SSID) точке доступа;
- точка доступа отвечает словом-вызовом;
- клиент шифрует слово-вызов с помощью разделяемого ключа и возвращает его точке доступа;
- точка доступа с помощью разделяемого ключа расшифровывает слово-вызов, полученное от клиента и сравнивает его с исходным, в случае совпадения аутентификация считается успешной.

Шифрование данных с помощью разделяемого ключа происходит независимо от того, какой режим аутентификации выбран.

Протокол WEP не обладает высокой степенью криптостойкости. В конце 90-х годов, когда локальные беспроводные сети получили свое распространение, на расшифровку WEP трафика типичным компьютером уходило около 24 часов; сегодня это можно сделать за минуты, если у вас в распоряжении имеется достаточное количество перехваченных WEP-пакетов и программа дешифрования, например aircrack-ng. Из-за этого WEP уже давно не рекомендуют применять для защиты локальных беспроводных сетей, а с 2008 года он запрещен для работы в системах обработки кредитных карт организацией Payment Card Industry (PCI). Возможно, это решение было ускорено описанным выше случаем с компанией TJX, которая как раз и использовала WEP во взломанной сети.

Для разработки более защищенного варианта беспроводных локальных сетей была создана рабочая группа 802.lli. В ожидании стандарта 802ю1И в 2002 году консорциум Wi-Fi Alliance разработал временную спецификацию TKIP (Temporal Key Integrity Protocol), которая была затем одобрена под названием WPA (Wi-Fi Protected Access — защищенный доступ к Wi-Fi). WPA, как и WEP, использовал RC4 в качестве алгоритма шифрования, но усиливал защиту сети за счет более сложного способа комбинирования вектора инициации с разделяемым секретом, заменяющего простую их конкатенацию в варианте WEP. В 2009 году организация IEEE рекомендовала более не использовать протокол WPA/TKIP из-за его недостаточной криптостойкости.

Окончательный вариант стандарта 802.lli, одобренный в 2004 году, получил неофициальное название WPA2.

Стандарт WPA2

Этот стандарт описывает надежное средство защиты беспроводных локальных сетей, сочетающее в себе гораздо более совершенные средства аутентификации пользователей и шифрования данных по сравнению со средствами стандартов WPA и WEP. В настоящее время поддержка

протокола WPA2 является необходимым условием сертификации оборудования консорциумом Wi-Fi Alliance.

Взаимная аутентификация точки доступа и клиента сети происходит в WPA2 на основе стандартов ***802.1x***/EAP. ***EAP (Extensible Authentication Protocol***) представляет собой расширяемый протокол взаимной аутентификации, который позволяет использовать различные методы аутентификации; он может быть инкапсулирован в различные транспортные протоколы. Спецификация IEEE 802.lx описывает работу EAP в локальных сетях (***EAPOL*** — EAP Over LAN) и определяет использование сервера RADIUS для аутентификации клиента на основе пароля или сертификата безопасности. Точка доступа в схеме аутентификации 802.1x является передаточным звеном между клиентом и сервером RADIUS, при этом обмен сообщениями между клиентом и точкой доступа, а также между точкой доступа и сервером RADIUS происходит с помощью сообщений EAPOL.

В случае успешной аутентификации сервером RADIUS точка доступа и клиент вырабатывают так называемый мастер-ключ (Master Key), а также набор временных ключей для шифрования и подписи пакетов различного типа (например, отдельные ключи используются для шифрования пакетов с уникальными и групповыми адресами). Мастер-ключ используется только для выработки временных ключей и никак не проявляет себя при обмене пользовательским данными между клиентом и точкой доступа.

Шифрование и целостность трафика в WPA2 обеспечивается с помощью протокола***CCMP*** (Counter Cipher Mode with Block Chaining Message Authentication Code Protocol), который основан на спецификации AES. Протокол CCMP использует для шифрования временный ключ, полученный на этапе аутентификации, он гораздо более криптостоек, чем RC4.

Беспроводные системы обнаружения вторжений

Использование протокола WPA2 для защиты локальной беспроводной сети предприятия сегодня является нормой. Кроме этого, рекомендуется применять специальные ***беспроводные системы обнаружения вторжений (Wireless Intrusion Detection Systems, WIDS).*** Такие системы сканируют спектр радиоволн, выделенный для локальных беспроводных сетей, и пытаются обнаружить нелегальные (т. е. не установленные администратором данной сети для общего доступа) точки доступа, а также некорректно сконфигурированные точки доступа. Кроме этого, система WIDS может распознавать некорректно сконфигурированные клиентские компьютеры (например, с активированным режимом Internet Connection Sharing), а также клиентские компьютеры злоумышленников, маскирующиеся под легальные с

помощью спуфинга MAC-адреса какого-либо легального пользователя сети. Для того чтобы обнаружить компьютер злоумышленника, WIDS создает базу данных «отпечатков» радиосигналов передатчиков компьютеров легальных пользователей, так как эти «отпечатки» носят индивидуальный характер. Сравнение сигнала присоединившего к сети компьютера с его сохраненным «отпечатком» позволяет во многих случаях уверенно распознать подделку MAC-адреса.

Кроме того, WIDS умеют распознавать основные типы атак, характерные для проводных сетей, например TCP- и UDP-flood, и многие другие.

Архитектура WIDS подобна архитектуре проводных IDS. Существуют как коммерческие WIDS, так как и WIDS с открытым кодом, например Kismet.

Вопросы к главе 13

1. Беспроводные локальные сети стандартов 802.11b/g/n более уязвимы, чем проводные локальные сети Ethernet, потому что:

а) их зона покрытия не имеет четких границ;

б) прослушивания трафика в этих сетях практически незаметно;

в) пакеты этих сетей не могут быть зашифрованы;

г) пароли пользователей передаются в этих сетях в открытом виде.

2. Вы защитили свою домашнюю беспроводную сеть, активировав строгую аутентификацию и шифрование данных по протоколу WPA2 на точке доступа. Какую уязвимость может использовать злоумышленник для проникновения в вашу сеть?

а) домашний компьютер, работающий под управлением Microsoft Windows;

б) домашний компьютер, у которого активирован режим Internet Connection Sharing без аутентификации пользователей и шифрования данных;

в) выход области покрытия сети за пределы вашего дома.

3. Верно ли утверждение «Запрет широковещательной рассылки SSID точкой доступа существенно повышает безопасность беспроводной локальной сети»?

а) да; б) нет.

4. Верно ли утверждение «Точка доступа, работающая по протоколу WEP, всегда получает пароль пользователя в открытом виде»?

а) да;

б) нет.

5. Что из перечисленного характеризует спецификацию 802.1x:

а) она использует протокол EAP;

б) она описывает применение протокола EAP в локальных сетях;

в) она основана на аутентификации с использованием сервера RADIUS;

г) она не использует набор временных ключей.

6. Для обнаружения компьютера злоумышленника система WIDS использует следующий прием:

а) она хранит базу данных «отпечатков» радиосигналов всех компьютеров легальных пользователей;

б) она выполняет запрос на логический вход во все компьютеры, работающие в сети, неудача такого входа выдает компьютер злоумышленника;

в) она посылает специальную аутентификационную последовательность радиосигналов, на которую компьютер злоумышленника не может правильно ответить.

Ф iMPERNA

Защита отатак, мошенничества и утечки данных в реальном времени.

Число атак на базы данных, таких как SQLinjection, продолжает расти, и базы данных, содержащие конфиденциальную информацию, являются приоритетной целью для хакеров и злонамеренных инсайдеров. Многие базы данных находятся в уязвимом положении и злоумышленники пользуются тем, что создание виртуальных «заплаток» баз данных занимает в среднем 6-9 месяцев. Последние события показывают, что инсайдеры могут представлять значительную угрозу для безопасности данных.

ImpervaSecureSphere Database Firewall (DBF)

эффективно защищает базы данных от атак, потери и утечки данных. Мониторинг, оповещение и блокировки, а также готовые политики безопасности и правила аудита SecureSphere защищают наиболее ценные ресурсы и обеспечивают целостность данных в режиме реального времени.

Об Imperva

Имперва - мировой лидер в области защиты данных, предлагает передовые решения для обеспечения безопасности бизнес-кри- тичных приложений и информационных активов как в физических, так и в виртуальных инфраструктурах. Тысячи заказчиков, в том числе ведущие корпорации, интернет-провайдеры и государственные структуры полагаются на решения Imperva для предотвращения утечек в своих информационных системах, реализации требований стандартов безопасности и управления рисками. Решения сертифицированы ФСТЭК России.

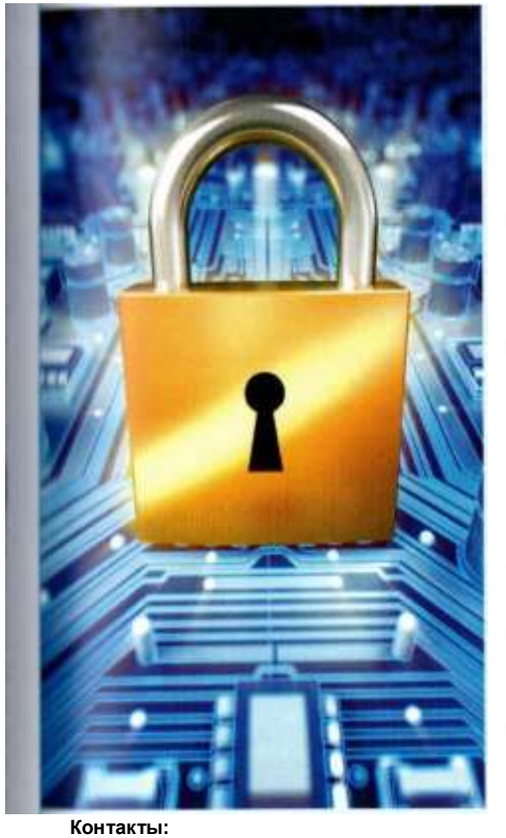
Больше информации:

www.imperva.com.

Компания RRC является дистрибьютором продуктов и решений для обеспечения информационной безопасности, сетевого и телекоммуникационного оборудования, систем автоматической идентификации данных, инфраструктурных решений и систем физической безопасности.

Нп,(Разделение RRC Security следует модели проектного дистрибьютора и поддерживает и портфеле продукты и решения мировых лидеров рынка информационной безопасности: (lieck Point, Radware, McAfee, RSA, Imperva, Websense, Splunk,Tufin, Dell Sonicwall.

Мы предлагаем нашим партнерам:



1. **Предпродажная поддержка:**
   - поддержка технических специалистов
   - демо-оборудование
   - демо-лаборатория
   - поддержка в проведении пилотов и PoC

2. **Актуальный склад (как под регулярные продажи, так и под проекты)**

3. **Послепродажная поддержка (первая линия, консалтинг)**

4. **Маркетинговая поддержка:**
   - поддержка в проведении мероприятий на базе партнера
   - маркетинговые материалы по продуктам и решениям

5. **Поддержка в обучение партнеров**

Контакты:
Дистрибуция, сертификация ФСТЭК, тестирование и демонстрация решений: security@rrc.ru **Техническая поддержка и внедрение:** security_.support@rrc.ru
Маркетинговая поддержка: security@rrc.ru

14 *Безопасность облачных сервисов*

Что такое «облачные сервисы»?

Модель облачных вычислений (Cloud Computing) существенно отличается от традиционной модели вычислений, используемой сегодня повсеместно в корпоративных информационных системах, и это отличие прежде всего проявляется в способе обеспечения безопасности ИС. Природа этого отличия довольна проста — вместо того, чтобы строить собственную ИС и управлять ею силами сотрудников предприятия, предприятие начинает пользоваться услугами ИС, которая была создана сторонней организацией-провайдером. При этом организация-провайдер владеет всей инфраструктурой ИС и управляет ею, обеспечивая в том числе безопасность данных, принадлежащих предприятию-клиенту. Сотрудники предприятия-клиента используют свои компьютеры только как терминалы доступа к облаку, а все данные предприятия, включая личные данные сотрудников, хранятся и обрабатываются где-то там, в облаке. Предприятию-клиенту остаётся только «доверять, но проверять», следуя совету Рональда Рейгана.

Такой революционный переворот в модели вычислений не мог не обеспокоить специалистов по безопасности, и некоторые из них достаточно дружно говорят о том, что облачные вычисления — это вещь хорошая, но отсутствие гарантий безопасности облачных вычислений сводит все преимущества облака на нет.

Для этих опасений безусловно есть основания, однако многое зависит от типа организации-клиента и модели облачных вычислений, которую клиент собирается использовать. Понимание моделей облачных вычислений — это необходимое условие для правильной оценки рисков предприятия при переходе на новый тип ИС, поэтому мы начнём этот раздел с обсуждения понятия «облачные вычисления» и краткого описания моделей облачных сервисов, предлагаемых сегодня на рынке информационных услуг.

По способу реализации облачные среды делятся на публичные, частные и гибридные. ***Публичные облака*** создаются провайдерами

услуг, и их сервисы предоставляются предприятиям или частным лицам как публичным клиентам. *Частные облака* создаются некоторым предприятием для использования только сотрудниками данного предприятия, т. е. как часть корпоративной сети этого предприятия. Г*ибридные облака* — это комбинация публичных и частных облаков, которыми пользуется некоторое предприятие.

Из приведенной классификации понятно, что специфические «облачные» угрозы исходят в основном от публичного облака, именно оно было описано в начале этого раздела и именно этому типу облака посвящён весь последующий материал этого раздела.

Определение облачных вычислений

Существуют различные определения облачных вычислений, приведём только одно — от NIST, так как эта организация, пользующаяся заслуженной репутацией, уделяет большое внимание этому сравнительно новому виду услуг:

«Облачные вычисления — это модель, которая предоставляет удобный доступ по требованию к разделяемому пулу конфигурируемых вычислительных ресурсов, которые могут быть быстро выделены пользователю и отданы обратно в пул с минимальными затратами на управление этим процессом или с минимальным взаимодействием с провайдером услуг.»*

Свойства облачных вычислений

Рассмотрим несколько подробнее *свойства облачных вычислений,* данные в этом определении.

Разделяемые ресурсы (multitenancy). В отличие от прежних моделей вычислений, в которых вычислительные ресурсы принадлежали одному владельцу (ресурсы корпоративной сети хотя и разделяются между пользователями, но принадлежат одной и той же организации- владельцу), облачные вычисления основаны на модели, в которой вычислительные ресурсы разделяются между многими арендаторами- клиентами и владельцем-провайдером на всех уровнях — сети, хоста и приложений.

Масштабируемость в очень широких пределах. Хотя отдельная организация может владеть сотнями или даже несколькими тысячами серверов, облачная среда позволяет масштабировать вычислительную мощность до десятков тысяч серверов и наращивать пропускную способность каналов доступа и объем хранилища данных в разы. Для обеспечения такой масштабируемости провайдер создает большое количество центров данных, распределенных географически.

The NIST Definition of Cloud Computing, Special Publication 800-145, NIST, August 2011.

Эластичность. Пользователи могут быстро наращивать и уменьшать вычислительные ресурсы по мере необходимости. Таким свойством обычная корпоративная ИС не обладает.

Оплата за использованные ресурсы. Пользователи платят только за те ресурсы, которые они действительно использовали и за тот период времени, в течение которого ресурсы использовались.

Самостоятельное выделение ресурсов. Пользователи могут управлять выделением необходимых им ресурсов самостоятельно, через удобный интерфейс, предоставляемый провайдером. Скорее всего, эти операции будут выполнять сотрудники ИТ-отдела предприятия, а не рядовые пользователи.

Технологии облачных вычислений

Важно понимать, что облачные вычисления не столько новая технология, сколько комбинация уже существовавших до этого технологий. Эти технологии развивались с разными скоростями и в разных условиях и не были созданы для работы как единое целое. Однако их смогли «притереть» друг к другу и создать новое качество — облачные среды. Новые достижения в таких областях как производство процессоров, технология виртуализации, производство систем дисковой памяти, широкополосный доступе к Интернету, а также появление быстрых и недорогих серверов внесли свой вклад в превращение облачных вычислений в очень привлекательную модель.

Виртуализация — это основополагающая технологическая платформа, на которой базируются облачные вычисления. Термин «виртуализация» относится к абстрагированию компьютерных ресурсов (процессора, памяти, дисковой памяти, сети, стека протоколов и баз данных) от прикладных программ и конечных пользователей облачного сервиса. Абстрагирование инфраструктуры приводит к такому понятию, как демократизация ресурсов, и создает возможность для потребления ресурсов из пула любым пользователем или его процессами с помощью стандартных методов.

Технологии виртуализации позволяют многочисленным арендаторам облака видеть ресурсы облака как ресурсы, выделенные только для них. Виртуализация применяется не только в облаке, но и в центрах данных, принадлежащих предприятию, — она улучшает использование ресурсов и упрощает операционную эффективность ИТ-отдела. Технологии виртуализации включают виртуализацию компьютера и ОС (VMWare, Хеп), виртуализацию хранилища данных (NAS, SAN), виртуализацию баз данных, виртуализацию приложений (Apache Tomcat, JBoss, Oracle App Server, WebSphere), а также виртуализацию сети (VLAN, MPLS VPN 2-го и 3-го уровня, виртуальные маршрутизаторы).

Модели сервисов облачных вычислений

На сегодня существуют три основных типа моделей сервисов, предоставляемых провайдерами облачных вычислений. Эти модели являются развитием популярных моделей хостинга (таких, как хостинг аппаратных серверов, программных веб-серверов и приложений), так что при рассмотрении вопросов безопасности облачных сред очень полезен опыт предприятия по использованию более традиционных форм хостинга. От того, какая модель облачных сервисов применяется, зависит распределение обязанностей по обеспечению безопасности между провайдером облака и специалистами по безопасности предприятия-клиента.

Модель «Приложения как сервис» (*Software-as-a-Service, SaaS).* Традиционные методы покупки программного обеспечения заключаются в загрузке программы в собственный компьютер в обмен за плату за лицензию на использование этой программы (капитальные затраты). Пользователь также может купить контракт на обслуживание, чтобы получать заплатки и обновления программы. Пользователь сам заботится о совместимости программы с операционной системой, установке обновлений и удовлетворении условий лицензии.

В модели SaaS пользователь не покупает программное обеспечение, а арендует его для использования на условиях подписки или оплаты за использование. Это уже текущие (операционные) расходы. В некоторых случаях сервис может быть бесплатным при его ограниченном использовании (например, программа используется не более часа в сутки).

Обычно сервис SaaS представляет собой законченное решение с точки зрения аппаратной поддержки, программной платформы и технической поддержки.

До появления сервиса SaaS уже существовала услуга хостинга приложений, в которой сотрудники предприятия пользовались каким- лио приложением, установленным на сервере провайдера. Нужно подчеркнуть, что при некоторой схожести услуг модель SaaS отличается от услуг хостинга приложений в двух существенных аспектах. Во- первых, приложение SaaS работает в режиме разделения времени между всеми арендаторами (т. е. различными организациями), которые подписались на доступ к данному приложению, в то время как приложение хостинга выделяется в единоличное пользование организации-клиента (хотя и разделяется между сотрудниками этой организации). Во-вторых, программы для хостинга приложений часто написаны без учета их использования через сеть, в то время как программы для SaaS всегда оптимизированы для сетевого доступа.

Примером сервисов SaaS являются сервисы Google Apps, Microsoft office 365, Apple iWork.

Модель «Платформа как сервис» (*Platform-as-a-Service, PaaS).* В модели PaaS провайдер предоставляет среду для разработчиков программного обеспечения. Провайдер обычно предоставляет набор

средств разработчика (SDK) и стандарты разработки программ, а также каналы для распространения этих программ и механизмы оплаты. Этот сервис направлен на быструю разработку приложений с низким уровнем начальных вложений и использованием устоявшихся каналов для приобретения пользователей программ. Разработчики программ обычно используют готовые программные блоки, предоставляемые провайдером.

Примером этого типа облачных сервисов является Microsoft Azure.

Модель «Инфраструктура как сервис» (The Infrastructure-as-a- Service, IaaS). В традиционной модели хостинга провайдер предоставляет в распоряжение клиента выделенную инфраструктуру для того, чтобы клиент выполнял на ней свои приложения. Модель IaaS тоже обеспечивает клиента необходимой инфраструктурой для выполнения приложений, но эта инфраструктура не является выделенной, она динамически изменяется в зависимости от требований клиента и оплачивается по схеме «оплата за использование». С точки зрения провайдера такая модель может быть реализована на инфраструктуре, которая может гибко реагировать на пики и спады требований каждого клиента и перераспределять ресурсы инфраструктуры в зависимости от ситуации. Клиент платит за количество потреблённых процессорных циклов, оперативной и дисковой памяти, объем сетевого трафика, но не заботится о физической природе ресурсов — делении диска на разделы, способе увеличения объёма памяти, резервировании ресурсов, резервном копировании данных. Провайдер имеет полный контроль над ресурсами инфраструктуры. Клиент в свою очередь имеет контроль над тем, какие операционные системы работают на виртуальных машинах инфраструктуры и какие приложения работают под управлением этих ОС.

Примером этого типа облачных сервисов является Amazon Web Services (AWS).

В зависимости от выбранной модели облачных сервисов, провайдер и клиент имеют различные зоны контроля над компонентами облачных вычислений. Эти различия показаны на рис. 14.1.

Для каждой модели показано распределение зон управления и ответственности между провайдером и клиентом для пяти основных компонент программно-аппаратного комплекса ИС:

- сетевой инфраструктуры (маршрутизаторы, коммутаторы, линии связи);
- хранилища данных;

Рис. 14.1. Распределение контроля и ответственности между провайдером и клиентом в разных моделях облачных сервисов

- серверов;
- виртуальной машины с операционной системой;
- приложений.

Как видно из рисунка, модель SaaS является крайним случаем, так как провайдер имеет полный контроль над всеми компонентами модели и, следовательно, клиент никак не участвует в обеспечении безопасности той части ИС, которая находится в облаке (но, естественно, должен обеспечить безопасность клиентских компьютеров сотрудников предприятия, которые используются как терминалы облачной среды).

При использовании услуг модели IaaS клиент имеет полный контроль над приложениями, работающими в облаке, а провайдер имеет такой же полный контроль над тремя нижними слоями модели. В то же время клиент и провайдер разделяют контроль над виртуальной машиной и операционной системой, работающей в среде этой виртуальной машины. Понятно, что клиент такой модели должен самостоятельно заботиться о безопасности своих приложений — следить за тем, чтобы обновления приложений периодически получались и устанавливались, устанавливать и обслуживать антивирусные программы и программы блокировки спама и выполнять все остальные действия в соответствии с политикой безопасности предприятия, которые относятся к приложениям. Обычно конфигурирование средств безопасности ОС также является делом клиента в этой модели.

Модель PaaS занимает промежуточное положение, когда и провайдер, и клиент разделяют контроль как за средствами виртуализации программного обеспечения (виртуальной машиной, ОС), так и самим прикладными программами, так как, хотя они и разрабатываются специалистами предприятия-клиента, но при этом в них работают программные модули провайдера, собранные в единую программу по

методике провайдера. Контроль над средствами безопасности верхних уровней этой модели также разделяется между провайдером и клиентом.

Преимущества облачных сервисов

Несмотря на то что облачные сервисы приято рассматривать как источник новых угроз безопасности, они могут существенно улучшить информационную безопасность предприятия, особенно если это небольшое предприятие, у которого нет специального подразделения, занимающегося безопасностью информационной системы. Причин такого преимущества облачной модели над традиционной в области безопасности несколько, некоторые из них являются потенциальными, так как предполагают соответствующую корректную организацию процессов управления безопасностью провайдером услуг, а некоторые являются следствием самой парадигмы облачных вычислений.

Специализация администраторов и операторов провайдера облачных сервисов. Провайдер облачных услуг является крупной организацией (иначе он не сможет обеспечить масштабируемость, эластичность и некоторые другие свойства этой модели) и, как всякая крупная организация, может себе позволить специализацию своих администраторов и операторов во всех областях обеспечения информационной безопасности. В результате специалисты провайдера получают большой опыт в распознавании всего спектра угроз, актуальных в настоящее время, а также в установке и конфигурировании средств отражения этих угроз, таких как файерволы, системы IDS/IPS, антивирусные системы и т. п., что, естественно, усиливает безопасность данных клиентов облачных сервисов.

Высокое качество платформы вычислений. Есть две причины, по которым платформа вычислений (сетевая и компьютерная) может оказаться более качественной у облачного провайдера, чем у предприятия-клиента. Первой причиной является ее более высокая степень однородности, которая определяется тем, что провайдер оказывает одни и те же услуги большому количеству клиентов. Поэтому центры данных провайдера, как правило, строятся на одних и тех же моделях маршрутизаторов, коммутаторов, файерволов, серверов и гипервизоров виртуальных машин. Однородность упрощает управление и сокращает количество ошибок конфигурирования, проще становится обеспечение безопасности такой платформы. Вторая причина заключается в масштабности центров данных и сети провайдера облачных сервисов, что позволяет покупать для платформы самые совершенные и производительные компоненты защиты данных, например высокопроизводительный файервол, способный отфильтровывать пакеты интенсивных DoS-атак. Провайдеры облачных услуг стараются сертифицировать свои платформы по различным программам сертификации безопасности, например по ISO 27001, Information Security management Systems — Requirements, для завоевания доверия клиентов, что также повышает вероятность того, что используемая платформа

обладает высоким качеством.

Доступность ресурсов и данных ***(High Availability).*** Высокая степень масштабирования ресурсов облачных провайдеров обеспечивает также высокую доступность этих ресурсов и, как следствие, высокую доступность данных, обрабатываемых этими ресурсами. Избыточность и резервирование ресурсов являются одними из основных принципов построения облачной среды. Обычно облачный провайдер располагает большим количеством центров данных в разных географических точках, а возможно, и странах, при этом реплики данных одного и того же клиента хранятся в нескольких таких центрах как для увеличения производительности за счет приближения данных к пользователям (если это данные публичные, предназначенные для всех пользователей Интернет), так и для обеспечения доступности данных в случае технических отказов или природных катастроф. Избыточность ресурсов также вызвана необходимостью обеспечивать эластичность сервисов, т. е. возможностью быстрого увеличения нагрузки по запросу клиента. В корпоративной сети такую высокую доступность ресурсов создать трудно, так как она чаще всего будет экономически не оправдана.

Поглощение DDoS-атак. Распределённая избыточная инфраструктура центров данных облачного провайдера позволяет также эффективно бороться с DDoS-атаками. Отражение мощной DDoS-атаки является, наверно, наиболее сложной задачей для администратора корпоративной сети, так как фильтрация потока пакетов интенсивностью в десятки, а то и сотни Гбайт/с, направленный на единственную копию сервера через канал связи, требует наличия очень производительного файервола. Такие файерволы, специально созданные для отражения DDoS-атак, существуют. Однако, даже если предприятие может себе позволить приобрести и установить такое весьма дорогостоящее устройство, то остаётся проблема узкого места, которое представляет собой единственная копия атакуемого сервера (скорее всего это будет веб-сервер, возможности которого публиковать открытые данные хотят подавить) и единственный канал связи сервера с Интернетом с фиксированной пропускной способностью. Поэтому в начальной стадии атаки, когда администратор ещё не успел определить признаки отличия пакетов атаки от пакетов легальных пользова-

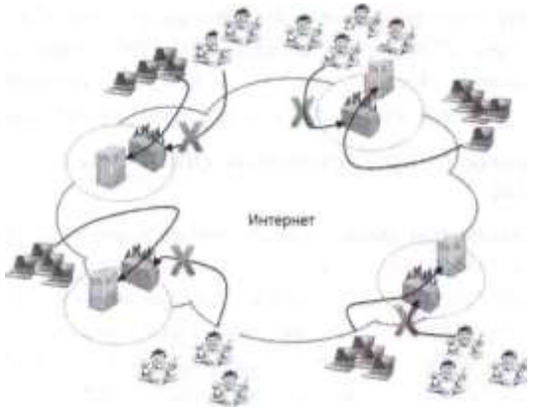


Рис. 14.2. Распределение и поглощение трафика DDoS-атаки инфраструктурой облачного провайдера

телей, файервол принципиально не может защитить сервер от атаки, пропуская весь трафик к серверу и тем самым блокируя его работу.

В сети провайдера облачных услуг отразить DDoS-атаку на ресурсы клиента принципиально проще. На рис. 14.2 показана достаточно типичная структура сети облачного провайдера с четырьмя центрами данных, рассредоточенными по географическим регионам, при этом в каждом центре данных имеется сервер с копией данных некоторого клиента. Для баланса нагрузки провайдер применяет anycast- маршрутизацию (см. раздел «Атаки на корневые DNS-серверы» главы 9), в результате чего трафик запросов клиентов направляется к ближайшему (относительно метрики маршрутизации) серверу.

При возникновении DDoS-атаки трафик ботов злоумышленника также распределяется между серверами провайдера, как и трафик клиентов, поэтому атака не может быть такой же эффективной, как в случае единственного сервера и единственной линии доступа к нему. К тому же провайдер, как правило, поддерживает значительный запас пропускной способности линий доступа для успешной отработки пиков потребностей клиентов, поэтому «забить» линии доступа облачного провайдера злоумышленнику труднее. Можно сказать, что в начальный период DDoS атаки инфраструктура облачного провайдера «впитывает» трафик атаки как губка, без значительного ущерба для трафика легальных пользователей. А так как серверы провайдера защищены высокопроизводительными файерволами, то после обнаружения факта атаки и определения признаков, по которым можно отличить трафик атаки от трафика пользователей, администратор провайдера вносит соответствующие изменения в правила файерволов и они начинают блокировать трафик атаки. Внимательный читатель

может заметить, что только что прочитанное пояснение вполне подходит к описанию DDoS-атак на корневые DNS-серверы из раздела «Атаки на корневые DNS серверы», и это не удивительно, так как механизм поглощения трафика атаки в обоих случаях один и тот же.

Проблемы безопасности облачных сервисов

Ограниченный контроль. Предприятие-клиент облачных сервисов имеет весьма ограниченный контроль над механизмами безопасности своих данных, обрабатываемых виртуальными машинами провайдера и хранящимися в виртуальных хранилищах. Особенно это справедливо для услуг модели SaaS, когда защиты всех элементов ИС, включая прикладные программы пользователя, осуществляется провайдером.

В таких условиях клиент должен стараться получить как можно более полный доступ к сообщениям и журналам средств безопасности, применяемых провайдером, — его виртуального файервола, системы IDS, антивирусных и антиспамовых программ и т. п. Получение информации от средств защиты провайдера в реальном времени (мониторинг) и доступ к историческим данным этих средств должен быть предусмотрен в договоре о предоставлении услуг провайдером. Здесь мы впервые упоминаем этот важный документ, который должен оговаривать все детали взаимоотношений провайдера и клиента, а так как облачные услуги являются очень новым видом телекоммуникационных услуг, то внимание ко всем его деталям должно быть самое пристальное. Кроме ежедневного и исторического мониторинга сообщений средств безопасности провайдера полезно также проводить аудит работы провайдера, привлекая для этого третьи фирмы, пользующиеся устойчивой репутацией в этой области.

Наличие сертификатов на соответствие средств безопасности провайдера популярным программам сертификации также может частично компенсировать отсутствие полного контроля над этими средствами.

Стандартная форма договора с провайдером. Полный набор документов, определяющих взаимоотношения провайдера облачных услуг и предприятие-клиента обычно состоит из нескольких документов.

- *Соглашение об уровне обслуживания* ***(Service Level Agreement, SLA),*** в котором оговариваются количественные параметры услуги провайдера, такие как доступность услуги в процентах времени от некоторого периода, чаще всего месяца (например 0,99 или 0,999), максимальный период отказа услуги (например, 30 мин), предельный уровень нагрузки на прикладную программу (например, не более 100 запросов в минуту) и т. п. В этом документе также должны быть оговорено, каким образом провайдер компенсирует невыполнение параметров SLA.
- *Политика конфиденциальности данных.* Этот документ описывает практику обработки данных пользователя провайдером — способы, с помощью которых данные собираются, используются, хранятся и

уничтожаются после окончания срока действия договора. Важно, чтобы политика конфиденциальности данных провайдера согласовывалась с общей политикой безопасности предприятия-клиента.
- *Политика допустимого использования сервисов* ограничивает действия клиента услуги, например запрещая ему отправлять спам, пользуясь ресурсами провайдера.
- *Условия использования* оговаривают такие важные детали, как лицензирование услуг, границы ответственности за инциденты с данными, а также любые изменения в условиях договора.

Важно понимать, что существуют стандартная форма документов договора и индивидуальная. Стандартная форма договора разработана специалистами и юристами провайдера и отражает его интересы. Такая форма может удовлетворять определённые категории клиентов, но не удовлетворять другие, например по причине несовпадения политик конфиденциальности. Возможна также индивидуальная форма договора, которая строится на основе стандартной, но предусматривает изменения, отражающие особенности клиента. Провайдеры не всегда идут на заключение индивидуальных договоров, особенно для клиентов, не представляющих особой значимости для провайдера (небольшое предприятие, неизвестное предприятие и т. п.). Заключение договора по стандартной форме всегда несет в себе опасность, и необходимо хорошо изучить все его положения, прежде чем соглашаться на них в том случае, когда провайдер отказывается заключать индивидуальный договор.

Разделение инфраструктуры провайдера с другими арендаторами. При использовании услуг облачного провайдера вы разделяете его инфраструктуру с другими арендаторами, которых вы не знаете. При этом разделение ресурсов провайдера осуществляется не на физическом уровне, а с помощью механизмов виртуализации. Злоумышленник может заключить договор с вашим провайдером и попытаться использовать бреши в механизмах виртуализации для получения несанкционированного доступа к вашим данным. Кроме того, нельзя исключать ошибок персонала провайдера, в результате которых виртуальные барьеры могут быть нарушены.

Сложность инфраструктуры провайдера. Инфраструктура облачного провайдера намного сложнее инфраструктуры стандартного цен

тра данных, и это представляет собой дополнительную угрозу безопасности облачных сервисов. Кроме таких стандартных элементов виртуализации, как гипервизоры, виртуальные машины, виртуальные маршрутизаторы и файерволы, такая инфраструктура включает многочисленные дополнительные компоненты управления, такие как средства самостоятельного динамического выделения ресурсов клиентами, измерители потребления ресурсов, управление квотами, управление нагрузкой, мониторинг качества услуг и т. п. Кроме того, облачные сервисы могут быть реализованы в облачной среде другого провайдера, что еще усложняет картину.

Административный доступ через Интернет. Обычно администраторы корпоративных ОС и приложений пользуются доступом к ним через локальную сеть. При использовании облачных сервисов административный доступ должен выполняться через Интернет, что несет дополнительные угрозы. Необходимо убедиться, что облачный провайдер использует только хорошо защищенные соединения при предоставлении административного доступа своим клиентам.

Сложности управления рисками. При обеспечении безопасности облачных сервисов к ним должен быть применен тот же самый системный подход, который применяется и к собственным сервисам корпоративной сети. Системный подход включает процедуры управления рисками, при этом оценка уровня угроз в случае обработки данных облачным провайдером становится еще более сложным делом, чем при оценке уровня угроз ресурсам собственной ИС. В такой ситуации возможно полагаться на результаты аудита услуг провайдера третьей стороной, а также на опыт других клиентов данного провайдера, если это возможно. Можно также по умолчанию считать уровень угроз высоким и соответственно этому проводить оценку риска для всех информационных ресурсов, передаваемых провайдеру.

Конфиденциальность персональных данных. При использовании услуг корпоративной сети персональные данные сотрудников предприятия хранятся в справочной службе предприятия (например, работающей на основе Microsoft Active Directory) и предприятие несет ответственность за их конфиденциальность в соответствии с соответствующим законами и правовыми актами. В том случае, когда сотрудники пользуются услугами облачного провайдера, их персональные данные (имена и пароли) хранятся в справочной службе провайдера. Так как ответственность за их конфиденциальность в конечном счете все равно несет предприятие, то необходимо убедиться, что провайдер надлежащим образом обеспечивает их конфиденциальность как при передаче личных данных через Интернет, так и при хранении их в разделяемой между арендаторами справочной службе провайдера. Для повышения уровня защиты личных данных можно применить раздельные наборы данных для локальной аутентификации

пользователей и для их аутентификации у облачного провайдера. Но это довольно громоздкое решение, и оно может стать нерабочим для большой организации.

Другим решением является применение схемы федеративной аутентификации, описанной в главе 5. В этом случае личные данные пользователей хранятся в справочной службе предприятия, а служба аутентификации провайдера взаимодействует со службой аутентификации предприятия через защищённое соединение Интернета.

Мультинациональность облачных услуг, законодательство и политика. В мире облачных вычислений существуют различные варианты реализации сервисов в отношении того, где данные физически размещены, где они обрабатываются и откуда происходит доступ к этим данным. Зачастую, эти три точки находятся в разных странах, в каждой из которых действует свое законодательство. В разных странах также могут быть зарегистрированы предприятие-клиент и облачный провайдер. Пока что законодательные аспекты таких многонациональных услуг определены плохо, эта работа находится в своей начальной стадии, в результате появляется много неясностей во взаимоотношениях провайдеров и клиентов многонациональных облачных услуг, а значит, риски использования этих услуг весьма высоки. Уменьшить риски, связанные с многонациональными облачными сервисами, можно за счет непосредственного указания в договоре конкретной судебной инстанции определенной страны, которая будут разбирать любые споры между провайдером и клиентом, если они возникнут.

Для того чтобы успешно развиваться как глобальная, не знающая границ услуга, облачные вычисления должны быть отделены от политики.

К сожалению, сегодня существуют глобальные политические и технологические силы и принимаются законы, которые могут оказывать негативное влияние на развитие глобального облака. Например, одним из результатов принятия Соединёнными Штатами Патриотического Акта 2004 года стало то, что Канада решила не использовать серверы Интернета, которые находятся на территории США, опасаясь за конфиденциальность данных, которые Канада хранит на своих компьютерах. Разоблачения Сноудена в отношении программы PRISM привели Бразилию к решению построить свой собственный сегмент Интернет с основными сервисами, поддерживаемыми серверами, находящимися на территории Бразилии.

Поэтому выживание облачных вычислений в значительной степени зависит от глобальной политики.

Значимость облачных сервисов

После выяснения новых видов угроз, связанных с применением публичных облачных сервисов, у корпоративных специалистов по безопасности может возникнуть вопрос — а стоит ли овчинка выделки? Ответ неочевиден, но при его выборе нужно принимать во внимание не только тактические, но и стратегические соображения. А стратегические соображения говорят, что у облачных сред есть большое будущее, их нужно изучать и, возможно, опробовать, имея в виду их потенциальную значимость для развития информационных технологий и не только их.

Хорошую аналогию, поясняющую значение облачных вычислений для развития общества, дал Николас Карр в своей книге The Big Switch*. Он сравнил появление облачных вычислений в информационном веке с электрификацией в индустриальном веке. До электрификации страны каждая организация, каждый производитель и каждый домовладелец должны были сами заботиться о выработке энергии — для этого существовали водяные колеса, ветряные мельницы, паровые машины. С распространением электрификации уже нет нужды вырабатывать энергию самостоятельно, нужно только подключиться к электрической сети. Карр считает, что облачные вычисления открывают новую эру в использовании компьютеров. Сейчас организации обеспечивают себя компьютерными ресурсами самостоятельно. В будущем предполагается, что организация должна просто подключиться к облаку (например, через Интернет, возможно используя защищенные каналы) и получить вычислительные ресурсы, которые ей нужны. И если нужно быстро увеличить вычислительную мощность или объем хранимых данных, то облако легко предоставит эти дополнительные ресурсы своему клиенту, точно так же, как электрическая сеть легко справляется с дополнительной нагрузкой, когда вы включаете микроволновую печь или электрическую дрель в дополнение к работающему телевизору и настольной лампе.

Соображения перспективности и значимости облачных сервисов также должны учитываться администраторами безопасности и руководством предприятия при выработке политики безопасности в отношении использования этих сервисов.

Вопросы к главе 14

X. Модель публичных облачных вычислений характеризуется тем, что:

а) информационная система принадлежит провайдеру, но управляется клиентом;

б) информационная система принадлежит клиенту, но управляется прова- дером;

www.ama2on.co.uk/The-Big-Switch-Rewiring-Edison/dp/0393333949

в) информационная система принадлежит провайдеру и управляется провайдером.

2. Что из перечисленного является свойствами модели публичных облачных вычислений:

а) доступ по требованию к разделяемому пулу вычислительных ресурсов;

б) возможность динамически изменять объем потребляемых вычислительных ресурсов;

в) возможность получать доступ к вычислительным ресурсам независимо от точки подключения к Интернет;

г) минимальное взаимодействие с провайдером при управлении потребляемыми вычислительными ресурсами.

3. Основной технологией, на которой основаны облачные вычисления, является:

а) MPLS VPN;

б) защищенные каналы;

в) виртуализация;

г) криптография.

4. Какие из приведенных ниже утверждений корректно описывают свойства модели «Приложения как сервис» (SaaS):

а) провайдер и клиент разделяют ответственность за обеспечение безопасности серверов и операционной системы;

б) провайдер и клиент разделяют ответственность за обеспечение безопасности сетевой инфраструктуры;

в) провайдер несет ответственность за безопасность всех элементов облачных вычислений.

5. В какой модели облачных вычислений клиент имеет полный контроль над приложениями?

а) приложения как сервис (SaaS);

б) платформа как сервис (PaaS);

в) инфраструктура как сервис (IaaS).

5. Какие факторы позволяют модели облачных вычислений обеспечивать более высокую безопасность информационной системы по сравнению с традиционной моделью вычислений:

а) специализация администраторов и операторов облачных сервисов в определенных областях информационной безопасности;

б) способность поглощать трафик DDoS-атак;

в) высокое качество платформы вычислений;

г) централизация вычислительных ресурсов.

Часть IV

Безопасность системного и прикладного программного обеспечения

Программное обеспечение компьютера — его операционная система (ОС) и прикладные программы — выполняют работу по обработке и хранению информации, поступающей в компьютер из сети и от его локального пользователя. Именно программное обеспечение компьютера делает его тем, чем он стал в современном мире, — усилителем человеческого интеллекта, выполняющего пока рутинные, но так необходимые нам манипуляции с информацией. И если нарушения в работе транспортной инфраструктуры сети ведут к перебоям с поставкой данных, то нарушения в работе программного обеспечения компьютера могут дать нам ложные данные в результате их неверной обработки или же лишить нас огромного количества хранящихся на дисках компьютера документов, фотографий и писем, составляющих значительную часть нашего жизненного багажа. Поэтому безопасность программного обеспечения компьютера не менее важна, чем безопасность сети, так как одно не может работать без другого.

Безопасность программного обеспечения как инженерная и научная отрасль знаний имеет более долгую историю, ведущую начало от изолированных компьютеров 60-х годов, и поэтому иногда рассматривается как единственная область компьютерной безопасности — но это не так. Обе составляющие информационного процесса — передача по сети и обработка и хранение в компьютере — определяют в конечном счете результат, эти составляющие влияют друг на друга и требуют взаимного учета. Поэтому при рассмотрении в данном разделе основных аспектов безопасности программ мы всегда имеем в виду и их сетевую составляющую.

Программное обеспечение может представлять угрозу безопасности информационной системы не только из-за своих внутренних проблем. Злоумышленники также используют всю мощь программ для организации весьма изощрённых атак на информационную систему, пытаясь нарушить нормальную работу как извне, направляя трафик на атакуемые компьютеры по сети, так и внедряя вредоносные коды вирусов, троянских коней, червей и тому подобных агентов влияния в программное обеспечение атакуемых компьютеров.

Для того чтобы защитить программное обеспечение от внутренних ошибок и внешних угроз, операционные системы и прикладные программы

структурируются таким образом, чтобы уменьшить взаимное влияние вычислительных процессов различных пользователей и создать преграды на пути распространения ошибок и вредоносных программ. Кроме того, операционные системы включают специальные программные модули для выполнения таких необходимых функций обеспечения безопасности, как аутентификация и авторизация пользователей, аудит событий безопасности, антивирусный контроль, фильтрация подозрительного сетевого трафика с помощью программных файерволов и некоторых других.

15 *Архитектурная безопасность ОС*

Безопасность ОС обеспечивается различными способами и не в последнюю очередь — ее архитектурой. Под *архитектурой* понимают структурную организацию ОС на основе различных программных модулей. Модули должны иметь четкое функциональное назначение с чётко оговоренными правилами взаимодействия. Ясное понимание роли каждого отдельного модуля существенно упрощает работу по модификации и развитию системы.

Функциональная сложность операционной системы неизбежно приводит к сложности ее архитектуры. Обычно в состав ОС входят исполняемые и объектные модули стандартных для данной ОС форматов, библиотеки разных типов, модули исходного текста программ, программные модули специального формата (например, загрузчик ОС, драйверы ввода-вывода), конфигурационные файлы, файлы документации, модули справочной системы и т. д.

Большинство современных операционных систем представляют собой хорошо структурированные модульные системы, способные к развитию, расширению и переносу на новые платформы. Какой-либо единой архитектуры ОС не существует, но существуют универсальные подходы к структурированию ОС, и безопасность ОС играет не последнюю роль в этих подходах.

На самом верхнем уровне обобщения эта структура состоит из трех типов программных систем:

- сетевая операционная система
- сетевые службы;
- сетевые приложения.

Сетевая операционная система и сетевые службы

Операционную систему компьютера часто определяют как взаимосвязанный набор системных программ, который обеспечивает эффективное управление ресурсами компьютера (памятью, процессором, внешними устройствами, файлами и др.), а также предоставляет пользователю удобный интерфейс для работы с аппаратурой компьютера и для разработки приложений.

Говоря о ***сетевой*** ОС мы, очевидно, должны расширить границы управляемых ресурсов за пределы одного компьютера.

Сетевой операционной системой называют операционную систему компьютера, которая помимо управления локальными ресурсами предоставляет пользователям и приложениям возможность эффективного и удобного доступа к информационным и аппаратным ресурсам других компьютеров сети.

Сегодня практически все операционные системы являются сетевыми.

Сетевой службой называется совокупность программных средств, которые предоставляют доступ пользователей к определенному типу ресурсов через сеть.

В зависимости от типа ресурсов сетевая служба может быть файловой службой, службой печати, почтовой службой и так далее.

Сетевая служба выполняет набор базовых операций, необходимых для использования ресурса определенного типа различными прикладными программами.

Например, потребность в доступе к удаленному принтеру может возникать у пользователей самых разных приложений: текстового редактора, графического редактора, системы управления базой данных (СУБД). Очевидно, что дублирование в каждом из приложений общих для всех них функций по организации удаленной печати является избыточным.

Более эффективным представляется подход, при котором эти функции исключаются из приложений и оформляются в виде пары специализированных программных модулей — клиента и сервера печати (рис. 15.1), функции которых ранее выполнялись соответственно приложениями на компьютерах ***A*** и ***B.*** Теперь эта пара клиент- сервер может быть использована любым приложением, выполняемым на компьютере ***A.***

Обобщая такой подход применительно к другим типам разделяемых ресурсов, дадим следующие определения:

Клиент — это модуль, предназначенный для формирования и передачи сообщений-запросов к ресурсам удаленного компьютера от разных приложений с последующим приемом результатов из сети и передачей их соответствующим приложениям.

Сервер — это модуль, который постоянно ожидает прихода из сети запросов от клиентов и, приняв запрос, пытается его обслужить, как правило, с участием локальной ОС; один сервер может обслуживать запросы сразу нескольких клиентов (поочередно или одновременно).

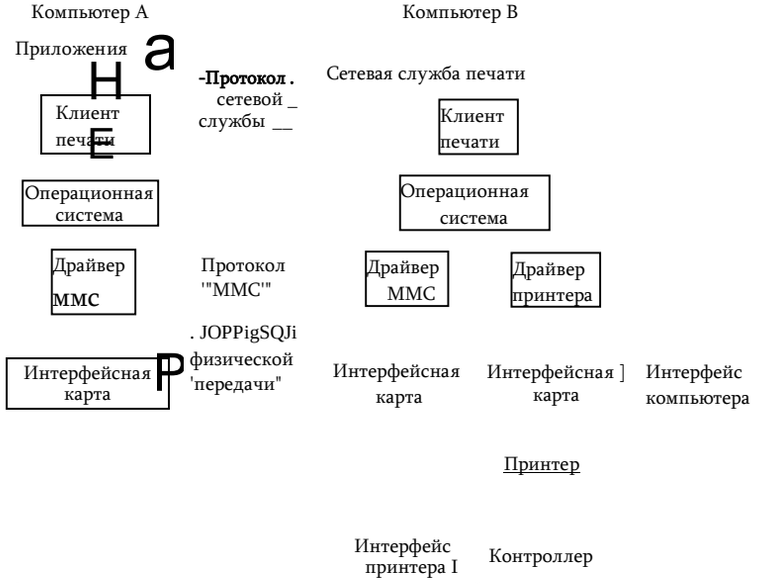


Рис. 15.1. Совместное использование принтера в компьютерной сети с помощью сетевой службы печати

Пара клиент-сервер, предоставляющая доступ к конкретному типу ресурса компьютера через сеть, образует *сетевую службу.*

Каждая служба связана с определенным типом сетевых ресурсов. Так, на рис. 15.1 модули клиента и сервера, реализующие удаленный доступ к принтеру, образуют сетевую *службу печати*

Файловая служба позволяет получать доступ к файлам, хранящимся на диске других компьютеров. Серверный компонент файловой службы называют ***файл-сервером.***

Веб вместе с веб-сервером образуют сетевую *веб-службу.* Разделяемым ресурсом в данном случае является веб-сайт — определенным образом организованный набор файлов, содержащих связанную в смысловом отношении информацию и хранящихся на внешнем накопителе веб-сервера.

На схеме веб-службы, показанной на рис. 15.2, два компьютера связаны через Интернет. Для того чтобы отразить этот факт графически, мы поместили между двумя компьютерами так называемое ***коммуникационное облако,*** которое позволяет нам абстрагироваться от всех деталей среды передачи сообщений. Обмен сообщениями между клиентской и серверной частями веб-службы выполняется по стандартному протоколу HTTP и никак не зависит от того, передаются ли эти сообщения «из рук в руки» (от интерфейса одного компьютера к интерфейсу другого) или через большое число посредников — промежуточных коммуникационных устройств.

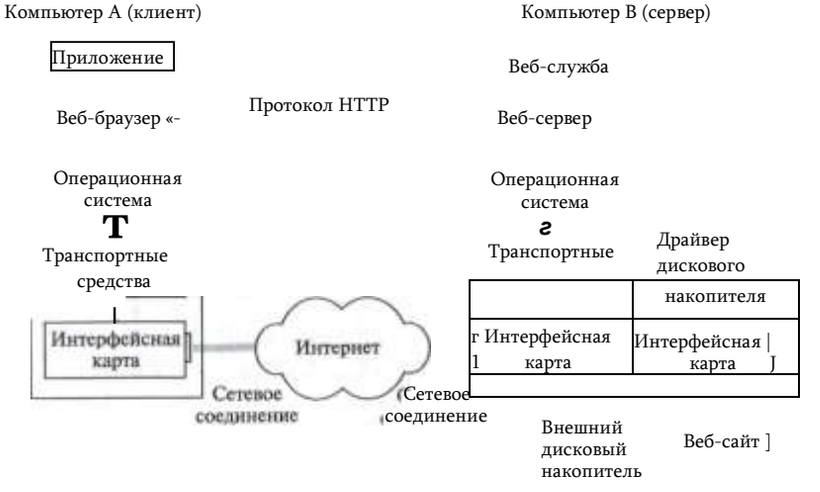


Рис. 15.2. Веб-служба

Для того чтобы сообщения протокола HTTP надежно доставлялись через Интернет от клиента к серверу и обратно, на обоих компь- терах должно иметься программное обеспечение стека TCP/IP, которое образует *транспортные средства* компьютера.

Из рассмотренных примеров мы видим, что удаленный доступ к сетевым ресурсам обеспечивается:

- сетевыми службами;
- средствами транспортировки сообщений по сети — транспортными средствами.

Следовательно, именно эти функциональные модули должны быть добавлены к ОС, чтобы она могла называться сетевой (рис. 15.3).

Среди сетевых служб можно выделить такие, которые ориентированы не на простого пользователя, как, например, файловая служба или служба печати, а на администратора. Такие службы направлены на организацию работы сети. Например, *централизованная справочная служба,* или *служба каталогов,* предназначена для ведения базы данных о пользователях сети,

обо всех ее программных и аппаратных компонентах33.

От того, насколько богатый набор сетевых служб и услуг предлагает операционная система конечным пользователям, приложениям и администраторам сети, зависит ее позиция в общем ряду сетевых ОС.

33 Другими примерами являются *служба мониторинга сети,* позволяющая захватывать и анализировать сетевой трафик, *служба безопасности,* в функции которой может входить, в частности, выполнение процедуры логического входа с проверкой пароля, *служба резервного копирования и архивирования.*

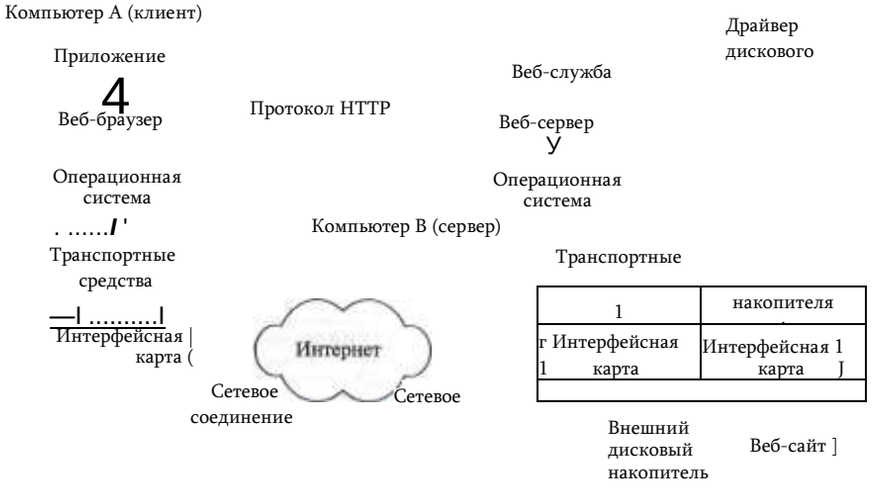


Рис. 15.3. Функциональные компоненты сетевой ОС

1/1 сетевые службы, и транспортные средства могут являться не-отъемлемыми (встроенными) компонентами ОС или существовать в виде отдельных программных продуктов. Например, сетевая файловая служба обычно встраивается в ОС, а вот веб-браузер чаще всего приобретается отдельно. Типичная сетевая ОС имеет в своем составе широкий набор драйверов и протокольных модулей, однако у пользователя, как правило, есть возможность дополнить этот стандартный набор необходимыми ему программами. Решение о способе реализации клиентов и серверов сетевой службы, драйверов и протокольных модулей принимается разработчиками с учетом самых разных соображений: технических, коммерческих и даже юридических.

Сетевая служба может быть представлена в ОС либо обеими (клиентской и серверной) частями, либо только одной из них.

В первом случае операционная система, называемая ***одноранговой,*** не только позволяет обращаться к ресурсам других компьютеров, но и предоставляет собственные ресурсы в распоряжение пользователей других компьютеров. Например, если на всех компьютерах сети установлены и клиенты, и серверы файловой службы, то все пользователи сети могут совместно применять файлы друг друга. Компьютеры, совмещающие функции клиента и сервера, называют одноранговыми узлами.

Операционная система, которая преимущественно содержит клиентские части сетевых служб, называется ***клиентской.*** Клиентские ОС устанавливаются на компьютеры, обращающиеся с запросами к ресурсам

других компьютеров сети. За такими компьютерами, также

Рис. 15.4. Локальное приложение

называемыми клиентскими, работают рядовые пользователи. Обычно клиентские компьютеры относятся к классу относительно простых устройств.

К другому типу операционных систем относится ***серверная ОС —*** она ориентирована на обработку запросов из сети к ресурсам своего компьютера и включает в себя в основном серверные части сетевых служб. Компьютер с установленной на нем серверной ОС, занимающийся исключительно обслуживанием запросов других компьютеров, называют ***выделенным сервером*** сети. За выделенным сервером, как правило, обычные пользователи не работают.

Сетевые приложения

Компьютер, подключенный к сети, может выполнять следующие типы приложений.

Локальное приложение целиком выполняется на данном компьютере и использует только локальные ресурсы (рис. 15.4). Для такого приложения не требуется никаких сетевых средств, оно может быть выполнено на автономно работающем компьютере.

Централизованное сетевое приложение целиком выполняется на данном компьютере, но обращается в процессе своего выполнения к ресурсам других компьютеров сети. В примере на рис. 15.5 приложение, которое выполняется на клиентском компьютере, обрабатывает данные из файла, хранящегося на файловом сервере, а затем распечатывает результаты на принтере, подключенном к серверу печати. Очевидно, что работа такого типа приложений невозможна без участия сетевых служб и средств транспортировки сообщений.

Распределенное (сетевое) приложение состоит из нескольких взаимодействующих частей, каждая из которых выполняет какую-то определенную законченную работу по решению прикладной задачи, при-

Централизованное сетевое приложение

Рис. 15.5. Централизованное сетевое при ложение

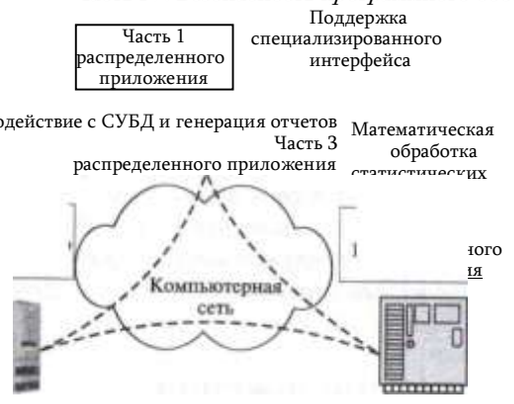


Рис. 15.6. Распределенное сетевое приложение

чем каждая часть может выполняться и, как правило, выполняется на отдельном компьютере сети (рис. 15.6). Части распределенного приложения взаимодействуют друг с другом, используя сетевые службы и транспортные средства ОС. Распределенное приложение в общем случае имеет доступ ко всем ресурсам компьютерной сети.

Заметим, что все сетевые службы, включая файловую службу, службу печати, службу электронной почты, службу удаленного доступа, Интернет-телефонию и т. д., по определению относятся к классу распределенных приложений. Действительно, любая сетевая служба включает в себя клиентскую и серверную части, которые могут и обычно выполняются на разных компьютерах.

На рис. 15.7, иллюстрирующем распределенный характер вебслужбы, мы видим различные виды клиентских устройств — персональные компьютеры, ноутбуки и мобильные телефоны — с установленными на них веб-браузерами, которые взаимодействуют по сети с веб-сервером. Таким образом, с одним и тем же веб-сайтом может одновременно работать множество — сотни и тысячи — сетевых пользователей.

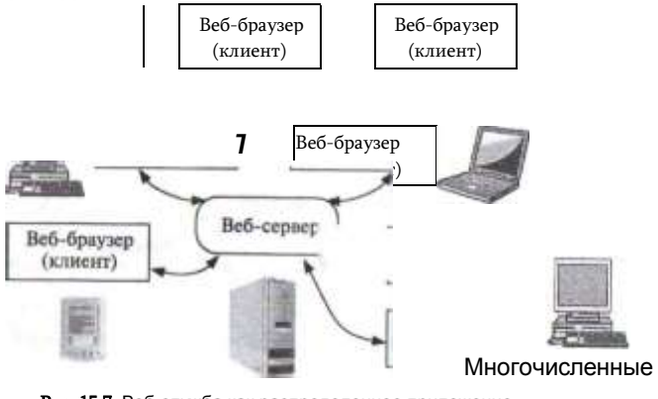


Рис. 15.7. Веб-служба как распределенное приложение

Многочисленные примеры распределенных приложений можно встретить и в такой области, как

обработка данных научных экспериментов. Это не удивительно, так как многие эксперименты порождают такие большие объемы данных, генерируемых в реальном времени, которые просто невозможно обработать на одном, даже очень мощном, суперкомпьютере. Кроме того, алгоритмы обработки экспериментальных данных часто легко распараллеливаются, что также важно для успешного применения взаимосвязанных компьютеров с целью решения какой-либо общей задачи.

Ядро и вспомогательные модули ОС

Наиболее общим подходом к структуризации операционной системы является разделение всех ее модулей на две группы:

- ядро — модули, выполняющие основные функции ОС;
- вспомогательные модули ОС.

Модули ***ядра*** выполняют базовые функции ОС, связанные с управлением процессами, памятью, устройствами ввода-вывода и т. п. Именно ядро занимается переключением контекстов, загрузкой/выгрузкой станиц, обработкой прерываний. Непосредственное выполнение такого рода действий недоступно для приложений. При необходимости они могут обращаться к ядру с системными вызовами, используя для этого имеющийся в их распоряжении интерфейс прикладного программирования — API.

Функции, отнесенные в ведение ядра, являются наиболее часто используемыми функциями операционной системы, поэтому скорость их выполнения определяет производительность системы в целом. Для обеспечения высокой скорости работы ОС все модули ядра или большая их часть постоянно находятся в оперативной памяти, т. е. являются ***резидентными.***

Обычно ядро оформляется в виде программного модуля некоторого специального формата, отличающегося от формата пользовательских приложений.

Ядро является движущей силой всех вычислительных процессов в компьютерной системе, и крах ядра равносилен краху всей системы. Поэтому разработчики операционной системы уделяют особое внимание надежности кодов ядра.

Вспомогательные модули ОС, к которым относятся и сетевые службы, выполняют весьма полезные, но менее обязательные функции. Например, к таким модулям могут быть отнесены программы архивирования данных на внешних носителях sout, дефрагментации

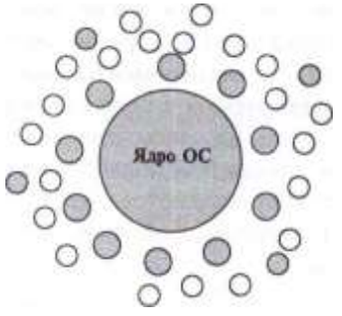


Рис. 15.8. Нечеткость границы между ОС и приложениями

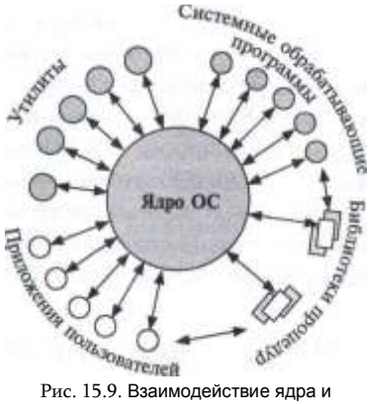


Рис. 15.9. Взаимодействие ядра и вспомогательных модулей ОС

диска, текстового редактора. Вспомогательные модули ОС оформляются либо в виде приложений, либо в виде библиотек процедур.

Поскольку некоторые компоненты ОС оформлены как обычные приложения, т. е. в виде исполняемых модулей стандартного для данной ОС формата, то часто бывает очень сложно провести четкую грань между операционной системой и приложениями (рис. 15.8).

Решение о том, должна ли какая-либо программа или служба стать частью ОС или нет, принимает производитель ОС. Среди многих факторов, способных повлиять на это решение, немаловажными являются перспективы того, будет ли программа иметь массовый спрос у потенциальных пользователей данной ОС.

Как и обычные приложения, для выполнения своих задач утилиты, обрабатывающие программы и библиотеки ОС обращаются к функциям ядра посредством системных вызовов (рис. 15.9).

Модули ОС, оформленные в виде утилит, системных обрабатывающих программ и библиотек, обычно загружаются в оперативную память только на время выполнения своих функций, т. е. являются ***транзитными.*** Постоянно в оперативной памяти располагаются только самые необходимые коды ОС, составляющие ее ядро. Такая организация ОС экономит оперативную память компьютера.

Важным свойством архитектуры ОС, основанной на ядре, является возможность защиты кодов и данных операционной системы за счет выполнения функций ядра в привилегированном режиме.

Ядро в привилегированном режиме

Для надежного управления ходом выполнения приложений операционная система должна иметь по отношению к приложениям определенные привилегии. Иначе некорректно работающее приложение

Рис. 15.10. Архитектура операционной системы с ядром в привилегированном режиме

может вмешаться в работу ОС и, например, разрушить часть ее кодов. Все усилия разработчиков операционной системы окажутся напрасными, если их решения воплощены в незащищенные от приложений модули системы, какими бы элегантными и эффективными эти решения ни были. Операционная система должна обладать исключительными полномочиями также для того, чтобы играть роль арбитра в споре приложений за ресурсы компьютера в мультипрограммном режиме. Ни одно приложение не должно иметь возможности без ведома ОС получать дополнительные области памяти, занимать процессор дольше разрешенного операционной системой периода времени, непосредственно управлять совместно используемыми внешними устройствами.

Обеспечить привилегии операционной системе невозможно без специальных средств аппаратной поддержки. Аппаратура компьютера должна поддерживать как минимум два режима работы — ***пользовательский*** (user mode) и ***привилегированный,*** который также называют ***режимом ядра*** (kernel mode) или ***супервизора*** (supervisor mode). Подразумевается, что операционная система или некоторые ее части работают в привилегированном режиме, а приложения — в пользовательском режиме.

Так как основные функции ОС выполняются ядром, то чаще всего именно ядро становится той частью ОС, которая работает в привилегированном режиме (рис. 15.10). Иногда это свойство — работа в привилегированном режиме — служит основным определением понятия «ядро».

Службы и приложения ставятся в подчиненное положение за счет запрета для них выполнения в пользовательском режиме некоторых критичных команд (инструкций), связанных с переключением процессора с задачи на задачу, управлением устройствами ввода-вывода, доступом к механизмам распределения и защиты памяти. Выполнение некоторых команд в пользовательском режиме запрещается безусловно (очевидно, что к таким командам относится команда перехода в привилегированный режим), тогда как другие запрещается выполнять только при определенных условиях. Например, команды ввода-вывода

могут быть запрещены приложениям при доступе к контроллеру жесткого диска, который хранит данные, общие для ОС и всех приложений, но разрешены при доступе к последовательному порту, выделенному в монопольное владение определенного приложения. Важно, что условия разрешения выполнения критичных команд находятся под полным контролем ОС и этот контроль обеспечивается за счет набора команд, безусловно запрещенных для пользовательского режима.

Аналогичным образом обеспечиваются привилегии ОС при доступе к памяти. Например, выполнение команды доступа к памяти для приложения разрешается, если она обращается к области памяти, отведенной данному приложению операционной системой, и запрещается при обращении к областям памяти, занимаемым ОС или другими приложениями. Полный контроль ОС над доступом к памяти достигается за счет того, что команды конфигурирования механизмов защиты памяти (например, изменения ключей защиты памяти в мейнфреймов IBM или указателя таблицы дескрипторов памяти в процессорах Intel) разрешается выполнять только в привилегированном режиме.

Очень важно, что механизмы защиты памяти используются операционной системой не только для защиты своих областей памяти от приложений, но и для защиты областей памяти, выделенных ОС какому-либо приложению, от остальных приложений. Говорят, что каждое приложение работает в своем ***адресном пространстве.*** Это свойство позволяет локализовать некорректно работающее приложение в собственной области памяти, так что его ошибки не оказывают влияния на остальные приложения и операционную систему.

Защита данных за счет изоляции адресных пространств вычислительных процессов, принадлежащих разным пользователям, применима только по отношению к информации, находящейся в оперативной памяти компьютера. Информация, хранящаяся в файлах на дисках и других накопителях, защищается от несанкционированного доступа другими способами.

Между количеством уровней привилегий, реализуемых аппаратно, и количеством уровней привилегий, поддерживаемых ОС, нет прямого соответствия. Так, в свое время на базе четырех уровней, поддерживаемых процессорами компании Intel x8б, операционная система OS/2 реализовала трехуровневую систему привилегий, а операционные системы семейств Windows и Unix — двухуровневую. В то же время, если аппаратура поддерживает хотя бы два уровня привилегий, то ОС может на этой основе создать программным способом сколь угодно развитую систему защиты. Эта система может, например, поддерживать несколько уровней привилегий, образующих иерархию, соответствующую концепции мандатного доступа. Наличие нескольких уровней привилегий позволяет более тонко распределять полномочия

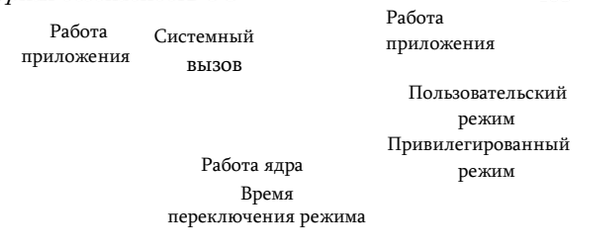


Рис. 15.11. Смена режимов при выполнении системного вызова к привилегированному ядру

как между модулями операционной системы, так и между самими приложениями. Появление внутри операционной системы более привилегированных и менее привилегированных частей дает возможность повысить устойчивость ОС к внутренним ошибкам программных кодов, так как такие ошибки будут распространяться только внутри модулей с определенным уровнем привилегий. Дифференциация привилегий в среде прикладных модулей позволяет строить сложные прикладные комплексы, в которых часть более привилегированных модулей может, например, получать доступ к данным менее привилегированных модулей и управлять их выполнением.

На основе двух режимов привилегий процессора ОС может построить сложную систему индивидуальной защиты ресурсов, примером которой является типичная система защиты файлов и каталогов. Такая система позволяет задать для любого пользователя определенные права доступа к каждому из файлов и каталогов.

Повышение устойчивости операционной системы, обеспечиваемое переходом ядра в привилегированный режим, достигается за счет некоторого замедления выполнения системных вызовов. Системный вызов инициирует переключение процессора из пользовательского режима в привилегированный, а при возврате к приложению — переключение из привилегированного режима в пользовательский (рис. 15.11). Во всех типах процессоров из-за дополнительной двукратной задержки переключения переход на процедуру со сменой режима выполняется медленнее, чем вызов процедуры без смены режима.

Архитектура ОС, основанная на привилегированном ядре и приложениях пользовательского режима, стала по существу классической. Ее используют большинство популярных операционных систем, в том числе семейства ОС MS Windows, Unix/Linux, Mac OS X, операционные системы мейнфреймов.

Микроядерная архитектура

Концепция

Микроядерная архитектура ОС интересна тем, что она существенно увеличивает архитектурную безопасность ОС. Расплатой за

это является снижение производительности ОС. Из-за потерь в производительности микроядерный подход в чистом виде не реализуется в современных ОС, однако компромиссные его варианты, в которых соблюдается желаемый баланс между архитектурной безопасностью и производительностью, находят применение и поэтому знание этого принципа полезно для специалиста в области информационной безопасности.

Микроядерная архитектура является альтернативой классическому варианту построения операционной системы. Под классической архитектурой в данном случае понимается рассмотренная ранее структурная организация ОС, в соответствии с которой все основные функции операционной системы, составляющие многослойное ядро, выполняются в привилегированном режиме. При этом некоторые вспомогательные функции ОС оформляются в виде приложений и выполняются в пользовательском режиме наряду с обычными пользовательскими программами (становясь системными утилитами или обрабатывающими программами). Каждое приложение пользовательского режима работает в собственном адресном пространстве и защищено тем самым от какого-либо вмешательства других приложений. Код ядра, выполняемый в привилегированном режиме, имеет доступ к областям памяти всех приложений, но сам полностью от них защищен. Приложения обращаются к ядру с запросами на выполнение системных функций.

В *микроядерных ОС* в привилегированном режиме остается работать только очень небольшая часть ОС, называемая ***микроядром.*** Все остальные высокоуровневые функции ядра оформляются в виде приложений, работающих в пользовательском режиме.

Микроядро защищено от остальных частей ОС и приложений (рис. 15.12). В состав микроядра обычно входят машинно-зависимые модули, а также модули, выполняющие базовые (но не все!) функции ядра по управлению процессами, обработке прерываний, управлению виртуальной памятью, пересылке сообщений и управлению устройствами ввода-вывода, связанные с загрузкой или чтением регистров устройств. Набор функций микроядра обычно соответствует функциям слоя базовых механизмов обычного ядра. Такие функции операционной системы трудно, если не невозможно, выполнить в пользовательском пространстве.

Все остальные более высокоуровневые функции ядра оформляются в виде приложений, работающих в пользовательском режиме. Однозначного решения о том, какие из системных функций нужно оставить в привилегированном режиме, а какие перенести в пользовательский, не существует. В общем случае многие менеджеры ресурсов, являющиеся неотъемлемыми частями обычного ядра, — файло-

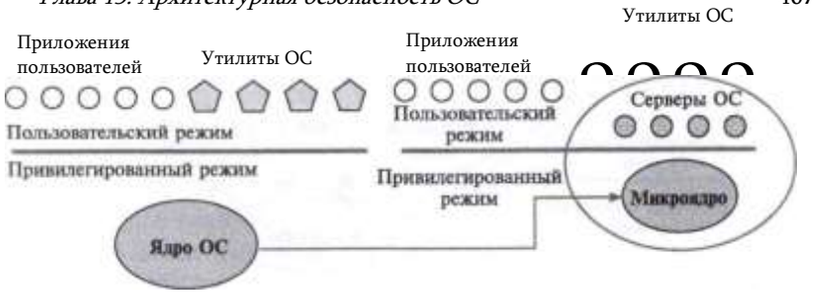


Рис. 15.12. Перенос основного объема функций ядра в пользовательское пространство

вая система, подсистемы управления виртуальной памятью и процессами, менеджер безопасности и т. п. — становятся «периферийными» модулями, работающими в пользовательском режиме.

Работающие в пользовательском режиме менеджеры ресурсов имеют принципиальные отличия от традиционных утилит и обрабатывающих программ операционной системы, хотя при микроядерной архитектуре все эти программные компоненты также оформлены в виде приложений. Утилиты и обрабатывающие программы вызываются в основном пользователями. Ситуации, когда одному приложению требуется выполнение функции (процедуры) другого приложения, возникают крайне редко. Поэтому в операционных системах с классической архитектурой отсутствует механизм, с помощью которого одно приложение могло бы вызывать функции другого.

Совсем другая ситуация возникает, когда в форме приложения оформляется часть операционной системы. По определению, основным назначением такого приложения является обслуживание запросов других приложений, например создание процесса, выделение памяти, проверка прав доступа к ресурсу и т. д. Именно поэтому менеджеры ресурсов, вынесенные в пользовательский режим, называются ***серверами ОС,*** т. е. модулями, основным назначением которых является обслуживание запросов локальных приложений и других модулей ОС. Очевидно, что для реализации микроядерной архитектуры необходимым условием является наличие в операционной системе удобного и эффективного механизма вызова процедур одного процесса из другого процесса. Поддержка такого механизма и является одной из главных задач микроядра.

Схематично механизм обращения к функциям ОС, оформленным в виде серверов, выглядит следующим образом (рис. 15.13). Клиент, которым может быть либо прикладная программа, либо другой компонент ОС, запрашивает выполнение некоторой функции у соответствующего сервера, посылая ему сообщение. Непосредственная

передача сообщений между

Рис. 15.13. **Реализация системного вызова в микроядерной архитектуре**

приложениями невозможна, так как их адресные пространства изолированы друг от друга. Микроядро, выполняющееся в привилегированном режиме, имеет доступ к адресным пространствам каждого из этих приложений и поэтому может работать в качестве посредника. Микроядро сначала передает сообщение, содержащее имя и параметры вызываемой процедуры нужному серверу, затем сервер выполняет запрошенную операцию, после чего ядро возвращает результаты клиенту с помощью другого сообщения. Таким образом, работа микроядерной операционной системы соответствует известной модели клиент-сервер, в которой роль транспортных средств исполняет микроядро.

Преимущества и недостатки микроядерной архитектуры

Микроядерная архитектура в высокой степени удовлетворяет большинству требований, предъявляемых к современным ОС: она способствует переносимости, расширяемости, повышению надежности системы и создает хорошие предпосылки для поддержки распределенных приложений. За эти достоинства приходится платить снижением производительности, и это является основным недостатком микроядерной архитектуры.

Высокая степень *переносимости* обусловлена тем, что весь машинно-зависимый код изолирован в микроядре, поэтому для переноса системы на новый процессор требуется меньше изменений и все они логически сгруппированы вместе.

Расширяемость присуща микроядерной ОС в очень высокой степени. В традиционных системах даже при наличии многослойной структуры нелегко удалить один слой и заменить его на другой по причине множественности и размытости интерфейсов между слоями. Добавление новых функций и изменение существующих требует хорошего знания операционной системы и больших затрат времени. В то же время ограниченный набор четко определенных интерфейсов микроядра открывает путь к упорядоченному росту и эволюции

ОС. Добавление новой подсистемы требует разработки нового приложения, что никак не затрагивает целостность микроядра. Микроядерная структура позволяет не только добавлять, но и сокращать число компонентов операционной системы, что также бывает очень полезно. Например, не всем пользователям нужны средства безопасности или поддержки распределенных вычислений, а удаление их из традиционного ядра чаще всего невозможно. Обычно традиционные операционные системы позволяют динамически добавлять в ядро или удалять из ядра только драйверы внешних устройств — ввиду частых изменений в конфигурации подключенных к компьютеру внешних устройств подсистема ввода-вывода ядра допускает загрузку и выгрузку драйверов «на ходу», но для этого она разрабатывается особым образом (например, среда STREAMS в Unix). При микроядерном подходе *конфигурируемость* ОС не вызывает никаких проблем и не требует особых мер — достаточно изменить файл с параметрами начальной конфигурации системы или же остановить не нужные больше серверы в ходе работы обычными для остановки приложений средствами.

Использование микроядерной модели повышает *безопасность* ОС. Каждый сервер выполняется в виде отдельного процесса в собственной области памяти и, таким образом, защищен от других серверов операционной системы, что не наблюдается в традиционной ОС, где все модули ядра могут влиять друг на друга. И если отдельный сервер терпит крах, то он может быть перезапущен без останова или повреждения остальных серверов ОС. Более того, поскольку серверы выполняются в пользовательском режиме, они не имеют непосредственного доступа к аппаратуре и не могут модифицировать память, в которой хранится и работает микроядро. Другим потенциальным источником повышения надежности ОС является уменьшенный объем кода микроядра по сравнению с традиционным ядром — это снижает вероятность появления ошибок программирования.

Модель с микроядром хорошо подходит для поддержки *распределенных вычислений,* так как использует механизмы, аналогичные сетевым: взаимодействие клиентов и серверов путем обмена сообщениями. Серверы микроядерной ОС могут работать как на одном, так и на разных компьютерах. В этом случае при получении сообщения от приложения микроядро может обработать его самостоятельно и передать локальному серверу или же переслать по сети микроядру, работающему на другом компьютере. Переход к распределенной обработке требует минимальных изменений в работе операционной системы — просто локальный транспорт заменяется сетевым.

Производительность. При классической организации ОС выполнение системного вызова сопровождается двумя переключениями режимов (рис. 15.14,а), а при микроядерной организации — четырьмя

Рис. 15.14. Смена режимов при выполнении системного вызова

(рис. 15.14,6). Таким образом, операционная система на основе микроядра при прочих равных условиях всегда будет менее производительной, чем ОС с классическим ядром.

Серьезность этого недостатка хорошо иллюстрирует история развития ОС Windows. В первых версиях Windows NT 3.1 и 3.5 диспетчер окон, графическая библиотека и высокоуровневые драйверы графических устройств входили в состав сервера пользовательского режима и вызов функций этих модулей осуществлялся в соответствии с микроядерной схемой. Однако очень скоро разработчики Windows NT поняли, что такой механизм обращений к часто используемым функциям графического интерфейса существенно замедляет работу приложений, делая данную операционную систему уязвимой в условиях острой конкуренции. В результате в версию Windows NT 4.0 были внесены существенные изменения — все перечисленные модули были перенесены в ядро, что отдалило эту ОС от идеальной микроядерной архитектуры, но зато резко повысило ее производительность.

Этот пример иллюстрирует главную проблему, с которой сталкиваются разработчики операционной системы, решившие применить микроядерный подход, — что включать в микроядро, а что выносить в пользовательское пространство. В идеальном случае микроядро может состоять только из средств передачи сообщений, средств взаимодействия с аппаратурой, в том числе средств доступа к механизмам привилегированной защиты. Такое микроядро было создано в ходе проекта Mach университета Карнеги-Мэллона, проект был завершен в 1994 году выпуском версии Mach 3.0, но из-за низкой производительности коммерческого продолжения проект не получил. Тем не менее коды и идеи Mach вошли впоследствии в различные ОС, например в Apple OS X и Microsoft Windows, но в переработанном виде, а не в виде чистого микроядра (возможно, тут сказалось и влияние личности, так как один из ведущих разработчиков Mach стал работать на Microsoft, а другой — на Apple).

Разработчики не всегда жестко придерживаются принципа минимизации функций ядра, часто жертвуя этим ради повышения производительности. В

результате реализации микроядерных ОС образуют некоторый спектр, на одном краю которого находятся системы с минимально возможным микроядром (например, Unix-подобная операционная система реального времени QNX), а на другом — современные системы семейства Windows и Apple OS X, в которых микроядро нагружено выполнением достаточно большого объема функций, так что с точки зрения чистой теории микроядром не является.

Вопросы к главе 15

1. Программное обеспечение компьютера, работающего в сети, состоит из следующих основных программных систем:

а) сетевая операционная система;

б) драйвер сетевого адаптера;

в) сетевые службы;

г) сетевые приложения.

2. Являются ли сетевой службой программа, реализующая функции протокола IP?

а) да;

б) нет.

3. Какими свойствами должна обладать программа для того, чтобы разработчик ОМ обязательно включил бы ее в состав ядра ОС:

а) выполнять наиболее часто используемые функции ОС;

б) обладать высоким быстродействием;

в) выполнять элементарные операции с аппаратными ресурсами компьютера, например, обрабатывать страничные прерывания;

г) обладать высокой надежностью.

4. Ядро ОС работает в привилегированном режиме процессора для того, чтобы:

а) программы ядра выполнялись быстрее;

б) получить возможность выполнять привилегированные команды процессора;

в) защитить программы ядра от некорректной работы приложений;

г) обеспечить безопасность ОС.

5. Каким минимальным числом уровней привилегий должен обладать процессор для построения эффективной системы управления доступом к ресурсам компьютера?

а) 3;

б) 2;

в) в соответствии с количеством уровней системы управления доступом.

6. Какие из приведенных терминов являются синонимами:

а) привилегированный режим;

б) режим супервизора;

в) пользовательский режим;

г) режим ядра.

7. Можно ли, анализируя двоичный код программы, сделать вывод о невозможности ее выполнения в пользовательском режиме?

а) да;

б) нет.

8. Что заставляет разработчиков ОС отходить от принципа микроядерной арки тектуры и расширять ядро за счет включения в него функций, которые можно были бы реализовать в виде процессов-серверов?

а) ненадежность архитектуры микроядра;

б) низкая производительность архитектуры микроядра;

в) требования безопасности ОС.

9. Правильно ли утверждение: «Микроядерная архитектура ОС повышает бм опасность ОС»?

а) да;

б) нет.

16 *Аутентификация и управление доступом в ОС*

Аутентификация пользователей в ОС

Система аутентификации пользователей является своего рода передовой линией фронта в борьбе за безопасность информации. Ошибки системы аутентификации исправить невозможно, какими бы совершенными не были остальные средства безопасности.

Система аутентификации ОС работает в пределах одного компьютера — она используют базу аутентификационных данных пользователей этого компьютера и результаты аутентификации могут быть использованы только для доступа к ресурсам этого компьютера.

В отличие от системы аутентификации ОС существуют *системы аутентификации домена,* которые основаны на центральной базе аутентификационных данных пользователей, хранящейся на одном из серверов, а результаты аутентификации используются для доступа к ресурсам группы компьютеров, называемых ***обычно*** *доменом аутентификации,* например к компьютерам некоторого предприятия. В этом разделе мы рассмотрим только системы аутентификации ОС, а системы аутентификации домена изучим после рассмотрения систем управления доступом ОС, так как обе эти составляющие —• аутентификации и авторизации — в централизованных системах тесно связаны.

Система аутентификации ОС работает как при логическом входе пользователя с ***терминала*** компьютера, так и при логическом входе пользователя через ***сеть.*** Первый вариант называют ***локальным*** или ***интерактивным*** логическим входом, а второй ***— удаленным,*** или ***сетевым,*** или ***неинтерактивным*** логическим входом.

Понятно, что при удаленном логическом входе риски безопасности выше, так как аутентификационные данные передаются через сеть — корпоративную или Интернет — и их легче перехватить. Перехват данных процедуры аутентификации представляет собой угрозу даже в том случае, когда пароль не передается в открытом виде по

сети или же не передается вовсе34 — при наличии большого массива аутентификационных данных, т. е. данных перехватов большого количества процедур входа одного и того же пользователя, пароль может быть вычислен по имеющимся результатам его применения.

Необходимо заметить, что иногда систему аутентификации ОС называют «локальной системой аутентификации». В этом случае хотят подчеркнуть тот факт, что аутентификационные данные этой системы имеют локальный характер, т. е. распространяются на пользователей только данного компьютера. При этом сама процедура логического входа может быть как локальной, так и удаленной.

Системы аутентификации ОС поддерживают все распространённые методы аутентификации, описанные в главе 5 и использующие:

- многоразовые пароли;
- одноразовые пароли (аппаратные и программные);
- биометрические данные;
- цифровые сертификаты.

Основным методом аутентификации пользователей ОС является метод, основанный на использовании *многоразового пароля.* Практически все универсальные ОС, такие как MS Windows, Unix/Linux и MAC OS X, используют этот метод по умолчанию.

Одноразовые пароли, обеспечивающие более надёжную аутентификацию, чем многоразовые, чаще используются при удалённом логическом входе через соединения VPN с шифрованием информации, где передача аутентификационной информации идёт через Интернет и, следовательно, риск ее перехвата и взлома особенно велик. Одноразовые пароли могут сочетаться с многоразовыми при двух-факторной аутентификации.

Аутентификация на основе сертификатов используется чаще всего при использовании смарт-карт, которые хранят сертификат пользователя, выданный сервером сертификации организации, к которой принадлежит пользователь.

Аутентификация на основе биометрических данных штатными средствами универсальных ОС обычно не поддерживается, так как их сверхнадежность нужна только в особо защищённых системах и, кроме того, для поддержки биометрической аутентификации нужны специальные устройства. Поэтому для организации биометрической аутентификации необходимо приобрести и установить соответствующее специальное программное и аппаратное обеспечение.

Необходимо отличать процедуру аутентификации пользователя операционной системы от *процедуры аутентификации пользователя серверной части некоторого приложения.* Многие серверные приложения имеют собственную систему аутентификации пользователей, никак не связанную с системой аутентификации ОС, под управлением которой они работают. Например, так работают многие реализации FTP-сервера, сервера баз данных.

Независимость системы аутентификации сервера приложений имеет как

34 Имеется в виду способ, когда по сети передается слово-вызов, зашифрованное с помощью пароля; этот способ был рассмотрен в главе 5.

свои положительные, так и отрицательные стороны. Преимуществом здесь является разграничение по умолчанию пользователей ОС, которым потенциально может понадобиться доступ к любому ресурсу компьютера, и пользователей некоторого сервиса, которым нужен доступ только к ресурсам, относящихся к данному сервису, например только к файлам, хранящимся в корневом каталоге FTP-сервера. К недостаткам же можно отнести следующее:

- Ведение двух или более (по числу сетевых приложений) баз данных пользователей может приводить к дополнительным сложностям для пользователей, например к необходимости запоминания двух различных имен и паролей для одного и того же пользователя, если он является пользователем ОС и пользователем сервиса; ошибкам администраторов ОС и сервиса из-за дублирования учетных записей и т. п.
- Е1изкая защищенность протокола аутентификации некоторых приложений. Классическим примером является сервер FTP, который использует протокол аутентификации с открытой передачей многоразового пароля по сети. Из-за этого сервис FTP не рекомендуется использовать при доступе к нему через Интернет, заменяя его более защищенным сервисом scp или SFTP.

Управление доступом в ОС

В универсальных ОС доминирует дискреционная модель управления доступа, согласно которой владелец ресурса (пользователь, который его создал или которому передано владение) самостоятельно определяет, кто имеет доступ к этому ресурсу и какие операции с ним он может выполнять. Практически все популярные сегодня семейства универсальных ОС — Unix/Linux/CentOS/Ubuntu, Mac OS X, MS Windows — используют дискреционную модель доступа как основную. Мандатная модель — это особенность специализированных ОС, рассчитанных на применение в среде с повышенными требованиями к безопасности. Тем не менее существует набор модулей ядра Linux под названием SELinux (Security Enhanced Linux), который реализует многие свойства мандатной модели в среде Linux.

Модель ролевого управления доступом используется в универсальных ОС частично, в виде механизма встроенных групп с предопределенными правами, сосуществуя с моделью дискреционного доступа для индивидуальных пользователей и групп.

В дальнейшим мы сосредоточимся на рассмотрении особенности реализации дискреционной модели в ОС семейства Unix и Windows.

Аутентификация пользователей в ОС Windows

При оценке безопасности системы аутентификации на основе многоразовых паролей важно рассмотреть следующие ее аспекты:

- структуру системы аутентификации;
- протокол аутентификации, который определяет способ использования пароля, в том числе и передачу его или результата его применения (например, хеш слова-вызова) по сети;
- политику выбора пароля (длина, специальные символы) и его периодической смены;
- способ хранения пароля;
- ограничения логического входа: по времени, по месту (с каких машин), по числу одновременных сессий.

В ОС семейства Windows свойства системы аутентификации пользователей в значительной степени зависят от того, какая модель аутентификации применяется: *система аутентификации ОС* или *доменная система аутентификации.*

Доменная система аутентификации является предпочтительным вариантом, он рекомендуется компанией Microsoft даже в том случае, если сеть состоит из нескольких рабочих станций и одного сервера.

Доменная система аутентификации основана на распределенной справочной службе Microsoft Active Directory, она обладает более мощными средствами управления политикой безопасности, в том числе и политикой управления учетными данными пользователей. В этом разделе мы сосредоточимся на свойствах локальной системы аутентификации, а доменную аутентификацию рассмотрим позже в отдельном разделе.

Структура и протоколы системы аутентификации ОС Windows

В ОС семейства Windows принят модульный подход к построению системы аутентификации, позволяющий использовать различные протоколы аутентификации. Этот подход иллюстрируется рис. 16.1. На нем показаны основные модули этой системы, причем также и те, которые относятся к доменной системе аутентификации по той простой причине, что без них трудно понять работу системы в целом.

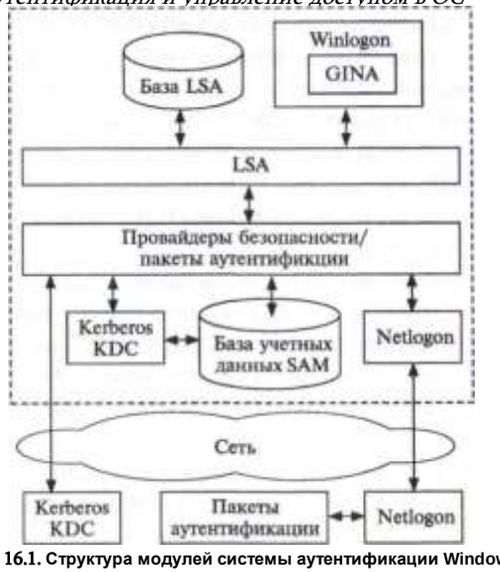


Рис. 16.1. Структура модулей системы аутентификации Windows

Winlogon является сервисом Windows, который отвечает за ло- ический вход пользователя в систему, в том числе за интерактивный вход, который выполняется с помощью модуля GINA (Graphical Identification and Autgentication). Модуль GINA отвечает за показ на экране графической панели, с помощью которой пользователь вводит с вое имя и пароль. В ОС Windows существует два режима вызова этой панели — с помощью набора пользователем комбинации клавиш CTRL-Alt-Del или с помощью выбора иконки пользователя на так называемой панели приглашения Windows (Welcome Screen). Последний вариант строго не рекомендуется к использованию в среде, требующей высокой безопасности, так как он не защищает от «ложной» панели логического входа, когда злоумышленнику удается внедрить в ОС вредоносный код, имитирующий панель логического входа для перехвата паролей пользователей. Набор комбинации CTRL+ALT+ DEL предотвращает такую возможность, так как он перехватывается на уровне ядра системы. Сервис Winlogon также ответственен за так называемую «удаленную интерактивную» аутентификацию пользователей, когда доступ к графической оболочке ОС выполняется по сети с помощью программы Remote Desktop.

Сервис локальной безопасности LSA (Local Security Authority) координирует работу остальных модулей системы аутентификации при выполнении интерактивного логического входа. Этот сервис получает от сервиса Winlogon имя и пароль пользователя и передает их для

аутентификации одному из пакетов аутентификации, установленных в системе. Однако перед этим сервис LSA проверяет свою базу данных, в которой содержатся правила политики логического входа, для проверки ограничений, например может ли данный пользователь выполнить интерактивный вход в данный компьютер в данный день недели и данное время? В базе данных LSA хранятся также и другие правила политики безопасности, а также некоторые права пользователя на выполнение системных действий.

В ОС Windows может быть установлено несколько пакетов аутентификации, в штатную конфигурацию входят пакеты Kerberos и MSV1JX Эти пакеты работают совместно с провайдерами безопасности — программными модулями, обеспечивающими стандартный интерфейс ***SSPI*** (*Security Services Programming Interface)* между пакетами аутентификации и приложениями.

Пакет Kerberos используется только для доменной системы аутентификации. Пакет MSV1_0 может поддерживать как локальную, так и доменную систему аутентификации.

При интерактивной аутентификации пакет MSV1_0 вычисляет хеш введенного пользователем пароля и сравнивает его с хешем пароля, хранящемся в локальной базе данных учетных записей пользователей ***SAM*** *{Security Accounts Manager database).* База SAM является частью Windows Registry (\HKEYJ_OCAL_MACHINE\SAM), хорошо защищенной от доступа обычных пользователей и даже членов группы Administrators (но не пользователя Administrator).

Пакет MSV1JD работает с различными версиями функций хеширования паролей — это сделано для обратной совместимости с ранними версиями Windows. Поэтому в MSV1_0 реализована поддержка трех видов хеша: LM, NTLMvl и NTLMv2.

Хеш LM* не обладает высокой криптостойкостью и довольно легко взламывается существующими программами взлома паролей. Версия NTLMvl была промежуточным вариантом, не усилившим стойкость хеша пароля значительно, поэтому рекомендуемой версией является NTLMv2. Именно она и используется по умолчанию для интерактивной аутентификации.

В том случае, когда пакет MSVIJD работает с доменной системой аутентификации, он обращается не к локальной базе SAM, а вызывает сервис Netlogon, который по сети взаимодействует с удаленным сервисом Netlogon на основе метода передачи слова-запроса, исключающего передачу пароля или его хеша по сети.

LM — это аббревиатура от LAN Manager, файлового начально разработанного IBM и Microsoft для OS/2. **сервера, перво-**

Политика управления паролями

Политика управления паролями в ОС Windows является достаточно гибкой. Правила этой политики можно задавать локально, и тгда они хранятся в базе LSA, а также в пределах домена — тогда они хранятся в базе Active Directory.

Рассмотрим для примера, как задаются правила локальной политики паролей в Windows Vista и Windows 7. Их можно задать с помощью утилиты администрирования Local Security Policy, выбрав пункт меню Password Policy. С помощью этой утилиты можно задать следующие правила политики паролей.

Правило ведения журнала паролей: позволяет задать число уникальных новых паролей (от 0 до 24) перед тем, как пользователь может использовать старый пароль вновь.

Правило, устанавливающее максимальный срок действия пароля: период времени в днях (от 1 до 999), в течение которого пароль может быть использован.

Правило, устанавливающее минимальный срок действия пароля: период времени в днях (от 0 до 998), в течение которого пользователь не может сменить действующий пароль; должен быть меньше, чем максимальный срок действия.

Правило, устанавливающее минимальную длину пароля: значение должно быть от 1 до 14, 0 означает отсутствие пароля. Рекомендуется использовать пароли не менее 12 символов длиной.

Правила, устанавливающие требования к сложности пароля:

- не содержать имени пользователя или частей его полного имени, превышающих два символа;
- содержать комбинацию как минимум из трех указанных ниже четырех категорий символов: прописные буквы, строчные буквы, цифры, специальные символы (например. !, $, #, %).

Правило хранения паролей, которое может потребовать хранить пароли в зашифрованном виде, причем использовать для шифрования паролей не односторонние функции шифрования, а обратимые функции шифрования, которые при необходимости позволяли бы расшифровывать пароли. Применение этой опции ведет к снижению защищенности ОС, поэтому использовать ее нужно очень осторожно.

Аутентификация пользователей в ОС Unix

Обзор средств аутентификации Unix

Семейство ОС Unix выросло из многотерминальной многопользовательской операционной системы для мини-компьютера PDP-11,

и ее традиционный интерактивный вход с алфавитно-цифрового терминала (или программы, его эмулирующей) с аутентификацией по паролю по-прежнему является основным способом логического входа в систему.

Утилита login поддерживает процедуру ***интерактивного входа*** в текстовом режиме, обращаясь для аутентификации к хешам паролей пользователей, хранящимся в файле /etc/passwd. С появлением графических интерфейсов пользователя, таких как GNOME. KDE, текстовый режим работы утилиты login был дополнен ее графическим вариантом, однако ее суть работы от этого не изменилась.

Удаленный сетевой вход пользователей Unix во времена сравнительно безопасного Интернета выполнялся с помощью протоко- ***natelnet,*** который является протоколом эмуляции текстового терминала поверх транспортных средств стека TCP/IP. Протокол telnet передает символ, набираемые пользователем, и ответы ОС в открытом виде, поэтому перехватить пароль, набираемый клиентом telnet, не составляет труда.

Из-за незащищенности telnet его строго не рекомендуется использовать для аутентификации пользователей Unix через Интернет; да и крупная корпоративная сеть также может представлять опасность для такой передачи.

Основным современным средством для ***интерактивного доступа*** к ОС Unix является многофункциональный пакет программ ***SSH*** (Secure Shell), который поддерживает различные методы защищенной аутентификации, а также некоторые виды защищённых операций с файлами через сеть.

Как и ОС Windows, семейство ОС Unix/Linux позволяет использовать для аутентификации не только локальные системы, использующие файл паролей, хранящийся на диске локального компьютера, но и централизованные системы, такие как NIS, NIS+ , Open Directory, Kerberos.

Для возможности работы с различными пакетами аутентификации компания Sun разработала программную среду ***PAM (Pluggeable Authentication Modules),*** которая стала популярной и существует сегодня в разных вариантах, в том числе и как программное обеспечение с открытым кодом (Linux PAM, Open PAM).

PAM представляет собой мультиплексор PAM Library, который предоставляет услуги аутентификации различным приложениям, таким как login или SSH, с помощью стандартного API (рис. 16.2). Получив запрос на аутентификацию, мультиплексор передает его одному из пакетов аутентификации, установленному в ОС. Этот пакет должен быть написан с учетом архитектуры PAM и поддерживать ее программный интерфейс SPI.

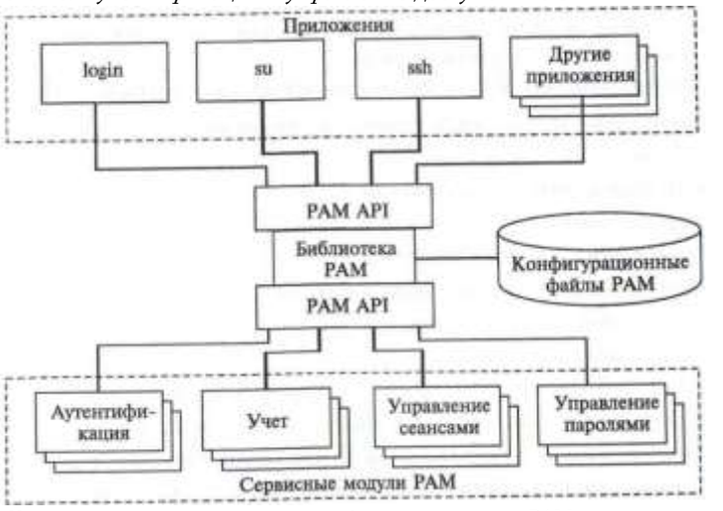


Рис. 16.2. **Мультиплексирование пакетов аутентификации с помощью** PAM

Как видно из рис. 16.2, архитектура PAM предусматривает наличие не только пакетов аутентификации, но и пакетов управления учетными записями, сессиями пользователей и паролями. На практике модули PAM функционально не обязательно дифференцируются точно в соответствии с данной схемой, например модуль pam.unix из проекта Linux PAM* поддерживает традиционную аутентификацию Unix по паролю и выполняет функции как аутентификации, так и управления паролями, учетными записями и сессиями пользователей.

Рассмотрим более подробно два аспекта локальной системы аутентификации Unix: безопасное хранение паролей и аутентификацию с помощью протокола SSH.

Хранение паролей

В ранних версиях Unix пароли пользователей (точнее — их хеши) хранились в файле passwd (находящего в каталоге административных утилит /etc) вместе с учетными данными пользователей. Доступ к файлу passwd для чтения был разрешен всем, а запись в него — только суперпользователю root. Формат файла passwd не изменился с тех пор, и это удобно для многочисленных приложений, которые его используют. Рассмотрим следующий фрагмент файла passwd: tomcat:**x:91:91**:Apache Tomcat:/usr/share/tomcat6:/sbin/nologin webalizer:x:67:67:Webalizer:/var/www/usage:/sbin/nologin sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin

http://www.linux-pam.org/Linux-PAM-html/sag-pam.unix.html

```
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash tcpdump:x:72:72::/:/sbin/nologin
victoro:x:500:500:Victor Olifer:/home/victoro:/bin/bash
```

Мы видим, что каждая запись файла passwd состоит из следующих 7 полей:

- логическое имя пользователя:
- хеш пароля;
- идентификатор пользователя;
- идентификатор группы пользователя;
- полное имя пользователя;
- домашний каталог;
- версия оболочки Unix (shell), которая запускается для данного пользователя после его успешного логического входа.

Со временем все больше атак стало осуществляться за счет того, что даже непривилегированный пользователь мог легко прочитать хеши паролей и взломать их с помощью одной из многочисленных программ, имеющихся в свободном доступе. Одним из вариантов усиления защиты паролей могло бы быть лишение непривилегированных пользователей права читать файл passwd. Однако последствия такого решения могли бы быть неприятными для тех утилит Unix, которые работают от имени непривилегированного пользователя и читают из файла passwd не пароли, а такие атрибуты пользователя, как его домашний каталог или его полное имя.

Поэтому популярным решением стало сохранение файла passwd в его обычном виде с разрешением чтения всем, с переносом хешей паролей в отдельный файл shadow. Этот файл также находится в каталоге /etc, однако доступ к нему как по чтению, так и по записи имеет только суперпользователь root. Записи файла shadow, соответствующие записям файла passwd из приведенного выше примера, будут такими:

```
tomcat:!!:15866::::::
webalizer:!!:15866::::::
sshd:!!:15866::::::
postgres:!!:15866::::::
tcpdump:!!:15866::::::
victoro:$6$rzn7RYTyEWi3nKFS$gqXtf73qUuNVLZNR2jkoKu41M9q0nwkL5Vmqyy/
Zrlq 68poIdDWik72HWLKhuWDx4VJ3/lyyCfnpTQOpLOkcnO:15866:0:99999:7::: Формат
```

записи файла shadow следующий:

- логическое имя пользователя;
- поле хеша пароля, которое состоит из трех подполей, разделённых знаком «$»: подполя условного номера односторонней функции, подполя криптографической «соли», добавленной к паролю при его хешировании, и подполя собственно хеша, такой формат можно описать как $function$salt$hash;
- количество дней, прошедших с 1 января 1970 года (момент начала отсчета времени Unix — Unix epoch) до последнего изменения пароля;
- количество дней до разрешенной смены пароля (минимальный срок действия пароля);
- количество дней до необходимой смены пароля (максимальный срок действия пароля);
- количество дней до предупреждения об истечении пароля;
- количество дней до неактивности учетной записи;
- количество дней с 1 января 1970 года до истечения действия учетной записи.

Возможно, внимательный читатель заметил, что в приведенном примере записях файла passwd в поле хеша пароля у всех записей стоит символ «x». Это означает, что в данной версии Unix используется файл shadow, в котором и находятся хеши паролей. Для пояснения нужно заметить, что в записях файла shadow хеш пароля имеется только для записи victoro, а у остальных записей, которые соответствуют системным сервисам, пароль отсутствует, о чем говорит значение «И» вместо хеша.

Аутентификация по протоколу SSH Протокол ***SSH (Secure Shell)*** состоит из нескольких протоколов, или слоев, решающих различные задачи безопасной аутентификации, а также безопасной передачи данных по сети. Первая версия SSH была разработана в 1995 году Тату Илоненом, бывшим в то время исследователем Технологического университета Хельсинки. Протокол SSH быстро приобрел большую популярность, за годы его существования было разработано много как открытых, так и коммерческих его версий. Сегодня наиболее распространенной является версия ***Open SSH*** v.2, клиент и сервер которой включены практически во все свободно распространяемые версии Unix/Linux: Fedora, CentOS, Ubuntu и другие.

Архитектура протокола SSH и многие его функциональные свойства стандартизованы IETF в многочисленных RFC, например RFC 4251 описывает общую архитектуру SSH, RFC 4252 — протокол (слой) аутентификации SSFH, RFC 4253 — транспортный протокол SSH, а RFC 4254 — протокол управления соединениями.

Протокол SSFH работает в архитектуре клиент-сервер, серверная часть представлена демоном sshd, который работает в режиме сервиса на хостах, к которым пользователи хотят получить доступ. Клиентская часть представлена утилитой ssh, которая выполняет запрос на логический вход в удаленный хост, такой, например, как:

```
> ssh victor@ganymede.co.uk
```

Транспортный протокол SSHответственен за первоначальный обмен ключами, аутентификацию сервера и поддержание защищенного канала с обеспечением шифрования данных и проверкой их целостности. Транспортный протокол может работать с различными асимметричными алгоритмами шифрования, например RSA или DSA.

Использование ключей. Протокол SSH использует технику открытых ключей в двух целях.

- Для шифрования данных. Если пара ключей, открытый и закрытый, используется только для шифрования данных, но не для

аутентификации, то в этом случае она генерируется автоматически на каждом конце соединения, после чего открытый ключ передается по сети напарнику для шифрования данных.

- Для аутентификации. В случае, когда открытый ключ некоторого удаленного сервера используется для его аутентификации, пара ключей должна быть сгенерирована вручную и аутентичность открытого ключа должна быть подтверждена либо его помещением в специальный файл хоста-клиента, либо с помощью публичной системы PKI (есть также вариант хранения публичных ключей в записях DNS).

Аутентификация с помощью SSH может осуществляться различными способами.

Аутентификация на основе имен хостов. Пользователь аутентифицируется положительно, если:

а) если имя хоста, с которого пользователь осуществляет логический вход, содержится в файле /etc/hosts.equiv или файле /etc/ssh/ shosts.equiv на удаленном хосте, в который он пытается войти, и имя пользователя одно и то же в обоих хостах;

б) или же имя хоста, с которого пользователь осуществляет логический вход, содержится в файле ~/.rhosts или ~/.shosts в домашнем каталоге пользователя на удаленном хосте, причем это имя сопровождается именем пользователя на клиентской машине;

в) а также удаленный хост смог верифицировать открытый ключ клиентского компьютера, предъявленный в процессе данной сессии аутентификации, сравнивая его с ключами, хранящимися в своих файлах ~/.ssh/known_hosts и /etc/known_hosts. В эти файлы помещаются ключи всех хостов, с которых пользователь успешно совершил в прошлом логический вход в данный удаленный хост, используя другие методы аутентификации (например, с помощью пароля). Если же предъявленный ключ клиентского компьютера не совпадает с имеющимся для него ключом из файлов ~/.ssh/known_hosts и /etc/known_hosts, то выдается предупреждение о возможной фальсификации и аутентификация не считается успешной. Проверка ключа предохраняет от .1 гак типа IP-спуфинга или DNS-спуфинга.

Аутентификация с помощью публичных ключей. Пользователь должен сначала сгенерировать пару ключей, пользуясь утилитой ssh- keygen. Эта утилита позволяет генерировать ключи RSA или DSA.

(генерированные ключи помещаются в файлы хоста пользователя — i.(крытый ключ в файл ~/.ssh/id_dsa (для протокола DSA) или /.ssh/id_rsa (для протокола RSA), а публичный ключ — в файл / ssh/id_dsa.pub (протокол DSA) или ~/.ssh/id_rsa.pub (протокол RSA). После этого пользователь должен скопировать открытый ключ и файл ~/.ssh/authorized_keys, находящийся в своем домашнем каталоге на том сервере, в который он хочет удаленно входить. После этого он может выполнить удаленный вход по SSH без использования пароля, так как уделенный сервер найдет открытый ключ

хоста клиента в своем файле и сможет с его помощью расшифровать аутентификационную информацию, зашифрованную клиентом с помощью соответствующего закрытого ключа и переданную в ходе процедуры установления соединения.

Аутентификация на основе слова-вызова. Это стандартная процедура аутентификации с использованием слова-вызова — его передает удаленный хост, клиент вычисляет от него хеш, используя ключ на основе хеша своего пароля, и возвращает результат удаленному хосту. Удаленный хост должен иметь хеш пароля пользователя в своем файле паролей, при этом он его использует аналогичным образом для получения хеша слова-вызова. В случае совпадения результата, переданного клиентом, со своим удаленный хост считает процедуру аутентификации успешной. Передача слова-вызова и результата происходит по защищенному каналу, который образуется на основе автоматически сгенерированных ключей хоста и сервера.

Аутентификация на основе пароля. Отличается от описанной выше процедуры тем, что клиент передает пароль по сети, но так как пароль передается по защищенному каналу, то этот способ считается также защищенным, хотя и менее защищенным, чем предыдущие.

Клиент SSH пытается использовать различные способы аутентификации в том порядке, в котором они описаны выше, т. е. начиная с аутентификации на основе хостов и кончая аутентификацией на основе паролей (переход к более низкоприоритетному способу аутентификации происходит в том случае, если очередной способ аутентификации не поддерживается обеими сторонами). В то же время пользователь может задать предпочтительный способ аутентификации принудительно, использовав переменную PreferredAuthentications в файле конфигурации клиента ssh_config или же задавая эту переменную как опцию команды SSH.

Как видно из описания, программное обеспечение SSH является гибким и достаточно мощным, предоставляя весьма надежные средства для удаленной аутентификации пользователей.

Кроме аутентификации во время логического входа программы пакета SSH могут использоваться для таких полезных операций, как защищенное копирование файлов между удаленными хостами (с помощью утилиты scp) или запуск приложения на удаленном хосте с отображением графического интерфейса пользователя этого приложения на хосте клиента (при логическом входе с ключом — X).

Контроль доступа в ОС Unix

Файловая модель доступа

Файловая модель является основой абстрактного представления ресурсов всех типов: файлов, каталогов, принтеров, устройств ввода- вывода.

В ОС Unix права доступа к файлу или каталогу определяются для трех субъектов:

- владельца файла (идентификатор User ID, UID);
- членов группы, к которой принадлежит владелец (Group ID, GID);
- всех остальных пользователей системы.

С учетом того, что в Unix определены всего три операции над файлами и каталогами (*чтение, запись, выполнение),* характеристики безопасности файла включают девять признаков, задающих возможность выполнения каждой из трех операций для каждого из трех субъектов доступа. Например, если владелец файла разрешил себе выполнение всех трех операций, для членов группы — чтение и выполнение, а для всех остальных пользователей — только выполнение, то девять характеристик безопасности файла выглядят следующим образом:

rwx г-х г--------

Здесь г, w и х обозначают операции чтения, записи и выполнения соответственно. Именно в таком виде выводит информацию о правах доступа к файлам команда просмотра содержимого каталога ls. Суперпользователю Unix с именем root все виды доступа позволены всегда, поэтому его идентификатор (он имеет значение 0) не фигурирует в списках управления доступом. Суперпользователь root — это элемент ролевой модели в Unix.

С каждым процессом Unix связаны два идентификатора: пользователя, от имени которого был создан этот процесс, и группы, которой принадлежит данный пользователь. Эти идентификаторы называются *реальным идентификатором пользователя (Real User ID,* ***RUID)*** и *реальным идентификатором группы (Real Group ID,* ***RGID).***

Однако, как показано на рис. 16.3, при проверке прав доступа к файлу используются не эти идентификаторы, а так называемые *эффективные идентификаторы пользователя (Effective User ID,* ***EUID)*** и *группы (Effective Group ID,* ***EG ID).***

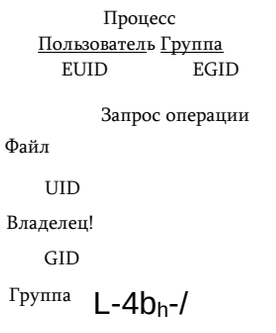


Рис. 16.3. Проверка прав доступа в Unix

Введение эффективных идентификаторов позволяет процессу выступать в некоторых случаях от имени пользователя и группы, отличных от тех, которые ему достались «при рождении». В исходном состоянии эффективные идентификаторы совпадают с реальными.

Случаи, когда процесс выполняет системный вызов exec для запуска приложения, хранящегося в некотором файле, в Unix связаны со сменой процессом исполняемого кода. В рамках данного процесса начинает выполняться новый код, и если в характеристиках безопасности этого файла указаны признаки разрешения смены идентификаторов пользователя и группы, то происходит смена эффективных идентификаторов процесса. Файл имеет два признака разрешения смены идентификатора — Set User ID on execution (SUID) и Set Group ID on execution (SGID), которые разрешают смену идентификаторов пользователя и группы при выполнении данного файла.

Механизм эффективных идентификаторов позволяет пользователю получать некоторые виды доступа, которые ему явно не разрешены, но только с помощью вполне ограниченного набора приложений, хранящихся в файлах с установленными признаками смены идентификаторов. Пример такой ситуации приведен ниже.

```
## ## user MACHINE=COMMANDS
## ## The COMMANDS section may have other options added to it.
## ## Allow root to run any commands anywhere root
ALL=(ALL) ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
#   */„sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING,
PROCESSES, # LOCATE, DRIVERS
## Allows people in group wheel to run all commands
#   '/.wheel ALL=(ALL) ALL
## Same thing without a password
#   '/.wheel ALL=(ALL) NOPASSWD: ALL
## Allows members of the users group to mount and unmount the ## cdrom as root
#   '/.users ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom
```

```
## Allows members of the users group to shutdown this system
#   У,users localhost=/sbin/shutdown -h now
## Read drop-in files from /etc/sudoers.d (the # here does not mean
a comment)
#   includedir /etc/sudoers.d
```

Первоначально процесс ***A*** имел эффективные идентификаторы пользователя и группы (12 и 23 соответственно), совпадающие с реальными. На каком-то этапе работы процесс запросил выполнение приложения из файла b.exe. Процесс может выполнить файл b.exe, хотя его эффективные идентификаторы не совпадают с идентификатором владельца и группы файла, так как выполнение разрешено всем пользователям.

Файл b.exe имеет установленные признаки смены идентификаторов SUID и SGID, поэтому одновременно со сменой кода процесс меняет значения эффективных идентификаторов (35 и 47). Вследствие этого при последующей попытке записать данные в файл fl.doc процессу ***A*** это удается, так как его новый эффективный идентификатор группы совпадает с идентификатором группы файла fl.doc. Без смены идентификаторов эта операция для процесса ***A*** была бы запрещена.

Использование модели файла как универсальной модели разделяемого ресурса позволяет в Unix применять одни и те же механизмы для контроля доступа к файлам, каталогам, принтерам, терминалам и разделяемым сегментам памяти.

Суперпользователь root

Особое внимание должно уделяться использованию учетной записи root. Из-за того, что root может выполнять в Unix любые операции без каких-либо ограничений, злоумышленник, завладевший паролем суперпользователя, может нанести очень серьезный ущерб системе, читая любые данные, изменяя ее конфигурацию или разрушая ее элементы. Кроме того, даже если вход с именем root сделал легальный администратор системы, то далеко не все операции, которые ему необходимо выполнять в процессе работы, требуют полномочий суперпользователя. А вот последствия ошибок администратора будут намного более серьезными, если во время всего сеанса своей работы он будет обладать полномочиями суперпользователя.

Поэтому считается нормальной практикой свести до минимума работу пользователей Unix от имени root, т. е. следовать принципу минимальных прав.

Для реализации концепции минимального использования прав суперпользователя в системе Unix введены две команды: su и sudo. Обе они предполагают, что пользователи, которым нужно и разрешено выполнять некоторые административные привилегированные действия, нходят в систему не под именем root.

Команда su -username выполяет замену учетной записи пользователя на учетную запись, указанную в параметре username. Для)того ОС

запрашивает пароль пользователя username, если он введен правильно, то далее пользователь работает от имени username, как если бы он осуществил логический вход под этим именем. Хотя команда su может заменить текущего пользователя на любого другого зарегистрированного пользователя, чаще всего она используется для временного получения статуса суперпользователя root, именно он подразумевается по умолчанию в том случае, когда команда su вводится без параметра username. Для того чтобы вернуться к своей начальной учетной записи, пользователь должен выполнить команду exit. Применение команды su не так уж сильно повышает защищенность системы, так как пароль суперпользователя все равно вводится, а пользователь, ставший root, может находится в этом статусе произвольное время, просто забыв вернуться к нормальному статусу, а значит, вероятность совершения им ошибок с тяжелыми последствиями для системы остается высокой.

Команда sudo гораздо надежнее защищает систему от случайных ошибок администрирования или перехвата пароля root злоумышленником. Эта команда избирательно позволяет некоторым пользователям или группам пользователей выполнять некоторые привилегированные команды.

Например, для того чтобы создать нового пользователя с помощью команды /usr/sbin/useradd katerina, необходимо быть суперпользователем, так как права доступа этого файла выглядят следующим образом:

```
> Is -1 /usr/sbin/useradd
> -rwxr-x-------  1 root root
```

Однако при определенной конфигурации системы и обычный пользователь может создать нового пользователя с помощью sudo, например, введя команды:

```
> sudo /usr/sbin/useradd katerina
> password:   ******
```

Это приведет к созданию пользователя katerina, если пользователь успешно ввел свой ***собственный*** (а не суперпользователя) пароль.

Гибкий контроль доступа с помощью команды sudo выполняется записями в файле конфигурации /etc/sudoers, который может редактировать только root с помощью утилиты visudo.

Обычно после установки ОС файл sudoers не разрешает обычным пользователям выполнять привилегированные команды, например в ОС CentOS 6.4 его начальное состояние выглядит так:

```
##
## user MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere root
ALL=(ALL) ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
#    '/.sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING,
PROCESSES, # LOCATE, DRIVERS
## Allows people in group wheel to run all commands
#    '/.wheel ALL=(ALL) ALL
## Same thing without a password
#    '/.wheel ALL= (ALL) NOPASSWD: ALL
## Allows members of the users group to mount and unmount the ## cdrom as root
#    '/.users ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom ## Allows
members of the users group to shutdown this system
#    "/.users localhost=/sbin/shutdown -h now
## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers .d
```

Как видно из первого комментария, общий формат записей имеет три параметра USER, MACHINE и COMMAND, что означает, что пользователю USER можно выполнять с помощью sudo команды COMMAND на машине MACHINE.

Единственной работающей строкой приведенной выше конфигурации является

```
root ALL=(ALL) ALL
```

которая говорит о том, что суперпользователю root можно выполнять с помощью sudo любые команды на любых машинах от имени любого пользователя.

Закрытые комментариями команды дают пример управления правами доступа. Например, строка

```
"/„users ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom
```

показывает, как можно членам некоторой группы users (имя произвольное) дать возможность монтировать и демонтировать CD ROM.

Во многих версиях Unix существует стандартная (встроенная) группа wheel, которая должна объединять администраторов системы.

Дне записи из файла sudoers разрешают членам этой группы выпол- п пь любые команды на любых машинах без необходимости задавать ной пароль:

```
*/„wheel ALL=(ALL) ALL
"/„wheel ALL=(ALL) NOPASSWD: ALL
```

Если мы хотим дать обычному пользователю victoro право созда- и.пь новых пользователей, то для этого достаточно добавить к файлу Midoers такую строку:

```
victoro ALL=/usr/sbin/useradd
```

Контроль доступа в ОС семейства Windows

Система управления доступом в ОС семейства Windows (Windows ■I', Windows Server 2003, Windows Vista, Windows 7, Windows Server 2008, Windows 8) отличается высокой степенью гибкости, которая достигается за счет большого разнообразия субъектов и объектов доступа, а также детализации операций доступа.

Для разделяемых ресурсов в ОС семейства Windows применяет- < я общая модель объекта, который содержит такие характеристики пезопасности, как набор допустимых операций, идентификатор владельца, список управления доступом. Объекты создаются для любых ресурсов в том случае, когда они являются или становятся разделяемыми — файлов, каталогов, устройств, секций памяти, процессов. Характеристики объектов делятся на две части — общую часть, сос- tdB которой не зависит от типа объекта, и индивидуальную, определяемую типом объекта.

Все объекты хранятся в древовидных иерархических структурах, >лементами которых являются ***объекты-ветви*** (каталоги) и ***объекты-листья*** (файлы). Для объектов файловой системы такая схема отношений является прямым отражением иерархии каталогов и файлов. Для объектов других типов иерархическая схема отношений имеет свое содержание, например для процессов она отражает святи «родитель-потомок», а для устройств отражает принадлежность к определенному типу устройств и связи устройства с другими устройс- 1вами, например SCSI-контроллера с дисками.

Проверка прав доступа для объектов любого типа выполняется централизованно с помощью ***монитора безопасности (Security Reference Monitor),*** работающего в привилегированном режиме. Централизация функций контроля доступа повышает надежность средств кащиты информации операционной системы по сравнению с распределенной реализацией, когда в различных модулях ОС имеются свои

процедуры проверки прав доступа и вероятность ошибки программиста от этого возрастает.

Для системы безопасности ОС семейства Windows характерно наличие большого количества различных предопределенных (встроенных) субъектов доступа — как отдельных пользователей, так групп. Так, в системе всегда имеются пользователи ***Administrator, System u Guest,*** а также группы ***Users, Administrators, Account Operators, Server Operators, Everyone*** и другие. Смысл этих встроенных пользователей и групп состоит в том, что они наделены некоторыми правами, облегчая администратору работу по созданию эффективной системы разграничения доступа. При добавлении нового пользователя администратору остается только решить, к какой группе или группам отнести этого пользователя. Конечно, администратор может создавать новые группы, а также добавлять права к встроенным группам для реализации собственной политики безопасности, но во многих случаях встроенных групп оказывается вполне достаточно.

ОС семейства Windows поддерживает три класса ***операций доступа,*** которые отличаются типом субъектов и объектов, участвующих в этих операциях:

- разрешения;
- права;
- возможности пользователей.

Разрешения (***permissions***) — это множество операций, которые могут быть определены для субъектов всех типов по отношению к объектам любого типа: файлам, каталогам, принтерам, секциям памяти и т. д. Разрешения по своему назначению соответствуют правам доступа к файлам и каталогам в ОС Unix.

Права ***(user rights***) определяются для субъектов типа группа на выполнение некоторых системных операций: установку системного времени, архивирование файлов, выключение компьютера и т. п. В этих операциях участвует особый объект доступа — операционная система в целом. В основном именно права, а не разрешения отличают одну встроенную группу пользователей от другой. Некоторые права у встроенной группы является также встроенными — их у данной группы нельзя удалить. Остальные права встроенной группы можно удалять (или добавлять из общего списка прав).

Возможности пользователей ***(user abilities)*** определяются для отдельных пользователей на выполнение действий, связанных с формированием их операционной среды, например, изменение состава главного меню программ, возможность пользоваться пунктом меню Run (Выполнить) и т. п. За счет уменьшения набора возможностей (доступных пользователю по умолчанию) администратор может «заставить» пользователя работать с той операционной средой, которую администратор считает наиболее подходящей и ограждающей поль- ювателя от возможных ошибок.

Права и разрешения, данные группе, автоматически предостав- мнются ее членам, позволяя администратору рассматривать большое количество пользователей как единицу учетной информации и минимизировать свои

действия.

Разрешения доступа процесса к объекту проверяются в ОС семейства Windows следующим образом.

При входе пользователя в систему для него создается так назы- иаемый *токен доступа* ***(access token),*** включающий:

- идентификатор пользователя;
- идентификаторы всех групп, в которые входит пользователь;
- список управления доступом (ACL) по умолчанию, который состоит из разрешений и применяется к создаваемым процессом объектам;
- список прав пользователя на выполнение системных действий. Все

объекты, включая файлы, потоки, события, даже токены доступа, когда они создаются, снабжаются дескриптором безопасно- | I и *Дескриптор безопасности* содержит *список управления доступом* ***(Access Control List,*** *ACL).* Владелец объекта, обычно пользователь, который его создал, обладает правом избирательного управления доступом к объекту и может изменять ACL объекта, чтобы позволить или не позволить другим осуществлять доступ к объекту. Встроенный пользователь Administrator в ОС семейства Windows в отличие от ' уперпользователя в ОС Unix может не иметь некоторых разрешений на доступ к объекту. Для реализации этой возможности идентификаторы администратора и группы администраторов могут входить в ACL, как и идентификаторы рядовых пользователей. Однако администратор все же имеет возможность выполнять любые операции с любыми объектами, так как он всегда может стать владельцем объекта, а затем уже как владелец получить полный набор разрешений.

Чтобы компенсировать такое «ущемление» прав владельца, операционная система ограничивает администратора тем, что он не может вернуть владение предыдущему владельцу объекта. В результате пользователь-владелец всегда может узнать о том, что с его файлом или принтером работал администратор.

При запросе процессом некоторой операции доступа к объек- ty в ОС семейства Windows управление всегда передается монитору безопасности, который сравнивает идентификаторы пользователя и ipynn пользователей из токена доступа с идентификаторами, хранящимися в элементах ACL объекта. В отличие от Unix, здесь в элемен- I.lx ACL могут существовать как списки разрешенных, так и списки мпрещенных для пользователя операций.

Система безопасности могла бы осуществлять проверку разрешений каждый раз, когда процесс использует объект. Но список ACL состоит из многих элементов, процесс в течение своего существования может иметь доступ ко многим объектам, и количество активных процессов в каждый момент времени также велико. Поэтому проверка выполняется только при каждом открытии, а не при каждом использовании объекта.

Для смены в некоторых ситуациях процессом своих идентификаторов в ОС семейства Windows используется механизм ***олицетворения*** (impersonation). В этих ОС существуют простые субъекты и субъекты-серверы. *Простой субъект* — это процесс, которому не разрешается смена токена доступа и, соответственно, смена идентификаторов. *Субъект-сервер* — это процесс, который работает в качестве сервера и обслуживает процессы своих клиентов (например, процесс файлового сервера). Поэтому такому процессу разрешается получить токен доступа у процесса-клиента, запросившего у сервера выполнение некоторого действия, и использовать его при доступе к объектам.

В ОС семейства Windows однозначно определены правила, по которым вновь создаваемому объекту назначается список ACL. Если вызывающий код во время создания объекта явно задает все права доступа к вновь создаваемому объекту, то система безопасности приписывает объекту этот список ACL.

Если же вызывающий код не снабжает объект списком ACL, а объект имеет имя, то применяется *принцип наследования* разрешений. Система безопасности просматривает ACL того каталога объектов, в котором хранится имя нового объекта. Некоторые из входов ACE каталога объектов могут быть помечены как наследуемые. Это означает, что они могут быть приписаны новым объектам, создаваемым в этом каталоге.

В случае, когда процесс не задал явно список ACL для создаваемого объекта и объект-каталог не имеет наследуемых элементов ACE, используется список ACL по умолчанию из токена доступа процесса.

Наследование разрешений чаще всего применяется при создании нового объекта. Особенно эффективно наследование разрешений при создании файлов, так как эта операция самая распространенная в системе.

Разрешения на доступ к каталогам и файлам

В ОС семейства Windows администратор может управлять доступом пользователей к каталогам и файлам только в разделах диска, в которых установлена файловая система NTFS. Разделы FAT не

Таблица 16.1

Индивидуальные разрешения для каталогов и файлов

Разрешение	Для каталога	Для файла
Read (R)	Чтение имен файлов и каталогов, входящих в данный каталог, а также атрибутов и владельца каталога	Чтение данных, атрибутов, имени владельца и разрешений файла
Write (W)	Добавление файлов и каталогов, изменение атрибутов каталога, чтение владельца и разрешений каталога	Чтение владельца и разрешений файла, изменение атрибутов файла, изменение и добавление данных файла
Execute (X)	Чтение атрибутов каталога, выполнение изменений в каталогах, входящих в данный каталог, чтение имени владельца и разрешений каталога	Чтение атрибутов файла, имени владельца и разрешений. Выполнение файла, если он хранит код программы
Delete (D)	Удаление каталога	Удаление файла
Change Permission (P)	Изменение разрешений каталога	Изменение разрешений файла
Take Ownership (0)	Вступление во владение каталогом	Вступление во владение файлом

поддерживаются средствами защиты, так как в FAT у файлов и каталогов отсутствуют атрибуты для хранения списков управления доступом. Доступ к каталогам и файлам контролируется за счет установки соответствующих разрешений.

В ОС семейства Windows предусмотрено два типа разрешений:

- *индивидуальные разрешения* относятся к элементарным операциям с каталогами и файлами;
- *стандартные разрешения* являются объединением нескольких индивидуальных разрешений.

В табл. 16.1 показано шесть индивидуальных разрешений (элементарных операций), смысл которых различается для каталогов и файлов.

Для файлов определено четыре стандартных разрешения: ***No Access, Read, Change*** и ***Full Control,*** которые объединяют индивидуальные разрешения, перечисленные в табл. 16.2.

Таблица 16.2

Стандартные разрешения, объединяющие индивидуальные разрешения

Стандартное разрешение	Индивидуальные разрешения
No Access	Ни одного
Read	RX
Change	RWXD
Full Control	Все

Разрешение ***Full Control*** отличается от ***Change*** тем, что дает право на изменение разрешений (***Change Permission)*** и вступление во владение файлом (***Take Ownership).***

Для каталогов определено семь стандартных разрешений: ***No Access, List, Read, Add, Add&Read, Changes Full Control.*** В табл. 16.3

Таблица 16.3

Соответствие стандартных разрешений индивидуальным разрешениям при наследовании

Стандартные разрешения	Индивидуальные разрешения для каталога	Индивидуальные разрешения для файлов каталога при наследовании
No Access	Ни одного	Ни одного
List	RX	Не определены
Read	RX	RX
Add	WX	Не определены
Add&Read	RWX	RX
Change	RWXD	RWXD
Full Control	Все	Все

показано соответствие стандартных разрешений индивидуальным разрешениям для каталогов, а также то, каким образом эти стандартные разрешения преобразуются в индивидуальные разрешения для файлов, входящих в каталог, в том случае, если файлы наследуют разрешения каталога.

При создании файла он наследует разрешения от каталога указанным способом только в случае, если у каталога установлен признак наследования его разрешений. Стандартная оболочка ОС семейства Windows — Проводник Windows (Windows Explorer) — не позволяет установить такой признак для каждого разрешения отдельно (т. е. задать маску наследования), управляя наследованием по принципу «все или ничего».

Существует ряд правил, которые определяют действие разрешений.

- Пользователи не могут работать с каталогом или файлом, если они не имеют явного разрешения на это, или же они не относятся к группе, которая имеет соответствующее разрешение.
- Разрешения имеют накопительный эффект за исключением разрешения ***No Access,*** которое отменяет все остальные имеющиеся разрешения. Например, если группа ***Engineering*** имеет разрешение ***Change для*** какого-то файла, а группа ***Finance*** имеет для этого файла только разрешение ***Read*** и Петров является членом обеих групп, то у Петрова будет разрешение ***Change.*** Однако если разрешение для группы ***Finance*** изменится на ***No Access,*** то Петров не сможет использовать этот файл несмотря на то, что он член группы, которая имеет доступ к файлу.

По умолчанию в окнах Проводника Windows находят свое отражение стандартные права, а переход к отражению индивидуальных прав происходит только при выполнении некоторых действий. Это стимулирует администратора и пользователей к получению тех наборов прав, которые разработчики ОС посчитали наиболее удобными.

Встроенные группы пользователей и их права

Гибкость системы безопасности ОС семейства Windows во многом определяется наличием в ней достаточно широкого набора прав групп пользователей на выполнение системных действий. Для иллюстрации этого утверждения в табл. 16.4 и 16.5 приводятся списки изменяемых и встроенных прав для встроенных групп ОС семейства Windows NT.

При создании новых групп администратор может наделить их любым изменяемым правом, но распоряжаться встроенными правами он не может — они являются неотъемлемыми атрибутами встроенных и только встроенных групп.

Система Kerberos

Kerberos — это сетевая служба, предназначенная для централизованного решения задач аутентификации в крупных сетях. Kerberos реализует процедуру единого логического входа SSO, которую мы рассмотрели в главе 5, в пределах домена, где клиенты и серверы поддерживают этот протокол.

Система централизованной аутентификации тесно связана с системой централизованного управления доступом, так как последняя должна использовать результаты аутентификации каждый раз, когда вычислительный процесс, представляющий пользователя, пытается получить доступ к ресурсу компьютера, входящего в некоторый домен, поэтому результат аутентификации должен быть оформлен в виде, пригодном для безопасной передачи по сети.

Система Kerberos может работать в среде многих популярных ОС, например в ОС семейства Windows система Kerberos встроена как основной компонент безопасности. Существуют реализации Kerberos для семейства Unix, включая Red Hat Linux, Fedora, Centos, Ubuntu, и для Mac OS X. Первая версия Kerberos была разработана для проекта Athena в Массачусетском технологическом институте. Текущей версией является версия 5, которая стандартизована IETF в RFC 4120.

В основе функционирования этой достаточно громоздкой системы лежит несколько простых принципов:

- в сетях, использующих систему безопасности Kerberos, все процедуры аутентификации между клиентами и серверами сети выполняются через посредника, которому доверяют обе стороны процесса аутентификации, причем таким авторитетным арбитром является сама система Kerberos;
- в системе Kerberos клиент должен доказывать свою аутентичность для доступа к каждой службе, услуги которой он запрашивает;

Таблица 16.4
Изменяемые права встроенных групп

Право	Administrators	Server Operators	Account Operators	Print Operators	Backup Operators	Everyone	Users	Guests
Log on locally (локальный логический вход)	Есть	Есть	Есть	Есть	Есть	Нет	Нет	Нет
Access this computer from network (доступ к данному компьютеру через сеть)	Есть	Нет	Нет	Нет	Нет	Есть	Нет	Нет
Take ownership of files (установление прав собственности на файлы)	Есть	Нет	Нет	Нет	Нет	Нет	Нет	Нет
Manage auditing and security log (управление аудитом и учетом событий, связанных с безопасностью)	Есть	Нет	Нет	Нет	Нет	Нет	Нет	Нет
Change the system time (изменение системного времени)	Есть	Есть	Нет	Нет	Нет	Нет	Нет	Нет
Shutdown the system (останов системы)	Есть	Есть	Нет	Нет	Есть	Нет	Нет	Нет
Force shutdown from remote system (инициирование останова с удаленной системы)	Есть	Есть	Нет	Нет	Нет	Нет	Нет	Нет
Backup files and directories (резервное копирование файлов и каталогов)	Есть	Есть	Есть	Есть	Есть	Нет	Нет	Нет
Restore files and directories (восстановление файлов и каталогов со стримера)	Есть	Есть	Нет	Нет	Есть	Нет	Нет	Нет
Load and unload device drivers (загрузка и выгрузка драйверов устройств)	Есть	Нет	Нет	Нет	Нет	Нет	Нет	Нет
Add workstation to domain (добавление рабочих станций к домену)	Есть	Нет	Нет	Нет	Нет	Нет	Нет	Нет

Таблица 16.5
Встроенные права встроенных групп

Встроенное право	Administrators	Server Operators	Account Operators	Print Operators	Backup Operators	Everyone	Users	Guests
Create and manage user accounts (создание и управление пользовательской учетной информацией)	Есть	Нет	Есть	Нет	Нет	Нет	Нет	Нет
Create and manage global groups (создание и управление глобальными группами)	Есть	Нет	Есть	Нет	Нет	Нет	Нет	Нет
Create and manage local groups (создание и управление локальными группами)	Есть	Нет	Есть	Нет	Нет	Нет	Нет	Нет
Assign user rights (назначение прав для пользователей)	Есть	Нет	Нет	Нет	Нет	Нет	Нет	Нет
Manage auditing of system events (управление аудитом системных событий)	Есть	Нет	Нет	Нет	Нет	Нет	Нет	Нет
Lock the server (блокирование сервера)	Есть	Есть	Нет	Нет	Нет	Есть	Нет	Нет
Override the lock of the server (преодоление блокировки сервера)	Есть	Есть	Нет	Нет	Нет	Нет	Нет	Нет
Format server's hard disk (форматирование жесткого диска сервера)	Есть	Есть	Нет	Нет	Нет	Нет	Нет	Нет
Create common groups (создание общих групп)	Есть	Есть	Нет	Нет	Нет	Нет	Нет	Нет
Keep local profile (хранение локального профиля)	Есть	Есть	Есть	Есть	Есть	Нет	Нет	Нет
Share and stop sharing directories (разделение и прекращение разделения каталогов)	Есть	Есть	Нет	Нет	Нет	Нет	Нет	Нет
Share and stop sharing printers (разделение и прекращение разделения принтеров)	Есть	Есть	Нет	Есть	Нет	Нет	Нет	Нет

Рис. 16.5. Три этапа работы системы Kerberos: 1 — аутентификация пользователя, получение доступа в сеть и разрешение на дальнейшую авторизацию; 2 — авторизация Kerberos-сервера и получение доступа к ресурсному серверу; 3 — авторизация у ресурсного сервера и получение доступа к сетевому серверу (файлам, приложениям, устройствам)

- все обмены данными в сети выполняются в защищенном виде с использованием симметричного алгоритма шифрования AES (или DES в ранних реализациях Kerberos).

Сетевая служба Kerberos построена в архитектуре клиент-сервер, что позволяет ей работать в самых сложных сетях. Kerberos-клиент устанавливается на всех компьютерах сети, которые могут обратиться к какой-либо сетевой службе. В таких случаях Kerberos-клиент от лица пользователя передает запрос на Kerberos-сервер и поддерживает с ним диалог, необходимый для выполнения функций системы Kerberos.

Итак, в системе Kerberos имеются следующие участники: ***Kerberos-сервер, Kerberos-клиенты, ресурсные серверы*** (рис. 16.5). Kerberos-клиенты пытаются получить доступ к сетевым ресурсам — файлам, приложениям, принтеру и т. д., находящимся на ресурсных серверах. Этот доступ может быть предоставлен, во-первых, только легальным пользователям, а во-вторых, при наличии у них достаточных полномочий, определяемых службами авторизации соответствующих ресурсных серверов — файловым сервером, сервером приложений, сервером печати. Однако в системе Kerberos ресурсным серверам запрещается «напрямую» принимать запросы от клиентов, им разрешается начинать рассмотрение запроса клиента только тогда, когда на это поступает разрешение от Kerberos-сервера. Таким образом, путь клиента к ресурсу в системе Kerberos состоит из трех этапов:

- определение легальности клиента, логический вход в сеть, получение разрешения на продолжение процесса получения доступа к ресурсу;
- получение разрешения на обращение к ресурсному серверу;
- получение разрешения на доступ к ресурсу.

Для решения первой и второй задач клиент обращается к Kerberos-серверу. Каждая из этих двух задач решается отдельным сервером, входящим в состав Kerberos-сервера. Выполнение первичной аутентификаций и выдача разрешения на продолжение процесса получения доступа к ресурсу осуществляются так называемым ***сервером*** *аутентификации Kersberos* (***Authentication Server,*** AS). Этот сервер хранит в своей базе данных информацию об идентификаторах и паролях пользователей. Пароли пользователей, а точнее — хеш-функции от паролей, являются секретными ключами пользователей.

Вторую задачу, связанную с получением разрешения на обращение к ресурсному серверу, решает другая часть Kerberos-сервера — *сервер квитанций Kerberos* (***Ticket-Granting Server, TGS***). Сервер квитанций для легальных клиентов выполняет дополнительную проверку и дает клиенту разрешение на доступ к нужному ему ресурсному серверу, для чего наделяет его электронной формой-квитанцией. Для выполнения своих функций сервер квитанций использует копии секретных ключей всех ресурсных серверов, которые хранятся у него в базе данных. Помимо этих ключей TGS-сервер имеет еще один секретный ключ, общий с AS-сервером.

Третья задача — получение разрешения на доступ непосредственно к ресурсу — решается на уровне ресурсного сервера собственными средствами, ***не относящимися*** непосредственно к системе Kerberos, но способными взаимодействовать с ней.

Секретные ключи пользователей и ресурсных серверов образуют базу данных ключей Kerberos-сервера. Собственно, обладание секретным ключом и является условием аутентификации пользователя или ресурсного сервера. Помимо секретных ключей пользователей и ресурсных серверов в Kerberos также используются секретные ключи сеансов аутентификации, которые распределяет Kerberos-сервер. Из-за этого обстоятельства Kerberos-сервер также называется *Kerberos Key Distribution Centre* или *Kerberos KDC.* Секретные ключи пользователей и ресурсных серверов также называют мастер-ключами, так как они являются постоянными ключами, аутентифицирующими субъект, в отличие от ключей сеансов, которые имеют непродолжительный срок действия.

Введение центра аутентификации существенно повышает масштабируемость системы аутентификации на основе симметричного

шифрования по сравнению с децентрализованной системой. Действительно, если на предприятии имеется N пользователей и M ресурсных серверов, которым нужна взаимная аутентификация (мы предполагаем, что пользователям взаимная аутентификация не нужна), то при децентрализованной аутентификации необходимо NM ключей, что для предприятия с 1000 сотрудников и 50 ресурсными серверами дает 50000 ключей. При централизованной систем аутентификации необходимо иметь только N + M ключей, что равно 1050 ключам для нашего примера — т. е. почти в 50 раз меньше.

Необходимо подчеркнуть, что Kerberos обеспечивает защищенную аутентификацию сторон только в начальный момент сессии обмена данными между ними. После этого защита данных — их конфиденциальность, аутентичность и целостность — должны обеспечиваться средствами ресурсного сервера и клиента, если это необходимо.

При описании протоколов взаимодействия Kerberos-клиента и Kerberos-сервера, а также Kerberos-клиента и ресурсного сервера использован термин «квитанция» (ticket), означающий в данном случае электронную форму, выдаваемую Kerberos-сервером клиенту, которая играет роль некоего удостоверения личности и разрешения на доступ к ресурсу.

Первичная аутентификация

Процесс доступа пользователя к ресурсам включает две процедуры: во-первых, пользователь должен доказать свою легальность (аутентификация), во-вторых, он должен получить разрешение на выполнение определенных операций с определенным ресурсом (авторизация). В системе Kerberos пользователь один раз аутентифицируется во время логического входа в сеть, а затем проходит процедуры аутентификации и авторизации всякий раз, когда ему требуется доступ к новому ресурсному серверу.

Выполняя логический вход в сеть, пользователь, а точнее Kerbe- ros-клиент, установленный на его компьютере, посылает серверу аутентификации AS идентификатор пользователя Ю (рис. 16.б).

Вначале сервер аутентификации проверяет в базе данных, имеется ли в ней запись о пользователе с таким идентификатором. Затем, если такая запись существует, он извлекает из нее пароль пользователя ***p.*** Данный пароль потребуется для шифрования всей информации, которую направит сервер аутентификации Kerberos-клиенту в качестве ответа. А ответ состоит из квитанции TTGS на доступ к серверу квитанций Kerberos и ключа сеанса ***Ks***. Под сеансом здесь понимается все время работы пользователя от момента логического входа в сеть до момента логического выхода. Ключ сеанса потребуется для шифрования в процедурах аутентификации в течение всего пользовательского сеанса. Квитанция шифруется с помощью секретного мастер- ключа ***K,*** который разделяют серверы аутентификации и квитанций Kerberos KDC. Все вместе — зашифрованная квитанция и ключ сеанса — еще раз шифруются с помощью хеша пользовательского пароля ***p.*** Таким образом, квитанция

шифруется дважды ключом ***K*** и паролем ***p.*** В приведенных обозначениях сообщение-ответ, которое сервер аутентификации посылает клиенту, выглядит так: ***{{TTQS}K,KS}P***

После того как такое ответное сообщение поступает на клиентскую машину, клиентская программа Kerberos просит пользователя ввести свой пароль. Когда пользователь вводит пароль, то Kerberos- клиент пробует с помощью хеша пароля расшифровать поступившее

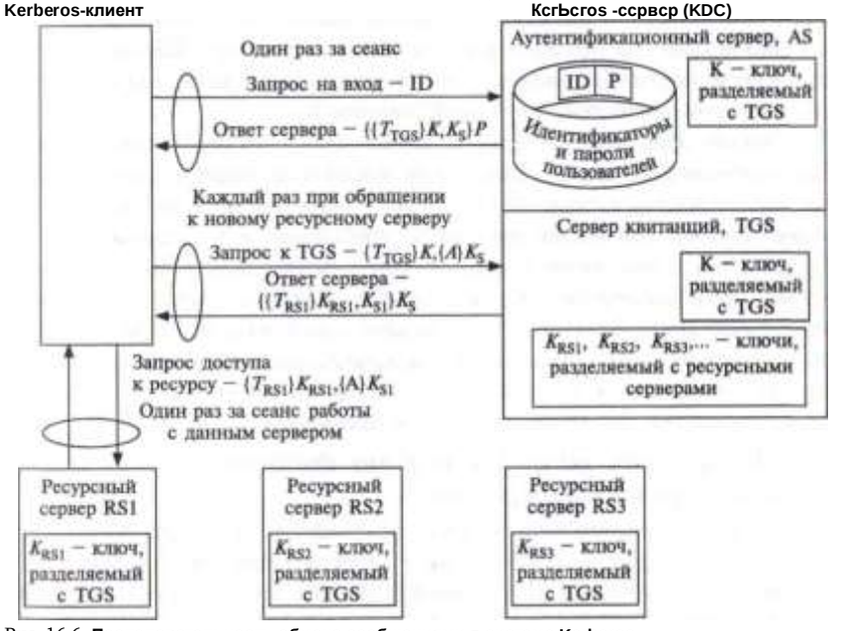


Рис. 16.6. **Последовательность обмена сообщениями в системе Kerberos**

сообщение. Если пароль верен, то из сообщения извлекаются квитанция на доступ к серверу квитанций {TTGS}-^ (в зашифрованном виде) и ключ сеанса ***Ks*** (в открытом виде). Успешное дешифрирование сообщения означает успешную аутентификацию. Заметим, что сервер аутентификации AS аутентифицирует пользователя без передачи пароля по сети. Нужно отметить, что успешность дешифрирования будет проверена позже, когда пользователь попытается использовать полученную квитанцию и ключ сеанса при обращении к серверу квитанций TGS.

Квитанция TTGS на доступ к серверу квитанций TGS является удостоверением легальности пользователя и разрешением ему продолжать процесс получения доступа к ресурсу. Эта квитанция содержит:

- идентификатор пользователя;
- идентификатор сервера квитанций, на доступ к которому получена квитанция;
- отметку о текущем времени;

- период времени, в течение которого может продолжаться сеанс;
- копию ключа сеанса ***Ks-***

Как уже было сказано, клиент обладает квитанцией в зашифрованном виде. Шифрование повышает уверенность в том, что никто,

даже сам клиент — обладатель данной квитанции — не сможет квитанцию подделать, подменить или изменить. Только TGS-сервер, получив от клиента квитанцию, сможет ее расшифровать, так как в его распоряжении имеется ключ шифрования ***K.***

Время действия квитанции ограничено длительностью сеанса. Разрешенная длительность сеанса пользователя, содержащаяся в квитанции на доступ к серверу квитанций, задается администратором и может изменяться в зависимости от требований к защищенности сети. В сетях с жесткими требованиями к безопасности время сеанса может быть ограничено 30 минутами, в других условиях это время может составить 8 часов. Информация, содержащаяся в квитанции, определяет ее срок годности. Предоставление квитанции на вполне определенное время защищает ее от неавторизованного пользователя, который мог бы ее перехватить и применить в будущем.

Получение разрешения на доступ к ресурсному серверу

Итак, следующим этапом для пользователя является получение разрешения на доступ к ресурсному серверу (например, к файловому серверу или серверу приложений). Но для этого надо обратиться к TGS-серверу, который выдает такие разрешения (квитанции). Чтобы получить доступ к серверу квитанций, пользователь уже обзавелся квитанцией (TTGS}-^. выданной ему AS-сервером. Несмотря на защиту паролем и шифрование, пользователю, помимо квитанции, нужно кое-что еще, чтобы доказать серверу квитанций, что он имеет право на доступ к ресурсам сети.

Как уже упоминалось, первое сообщение от сервера аутентификации содержит не только квитанцию, но и секретный ключ сеанса ***K$,*** который разделяется с сервером квитанций (TGS). Клиент использует этот ключ для шифрования еще одной электронной формы, называемой ***аутентификатором {A}K$-*** Аутентификатор ***A*** содержит идентификатор и сетевой адрес пользователя, а также собственную временную отметку. В отличие от квитанции ***{TTGSJK,*** которая в течение сеанса используется многократно, аутентификатор предназначен для одноразового использования и имеет очень короткое время жизни — обычно несколько минут. Kerberos-клиент посылает серверу квитанций сообщение-запрос, содержащее квитанцию и аутентификатор: {TTGS}^. {rll-A's-

Сервер квитанций расшифровывает квитанцию имеющимся у него ключом ***K,*** проверяет, не истек ли срок действия квитанции, и извлекает из нее идентификатор пользователя.

Затем TGS-сервер расшифровывает аутентификатор, применяя ключ сеанса пользователя ***Ks,*** который он извлек из квитанции. Сервер квитанций сравнивает идентификатор пользователя и его сетевой адрес с аналогичными параметрами в квитанции и сообщении. Если они совпадают, сервер квитанций удостоверяется, что данная квитанция действительно представлена ее законным владельцем. Применение ключа

сеанса из зашифрованной квитанции говорит серверу TGS, что квитанция действительно была выдана сервером AS, так как квитанция была расшифрована мастер-ключом, который известен только паре AS-TGS.

Заметим, что простое обладание квитанцией на получение доступа к серверу квитанций не доказывает идентичности пользователя. Так как аутентификатор действителен только в течение короткого промежутка времени, то маловероятно украсть одновременно и квитанцию, и аутентификатор и использовать их в течение этого времени. Каждый раз, когда пользователь обращается к серверу квитанций для получения новой квитанции на доступ к ресурсу, он посылает многоразовую квитанцию и новый аутентификатор.

Клиент обращается к серверу квитанций за разрешением на доступ к ресурсному серверу, который здесь обозначен как RS1. Сервер квитанций, удостоверившись в легальности запроса и личности пользователя, отсылает ему ответ, содержащий две электронных формы: многоразовую квитанцию на получение доступа к запрашиваемому ресурсному серверу TRSI и новый ключ сеанса ***K$1-***

Квитанция на получение доступа шифруется секретным ключом KRSI, общим только для сервера квитанций и того сервера, к которому предоставляется доступ, в данном случае — RS1. Сервер квитанций разделяет уникальные секретные ключи с каждым сервером сети. Эти ключи распределяются между серверами сети физическим способом или каким-либо иным секретным способом при установке системы Kerberos. Когда сервер квитанций передает квитанцию на доступ к какому-либо ресурсному серверу, то он шифрует ее, так что только этот сервер сможет расшифровать ее с помощью своего уникального ключа.

Новый ключ сеанса ***Ksi*** содержится не только в самом сообщении, посылаемом клиенту, но и внутри квитанции TRSI- Все сообщение шифруется старым ключом сеанса клиента ***Ks,*** так что его может прочитать только этот клиент. Используя введенные обозначения, ответ TGS-сервера клиенту можно представить в следующем виде: {{TRSI}ATRSI, ATsi}iGs-

Получение доступа к ресурсу

Когда клиент расшифровывает поступившее сообщение, то он отсылает серверу, к которому он хочет получить доступ, запрос, содержащий квитанцию на получение доступа и аутентификатор, зашифрованный новым ключом сеанса: ***{TrmRsi}KRsu*** {-AJJfsi-

Это сообщение обрабатывается аналогично тому, как обрабатывался запрос клиента TGS-сервером. Сначала расшифровывается квитанция ключом ***Krsi,*** затем извлекается ключ сеанса ***Ksi*** и расшифровывается аутентификатор. Далее сравниваются данные о пользователе, содержащиеся в квитанции и аутентификаторе. Если проверка проходит успешно, то доступ к сетевому ресурсу разрешается.

На этом этапе клиент тоже может захотеть проверить аутентичность сервера перед тем, как начать с ним работать. Взаимная процедура аутентификации предотвращает любую возможность попытки получения неавторизованным пользователем доступа к секретной информации от клиента путем подмены сервера.

Аутентификация ресурсного сервера в системе Kerberos выполняется в соответствии со следующей процедурой. Клиент обращается к серверу с предложением, чтобы тот прислал ему сообщение, в котором повторил временную отметку из аутентификатора клиента, увеличенную на 1. Кроме того, требуется, чтобы данное сообщение было зашифровано ключом сеанса ***Ksi***. Чтобы выполнить такой запрос клиента, сервер извлекает копию ключа сеанса из квитанции на доступ, использует этот ключ для расшифровки аутентификатора, наращивает значение временной отметки на 1, заново зашифровывает сообщение с помощью ключа сеанса и возвращает сообщение клиенту. Клиент расшифровывает это сообщение, чтобы получить увеличенную на единицу отметку времени.

При успешном завершении описанного процесса клиент и сервер удостоверяются в секретности своих транзакций. Кроме этого, они получают ключ сеанса, который могут использовать для шифрования будущих сообщений.

Достоинства и недостатки

Изучая довольно сложный механизм системы Kerberos, нельзя не задаться вопросом: какое влияние оказывают все эти многочисленные процедуры шифрования и обмена ключами на производительность сети, какую часть ресурсов сети они потребляют, и как это сказывается на ее пропускной способности?

Ответ весьма оптимистичный — если система Kerberos реализована и сконфигурирована правильно, ее работа незначительно сказывается на производительности сети. Так как квитанции используются многократно, сетевые ресурсы, затрачиваемые на запросы предоставления квитанций, невелики. Хотя передача квитанции при аутентификации логического входа несколько снижает пропускную способность, такой обмен требуется в любых других системах и при использовании любых методов аутентификации. Дополнительные же издержки незначительны. Опыт внедрения системы Kerberos показал, что время отклика при установленной системе Kerberos существенно не отличается от времени отклика без нее — даже в очень больших сетях с десятками тысяч узлов.

Такая эффективность делает систему Kerberos весьма перспективной.

Среди уязвимых мест системы Kerberos можно назвать централизованное хранение всех секретных ключей системы. Успешная атака на Kerberos-сервер, в котором сосредоточена вся информация, критическая для системы безопасности, приводит к краху информационной защиты всей сети. Поэтому репозиторий мастер-ключей должен быть хорошо защищен.

Для защиты секретных мастер-ключей в процессе первоначального создания (заведении новых учётных записей пользователей и ресурсных серверов в репозитории KDC) необходимо создавать защищенный канал между компьютером пользователя или ресурсным гервером и Kerberos KDC.

Возможен также полный отказ от использования паролей пользователя за счет использования цифровых сертификатов пользователя на первом этапе работы аутентификации, когда пользователь обращается к серверу AS квитанцией T_{TGS}- Это расширение протокола Kerberos описано в RFC 4556 Public Key Cryptography for Initial Authentication in Kerberos (PKINIT). Предполагается, что пользователь использует для логического входа смарт-карту, на которой хранится цифровой сертификат, выпущенный сертификационным центром, которому доверяет Kerberos AS. После того, как пользователь локально аутентифицирует себя как легальный владелец смарт-карты, его компьютер передаёт запрос на логический вход серверу Kerberos AS, в котором содержится цифровой сертификат пользователя с его открытым ключом, а также стандартный Kerberos-аутентификатор (идентификатор пользователя, его сетевой адрес и временная отметка), зашифрованный закрытым ключом пользователя, хранящимся на смарт-карте.

Сервер AS проверяет подлинность сертификата с помощью обращения к серверу сертификатов (корпоративному или публичному), а ытем расшифровывает аутентификатор с помощью открытого ключа пользователя и извлекает из него идентификатор пользователя. Если этот идентификатор имеется в базе идентификаторов пользователей (ервера AS, то сервер отвечает стандартным образом, т. е. посылает пользователю квитанцию 7TGS. зашифрованную ключом ***K,*** а также ключ сеанса ***Ks,*** однако ответ шифруется открытым ключом пользователя, а не его паролем. Пользователь расшифровывает ответ < помощью своего закрытого ключа, и на этом работа расширения PKINIT заканчивается.

Еще одной слабостью системы Kerberos является то, что исходные коды приложений, доступ к которым осуществляется через Kerberos, должны быть соответствующим образом модифицированы. Такая модификация называется «керберизацией» приложения. Некоторые поставщики продают «керберизованные» версии своих приложений. Однако если такой версии нет или нет исходного текста, то Kerberos не может обслуживать доступ к такому приложению.

В заключение хочется еще раз подчеркнуть, что стандартная версия Kerberos (соответствующая RFC 4210 и некоторым другим RFC, разработанным в IETF рабочей группой Kerberos) выполняет только аутентификацию пользователей. Авторизация оставлена ресурсным серверам, которые должны, например, использовать списки доступа при принятии решения о том, что может делать конкретный пользователь, представленный своим идентификатором в квитанции Kerberos, и что ему делать запрещено. Как мы увидим далее, компания Microsoft в своей реализации Kerberos расширила возможности этой системы и включила некоторую поддержку процедур авторизации, но эти расширения являются фирменными.

Справочная служба Active Directory компании Microsoft *Домены Active Directory*

Справочная служба *Active Directory* компании Microsoft построена на базе доменов. Домены являются основными элементами *логической* структуры Active Directory. В каждом из доменов реализуется централизованная справочная служба. Будучи связаны логически, службы доменов образуют распределенную справочную службу сети.

Справочная служба построена по архитектуре клиент-сервер. Клиентские компоненты, устанавливаемые на всех компьютерах сети, передают запросы к серверам справочной службы, которые в Active Directory называются *контроллерами домена (Domain Controllers, DC).* В каждом домене должен присутствовать хотя бы один контроллер домена. Он поддерживает доменную базу данных, т. е. БД о пользователях и ресурсах того домена, в котором установлен этот контроллер. Контроллер домена также занимается аутентификацией пользователей при их регистрации в сети.

Местоположение каждого контроллера домена администратор выбирает в ходе проектирования сети. Географическое разнесение серверов позволяет обеспечить приближенность информации к источникам

запросов. Любой компьютер в сети, будь то контроллер домена, рядовой сервер или рабочая станция, может быть присоединен только к одному домену.

Возможность декомпозиции сети на домены является важнейшим условием масштабируемости службы Active Directory. Решение о том, сколько и каких иметь доменов, принимается с учетом различных факторов.

Прежде всего, учитывается, что домен Active Directory является *единицей администрирования.* По умолчанию при создании домена права и разрешения, которыми наделяются администраторы и пользователи, распространяются только на этот домен. А для того чтобы пользователь (в том числе администратор) одного домена мог получить доступ к ресурсам другого домена, должны быть выполнены дополнительные административные действия. Таким образом, разделение сети на домены является наиболее надежным и наиболее естественным способом разделения зон ответственности между несколькими администраторами.

Для повышения надежности и производительности в домене устанавливается несколько равнозначных контроллеров домена, которые поддерживают несколько идентичных копий доменной базы данных. Процедура динамического копирования всех изменений, происходящих на каждом из контроллеров, на все оставшиеся контроллеры домена называется *репликацией.* Таким образом, домен является *единицей репликации,* т. е. определяет часть сети, в пределах которой происходит репликация базы данных домена.

Также следует принять во внимание, что каждый домен Active Directory в то же время является *доменом DNS-имен.*

Объекты

Информация в справочной базе данных Active Directory представлена в виде иерархически организованного набора объектов, которые соответствуют отдельным пользователям, группам пользователей, компьютерам, принтерам, разделяемым папкам, элементам структуры (доменам и организационным единицам35), конфигурационным параметрам и другим сетевым ресурсам. Объекты могут создаваться как «вручную» администратором, который использует для этой цели диалоговые средства Active Directory, так и автоматически службой Active Directory и другим программным обеспечением.

Объект уникально идентифицируется своим именем и представляет собой набор значений атрибутов, которые свойственны данному классу объектов.

Класс объектов — это формальное описание множества объектов,

35 Об организационных единицах см. далее в разделе «Иерархия организмционных единиц».

По желанию администратора в глобальный каталог могут быть добавлены дополнительные атрибуты.

имеющих сходную природу и вследствие этого характеризуемых одним и тем же набором обязательных и необязательных атрибутов.

Соотношение между классом объектов и объектом примерно такое же, как между переменной и ее значением. Атрибуты, определенные для класса объектов, принимают разные значения для разных объектов.

Из определения класса объектов следует, что объекты, содержащие информацию о компьютерах, относятся к одному классу объектов, а объекты, представляющие принтеры, — к другому. Рассмотрим, например, стандартный для Active Directory класс объектов user. Этот класс задает множество атрибутов, которыми может быть представлена информация о пользователях. Как и другие классы объектов, класс user определяет набор обязательных и необязательных атрибутов. В число семи обязательных атрибутов объектов этого класса входит, в частности, *каноническое имя пользователя,* а ***номер телефона*** является примером одного из 250 возможных необязательных атрибутов.

Когда администратор регистрирует нового пользователя, в базе данных Active Directory для этого пользователя создается учетная запись, которая и представляет собой объект. Для всех обязательных и для некоторых необязательных атрибутов данного объекта устанавливаются вполне определенные значения. Например, регистрируя в качестве пользователя некую Полину, администратор определяет значение канонического имени пользователя (обязательного атрибута) — Polina, и значение номера телефона (необязательного атрибута) — 22345777. Таким образом, появляется новый объект класса user.

Глобальный каталог

Задача распределенной справочной службы состоит в предоставлении клиентам контролируемого доступа ко ***всем*** объектам сети, даже если источник запроса и запрашиваемый ресурс находятся в разных доменах.

Для решения этой задачи используется *глобальный каталог.* В отличие от доменных баз данных, которые хранят объекты, относящиеся только к собственным доменам, в базе данных глобального каталога хранится информация обо всех объектах сети. Однако в отличие от доменных баз данных, хранящих объект со всеми его атрибутами, в глобальном каталоге каждый объект представлен в виде «усеченной» версии, которая, как минимум, должна содержать атрибут*, указывающий на местонахождение полной версии объекта. Такого рода атрибутом для большинства объектов является *отличительное имя (Distinguished Name, DN*), которое однозначно в пределах всей сети идентифицирует объект.

Глобальный каталог легко достижим для запросов всех клиен- 1ов, так как в каждом домене хранится одна или несколько его копий, обычно администратор выделяет для хранения глобального катало- i,i, по меньшей мере, по одному контроллеру на каждый сайт, такие контроллеры называют также серверами глобального каталога.

Для ***поиска объектов*** пользователь может направлять запрос к

Active Directory, указывая отличительное имя объекта. Это имя подобно полному составному символьному имени файла в иерархической файловой системе, только в нем вместо имен каталогов указы- идются имена доменов и других узлов иерархической структуры ба- 1Ы данных объектов. Средства пользовательского интерфейса поз- иоляют пользователю обращаться к объекту по его краткому имени — *относительному отличительному имени.* В этом случае служба Active Directory сама дополняет его до полного имени, используя контекстную информацию. Аналогично поиску файлов в файловой | истеме, поиск объекта может осуществляться и по значению какого- либо его атрибута. Например, чтобы найти объект класса user, клиент справочной службы может указать имя пользователя или адрес его электронной почты, а при поиске принтера — его тип. В этом I лучае Active Directory может вернуть клиенту информацию о нескольких объектах, каждый из которых удовлетворяет запросу, давая возможность пользователю самостоятельно выбрать интересующий его объект. Информация в главном каталоге позволяет определить DNS-имя контроллера, в котором хранится объект. На основании этого имени система DNS определяет IP-адрес контроллера, после чего задачу доступа к требуемому объекту можно считать решенной.

Наряду с поиском объектов, на основе глобального каталога решается еще одна важнейшая задача справочной службы — *глобальная аутентификация пользователей.* Слово «глобальная» в данном случае означает, что пользователь при определенных условиях может выполнять логический вход в сеть с любого компьютера любого домена сети. Такая принципиальная возможность появляется благодаря тому, что в глобальном каталоге хранится информация об *универсальных группах* пользователей, в которые могут включаться члены разных доменов. А это значит, что для процедуры аутентификации пользователя, входящего в одну из таких групп, достаточно обращения к ближайшему контроллеру локального домена, который хранит копию глобального каталога. Например, если сотруднику некоторого предприятия в Томске, оказавшемуся в командировке в барнаульском отделении, потребовалось выполнить некоторую работу на компьютере, то он может войти в сеть с любого компьютера сети предприятия в Барнауле независимо от того, относятся ли сети в Томске и Барнауле к одному и тому же домену или нет. Набрав идентификатор и пароль, полученные при регистрации от администратора в томском отделении, командированный сотрудник получает доступ к ресурсам сети в соответствии с теми же правами и разрешениями, которые он имел, входя в сеть со своего рабочего компьютера в Томске. Невидимый пользователю процесс междоменной аутентификации мы рассмотрим немного позже.

Active Directory позволяет также выполнять *глобальную авторизацию* при обращении к какому-либо ресурсу, расположенному в удаленном

домене, приложению или пользователю не требуется взаимодействовать для проверки правомочности доступа с контроллером этого домена, достаточно обратиться к ближайшему серверу глобального каталога, в котором, помимо атрибутов о местонахождении каждого из объектов сети, моут храниться атрибуты, описывающие, какой вид доступа к этим объектам разрешен.

Кроме универсальных групп пользователей существуют также локальные и глобальные группы пользователей домена.

Члены *локальной группы пользователей* могут получать доступ только к ресурсам данного домена. В то же время членами локальной группы домена могут быть как отдельные пользователи, так и глобальные и универсальные группы любых доменов.

Члены *глобальной группы пользователей* могут получать доступ к ресурсам любых доменов. Членами глобальной группы могут быть отдельные пользователи и глобальные группы из того же домена.

Члена *универсальной группы пользователей* могут получать доступ к ресурсам любых доменов леса. Членами универсальной группы могут быть отдельные пользователи, глобальные и универсальные группы любых доменов леса.

Иерархия организационных единиц Active Directory

Как и все современные справочные службы, Active Directory является иерархически организованной системой. Иерархия логически упорядочивает информацию о многочисленных объектах, дает возможность «увидеть» всю систему в целом, упрощает процесс именования объектов и делает более эффективным выполнение индивидуальных и групповых действий над ними, таких как поиск, удаление, изменение свойств и т. д.

В зависимости от размеров сети администратор создает более или менее сложную иерархическую структуру в Active Directory. Основными строительными блоками иерархической структуры Active Directory являются домены и организационные единицы.

В пределах каждого домена может существовать иерархия организационных единиц.

Организационная единица (*OU*) — это специфический объект Active Directory, который представляет группу объектов, объединенных в соответствии с теми или иными их свойствами.

Если таким свойством является, например, тип объектов, то организационные единицы могут представлять собой группы пользователей, компьютеров или принтеров. Территориальная общность или принадлежность к одному и тому же подразделению также может быть использована для группирования объектов в организационные единицы,

например: OU филиала предприятия в Москве и OU филиала в Перми, OU рекламного отдела, OU отдела перспективных разработок.

Как следует из определения, организационные единицы являются объектами. Такого рода объекты, используемые исключительно с целью группирования других объектов, называются также *контейнерами* ***(container).*** В этом случае оказывается полезной аналогия с файловой системой, в которой каталоги могут рассматриваться как контейнеры, содержащие внутри себя логически сгруппированные файлы. Контейнер, как и любой объект, имеет атрибуты и относится к определенному классу объектов.

Организационные единицы не только являются логической группой объектов, но и сами могут быть включены в другие организационные единицы. Группируя таким образом OU, можно образовывать древовидные структуры (OU-деревья), подобные древовидным структурам файловой системы. В Active Directory для каждого из доменов строится отдельное OU-дерево. В качестве корня в OU-дереве выступает не организационная единица, а объект, соответствующий домену, для которого построено это дерево. OU-дерево, показанное на рис. 16.7, включает два уровня организационных единиц. На верхнем уровне расположены OU офиса предприятия в Москве и OU центрального отделения в Барнауле. В каждую из этих организационных единиц входит еще по две организационные единицы: computers и users для московского офиса user и hardware для отделения в Барнауле.

Иерархическое структурирование объектов позволяет эффективно выполнять ***групповые операции*** (переименование, уничтожение, определение правил доступа и т. п.) сразу для нескольких объектов, представленных как отдельной организационной единицей, так и ветвью OU-дерева, а также ***делегировать администрирование*** отдельными объектами и ветвями OU-дерева другим администраторам. В примере на рисунке для каждого из двух подразделений предприятия администратором была создана отдельная ветвь OU-дерева, что дает возможность назначить для этих двух подразделений, находящихся в

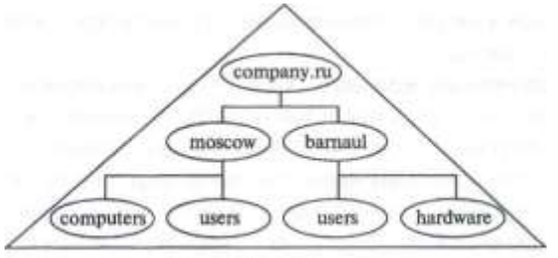


Рис. 16.7. OU-дерево

разных часовых поясах, двух разных администраторов, которые смогут более эффективно управлять пользователями и ресурсами, относящимися к этим структурным единицам. В частности, локальным администраторам может быть передано право создавать пользовательские учетные записи, переустанавливать пароли и выполнять другие действия, которые проще выполнять в непосредственной близости к пользователям.

Иерархия доменов. Доверительные отношения

Домены являются более независимыми структурными единицами, чем организационные единицы внутри домена. В частности, именно пределами домена, а не OU, ограничиваются действие политики паролей (password policies), устанавливающей длину и другие параметры паролей пользователей, и действие политики блокировки учетных записей (account lockout policies), определяющей функции администратора в тех случаях, когда пользователь забыл или не смог правильно набрать свой пароль и его учетная запись была заблокирована. Разграничение функций по администрированию на уровне доменов происходит очень естественно благодаря тому, что каждый домен имеет встроенные группы Administrators и Server Operators, членам которых разрешено совместно использовать папки и форматировать диски только на контроллерах своего домена.

Такие свойства доменов делают привлекательной идею представления сети в виде нескольких доменов. Наиболее часто используемой многодоменной структурой является *дерево доменов* (рис. 16.8).

На рисунке показано трехуровневое дерево доменов. Каждый домен условно изображен в виде треугольника. Упорядочение доменов по уровням иерархии происходит аналогично упорядочиванию каталогов файловой системы. Первый по времени создания домен становится *корнем дерева.* Для следующего создаваемого домена, называемого *потомком,* корневой домен является *родительским.* Далее при создании каждого последующего домена необходимо выбрать, какой

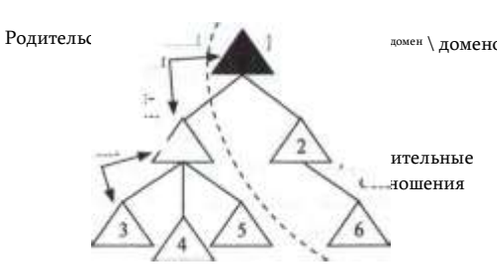


Рис. 16.8. Дерево доменов

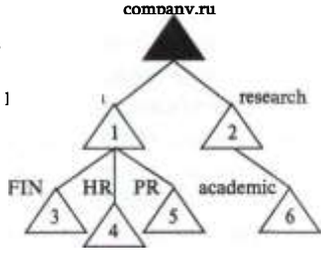


Рис. 16.9. Пространство имен дерева доменов

Домен- из существующих доменов будет его родительским доменом. Корневой домен на рисунке имеет двух потомков: домен 1 и домен 2. Каждый из них в свою очередь является родительским доменом для доменов третьего уровня. Домены 3, 4 и 5 — потомки домена 1, а домен б — потомок домена 2. Дерево доменов, показанное на рисунке, состоит из двух *ветвей.*

Иерархические отношения между доменами выражаются в том числе и в их именовании. Корневому домену при его создании присваивается DNS-имя, пусть это будет, например, company.ru (рис. 16.9). Все домены-потомки получают «в наследство» имя родительского домена, которое добавляется к их собственным именам, образуя DNS- имена следующего уровня. В нашем примере домен production имеет полное имя production.company.ru, а домен academic — имя academic. research.company.ru. Множество полученных таким образом имен доменов (а, как дальше мы увидим, и объектов) образует пространство имен дерева доменов.

Между каждым новым доменом-потомком и его родительским доменом автоматически устанавливаются так называемые *доверительные отношения.* Установление доверительных отношений включает передачу части доменного справочника присоединяемого домена в глобальный каталог родительского домена, после чего он становится общим для всего дерева доменов. Глобальный каталог создается автоматически во время установки первого контроллера в первом домене сети. Первоначально в главном каталоге размещается полная копия всех объектов, относящихся к этому первому домену. В результате создания новых доменов к главному каталогу прибавляются частичные копии баз данных каждого из этих доменов. Эти копии содержат атрибуты объектов, позволяющие определить местонахождение соответствующих объектов, а значит, дающие пользователям возможность получать доступ к объектам не только своих, но и других доменов.

Для обеспечения безопасности в процедуру установления доверительных отношений вовлечена система аутентификации Kerberos.

Установление доверительных отношений между двумя доменами дает возможность:

- пользователю (или группе пользователей) одного домена получать доступ к ресурсам другого домена (это означает, что нет необходимости создавать две учетные записи в этих двух доменах для одного и того же пользователя);
- администрировать один домен из другого домена, для чего необходимо включить в группу пользователей, наделенную правами администрирования домена, пользователей из другого домена.

В Active Directory доверительные отношения, устанавливаемые по умолчанию между родительским доменом и доменом-потомком, являются транзитивными и двусторонними.

Отношения *транзитивны,* если из того, что *А* доверяет *В, а* ***В*** доверяет *С,* автоматически следует, что *А* доверяет *С.* Отношения являются *двусторонними,* если из того, что *А* доверяет *В,* следует, что *В* доверяет *А.* Отсюда следует, что все домены дерева связаны друг с другом двусторонними транзитивными доверительными отношениями.

В Active Directory OU-деревья разных доменов никак не связаны между собой в отличие, например, от справочной службы NDS компании Novell, в которой OU-деревья всех доменов сети образуют единое дерево.

Active Directory может иметь и более сложную доменную структуру, состоящую из нескольких деревьев и называемую **лесом** (рис. 16.10). Все домены леса так же, как и домены дерева, связаны двусторонними транзитивными доверительными отношениями, т. е. имеют общий глобальный каталог. Следовательно, пользователи сети, имеющей доменную структуру в виде леса, могут быть наделены правом доступа к любым ресурсам любого домена.

Корневые домены каждого дерева, входящего в лес, получают независимые друг от друга DNS-имена, порождающие непересекаю- щиеся пространства имен. То есть в отношении именования доменов (а как мы позже увидим и объектов) деревья леса равноправны.

В то же время корневые домены деревьев, составляющих лес, относятся к разным уровням иерархии, а именно, домен, который по времени был создан раньше (пусть это домен comp.ru), становится ***корневым доменом леса*** и находится на более высоком уровне иерархии, нежели созданный позже домен branch.ru. Корневой домен леса отличается от всех корневых доменов деревьев, составляющих лес, тем, что только в корневом домене леса имеются встроенные группы Enterprise Admins и Schema Admins, члены которых наделе-

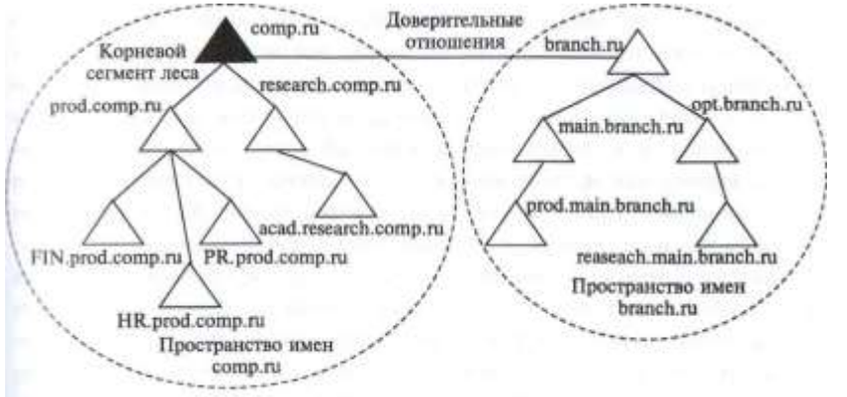


Рис. 16.10. Лес доменов

ны правом администрирования в пределах леса. Однако несмотря м.) более низкое положение в иерархии, было бы ошибкой интерпре- ировать фрагмент леса, порождаемый доменом branch.ru, как ветвь дерева comp.ru. Между доменами branch.ru и comp.ru нет отношений родитель-потомок и, как следствие, у них разные пространства имен.

Лес доменов Active Directory определяет границу действия механизмов и политик безопасности сети, так как за счет глобального ка- шлога и доверительных отношений пользователи любого домена леса могут быть аутентифицированы при логическом входе из компьютера любого другого домена и на них могут распространятся общие пра- нила политики леса доменов при соответствующей организации групп пользователей.

Еше одной доменной структурой, используемой в Active Directory, является модель ***множественных лесов*** Эта модель может оказаться полезной для крупных транснациональных корпораций, образованных н результате слияния нескольких предприятий, а также в случаях, когда организация представляет собой объединение нескольких крупных и достаточно независимых подразделений. Являясь наиболее децен- !рализованной структурой, модель множественных лесов состоит из нескольких изначально совершенно не связанных между собой лесов. Между доменами разных лесов могут быть установлены контролируемые администратором доверительные отношения, однако это происходит не автоматически, как при построении деревьев и лесов, а в результате некоторых дополнительных действий администратора.

Пространство имен

Пространство имен (***namespace***) справочной службы — это множество имен, однозначно соответствующих всем объектам и их атрибутам, хранящимся в данной справочной службе.

Пространство имен является необходимым компонентом любой справочной системы, позволяя по имени объекта (здесь мы используем слово «объект» в широком смысле) находить информацию, связанную с этим объектом, Например, в адресных или телефонных справочниках, в которых информацию об адресе или телефоне человека можно найти, зная имя этого человека, пространством имен является множество фамилий. Этот пример является не совсем точным, так как в реальной жизни существуют однофамильцы, а значит, соответствие между фамилиями и персонами, информация о которых представлена в справочнике, является неоднозначной. В компьютерных системах, в которых поиск информации выполняется автоматически, однозначность необходима. Так, множество имен файлов, сгенерированных на основе принятых в файловой системе правил, однозначно определяет существующие в системе файлы. Другим примером является система DNS, используемая для именования узлов в 1P-сети, она также включает в себя пространство DNS-имен, однозначно соответствующих IP-адресам.

Рассмотрим, как формируется пространство имен Active Directory. В Active Directory поддерживается несколько форм записи имени объекта. Мы остановимся на системе записи имен, принятой в стандарте LDAP. Согласно требованиям протокола LDAP, имя объекта должно представлять собой последовательность имен всех компонентов иерархии, лежащих на пути от объекта до корня дерева.

В Active Directory, как и в любой иерархической системе, имеющей древовидную структуру, имеется единственный путь от корня дерева до его конечного элемента (листа). Уникальность путей ко всем листьям в древовидных структурах делает уникальными имена, полученные перечислением имен транзитных узлов, лежащих на этих путях. Такие имена, однозначно определяющие объекты в пределах всего дерева, часто называют полными (составными) именами. Аналогом полного имени в Active Directory служит уже упоминавшееся отличительное имя36 объекта, которое описывает путь от данного объекта до корневого домена.

Другой тип имени, используемый в Active Directory, называется ***относительным отличительным именем37 (Relative Distinguished Name, RDN***). Это имя представляет собой компонент отличительного имени объекта, однозначно определяющий объект в пределах кон- И'йнера. Это имя подобно краткому имени каталога или файла и представляет собой значение одного атрибута (очень редко нескольких атрибутов) данного объекта, обеспечивающего уникальность этого объекта в контейнере. Из всего этого следует, что отличительное имя объекта есть цепочка

36 Можно встретить и другой вариант перевода — различающееся имя.

37 Не путать с относительным именем файла, которое определяет положение файла в дереве относительно текущего каталога. В Active Directory понятие текущего каталога (домена, организационной единицы) при именовании объектов не используется.

относительных отличительных имен объектов, находящихся на пути от объекта до корня.

Компактность относительного имени делает его удобным для применения в пользовательском интерфейсе, хотя в LDAP-пакетах всегда указывается отличительное имя.

Особенность именования объектов в Active Directory, отличающая ее, например, от именования файлов, состоит в неоднородности узлов дерева объектов. Действительно, в то время как в дереве файловой системы все транзитные узлы на пути от корневого каталога к файлу имеют единую природу — все они являются каталогами, в Active Directory путь, соединяющий объект с корневым каталогом, делится на две части: одна часть пролегает по дереву доменов, а другая проходит по дереву организационных единиц, находящемуся - внутри домена» (рис. 16.11). То есть компоненты полного имени объекта имеют разную природу. Этот факт отражается в синтаксисе имен Active Directory — имя строится приписыванием соответствующих префиксов к компонентам имени. К числу основных префиксов* относятся:

- DC (Domain Component) — компонент является относительным именем домена;
- OU (Organizational Unit) — компонент является относительным именем организационной единицы;

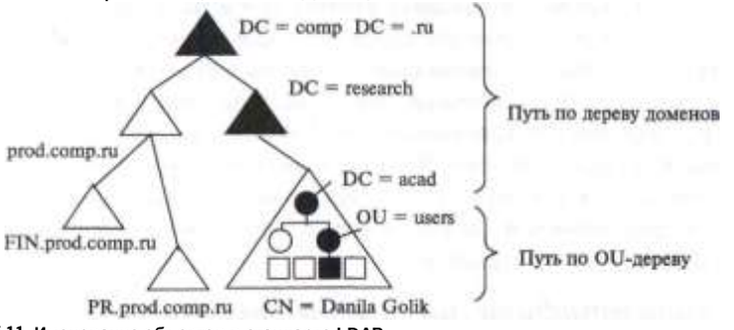


Рис. 16.11. Именование объектов в стандарте LDAP

*** Стандарт LDAP предусматривает и другие префиксы: C (страна), O (организация) и L (местонахождение), однако в Active Directory они не используются.**

• CN (Common Name) — компонент относится к объекту любого типа, не являющемуся ни организационной единицей, ни доменом.

Посмотрим, как будет выглядеть отличительное имя в стандарте LDAP для объекта, представляющего, например, пользователя Данилу Голика, который был помещен администратором в организационную единицу, имеющую относительное имя users. Эта организационная единица относится к домену, DNS-имя которого acad.research. comp.ru. В соответствии с принятым синтаксисом перечисление компонентов начинается от листа OU-дерева, которым является объект- пользователь, до корня дерева доменов. Каждое относительное имя предваряется соответствующим префиксом. В результате получаем

CN = Danila Golik, OU = users, DC = acad, DC = research, DC = comp, DC = ru

Как уже было сказано, в Active Directory поддерживается несколько форм записи имени объекта, например для приведенного имени альтернативой может служить одно из следующих имен:

acad.research.comp.ru/users/Danila Golik

DC=ru/DC=comp/DC=research/DC=acad/OU=users/CN=Danila Golik

Получив от клиента отличительное имя, справочная служба должна установить контакт с контроллером того домена, где находится интересующий клиента объект. Из имени объекта легко определяется DNS-имя контроллера, а для определения IP-адреса Active Directory использует систему доменных имен (DNS).

Контроллеры доменов Active Directory могут отличаться друг от друга набором предоставляемых услуг. Для того чтобы пользователи и приложения могли находить именно те контроллеры, на которых установлены необходимые им сервисные программные модули, контроллеры регистрируют имена своих сервисов в системе DNS, которая по запросам клиентов сообщает им о местонахождении (1P-адресах) серверов и сервисных программ Active Directory. В некоторых случаях система DNS может вернуть IP-адреса нескольких контроллеров, предоставляющих эту услугу. Используя дополнительную информацию о месте расположения контроллеров, клиентская станция выбирает ближайший к ней контроллер.

Аутентификация в многодоменной структуре Active Directory

Как было отмечено выше, Windows обладает модульной системой аутентификации, при этом доменную аутентификацию обеспечивают два типа модулей — Kerberos и MSV.1.

Модуль MSV_1 поддерживает протокол аутентификации NTLM

(версии 1 и 2), который использовался в доменах Windows NT и считается теперь устаревшим и используемым только для обратной совместимости с доменами, построенными на контроллерах или рабочих станциях Windows NT. Мы не будем рассматривать этот способ доменной аутентификации подробно, отметим только, что он использует так называемую «транзитную» (pass-through) аутентификацию, когда ресурсный сервер обменивается с аутентифицируемым пользователем словом-вызовом, но передает ответ пользователя для проверки аутентичности контроллеру домена, так как только контроллер домена знает пароли пользователей. Этот способ вносит задержки в процесс аутентификации, так как каждое обращение клиента к ресурсному серверу требует обращения к контроллеру домена. Механизм аутентификации Kerberos свободен от этого недостатка, так как квитанция на доступ к ресурсному серверу может использоваться многократно.

Kerberos-аутентификация в пределах одного домена не требует больших пояснений, так как ее схема в точности совпадает с описанной выше в разделе «Система Kerberos». Kerberos KDC в этом случае работает на контроллере домена, там же располагается база учетных данных пользователей и ключи ресурсных серверов. При логическом входе пользователя в домен происходит интерактивная аутентификация, при этом в качестве ресурсного сервера выступает компьютер пользователя, который должен быть членом домена. При успешной аутентификации пользователь получает доступ к своему компьютеру как пользователь домена, а также как член одной или нескольких групп домена — при условии, что данному пользователь и/или группам, в которые он входит, дано право логического входа в данный компьютер.

Сетевая (неинтерактивная аутентификация) при доступе пользователя к ресурсному серверу, отличному от компьютера, на котором он интерактивно работает, также происходит в точности со схемой, описанной в разделе «Система Kerberos».

В случаях, когда компьютер пользователя или ресурсный сервер находятся в домене, отличном от того домена, где был создана учетная запись пользователя, схема Kerberos-аутентификации немного усложняется. Как мы помним, аутентификационная информация пользователя — пароли — никогда не копируется в копии глобального каталога, а всегда хранится только в базе пользовательских данных того домена, к которому пользователь относится. Поэтому Kerberos- г.ерверы различных доменов должны взаимодействовать в том случае, когда пользователь и ресурс находятся в разных доменах.

В качестве примера мы рассмотрим простой случай, когда имеется три домена, входящие в одно дерево. Эти домены показаны на

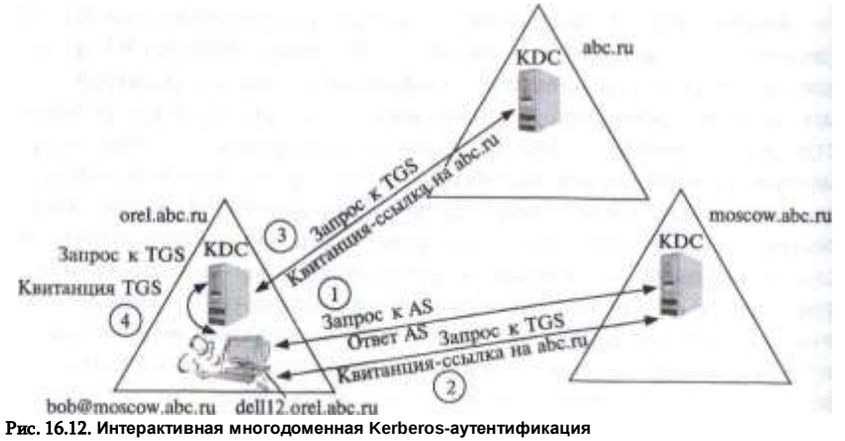


Рис. 16.12. Интерактивная многодоменная Kerberos-аутентификация

рис. 16.12: корнем дерева является домен abc.ru, а листьями — домены moscow.abc.ru и orel.abs.ru. Между доменами существуют доверительные отношения «родитель-потомок», установленные автоматически при создании доменов-потомков. Важным фактом для рассмотрения междоменной Kerberos-аутентификации является то, что при установлении доверительных отношений создается разделяемый междоменный ключ, который хранится в базе Kerberos KDC каждого домена.

Рассмотрим отдельно случаи интерактивной и неинтерактивной многодоменной аутентификации.

Интерактивная аутентификация

Боб является пользователем домена moscow.abc.ru, о чем говорит его имя bob@moscow.abc.ru. В рассматриваемом случае Боб выполняет логический вход с компьютера delll2.orel.abc.ru, являющегося членом домена orel.abc.ru, т. е. чужого домена (см. рис. 16.12).

После ввода имени bob@moscow.abc.ru модуль Kerberos-аутентификации компьютера delll2.orel.abc.ru отправляет запрос на аутентификацию AS серверу домена moscow.abs.ru, так как только этот сервер хранит пароли пользователей этого домена moscow.abc.ru. Получив этот запрос, сервер AS направляет обычный ответ, в котором содержатся квитанция на доступ к серверу квитанций TGS и ключ сеанса, зашифрованные паролем пользователя.

Получив ответ AS, модуль Kerberos компьютера delll2.orel.abc.ru расшифровывает квитанцию и ключ сеанса, используя пароль, введенный Бобом.

Второй этап состоит в отправке компьютером delll2.orel.abc.ru запроса к серверу квитанций домена moscow.abc.ru за квитанцией доступа Боба к компьютеру delll2.orel.abc.ru. Сервер TGS домена mos- cow.abc.ru не может выдать такую квитанцию, так как ресурс находится не в его домене. Вместо

этого он отправляет так называемую квитанцию-ссылку, которая указывает на домен abc.ru как на ближайший домен, с которым у домена orel.abc.ru, которому принадлежит ресурс, есть непосредственные доверительные отношения (а не транзитные). Квитанция-ссылка шифруется междоменным ключом, разделяемым доменами abc.ru и moscow.abc.ru.

Третий этап состоит в обращении компьютера delll2.orel.abc.ru с квитанцией-ссылкой к серверу TGS домена abc.ru. Этот сервер расшифровывает квитанцию-ссылку междоменным ключом и генерирует квитанцию доступа к компьютеру delll2.orel.abc.ru, зашифрованную ключом этого компьютера, вместе с новым ключом сеанса, а затем зашифровывает эту квитанцию старым ключом сеанса пользователя.

Четвертый этап является обычным для однодоменной схемы работы Kerberos — модуль Keberos-аутентификации пользователя расшифровывает квитанцию доступа старым ключом сеанса и передает ее вместе с аутентификатором пользователя, зашифрованным новым ключом сеанса, модулю Kerberos-аутентификации ресурсного сервера, в качестве которого выступает Kerberos-модуль компьютера dell 12. orel.abc.ru. Последний расшифровывает квитанцию ключом, разделяемым с сервером квитанций TGS, и извлекает из нее новый ключ сеанса, с помощью которого расшифровывает аутентификатор.

Неинтерактивная аутентификация

Этот случай иллюстрируется рис. 16.13, на котором представлены те же три домена, что мы рассматривали в предыдущем разделе, но на этот раз пользователь Алиса из домена orel.abc.ru работает за компьютером com33.orel.abc.ru, который также принадлежит этому домену.

Первый этап. Алиса уже аутентифицировалась для работы с этим компьютером (этот этап мы опускаем) и получила квитанцию на доступ к серверу TGS. Теперь ей нужен доступ к почтовому серверу mail.moscow.abc.ru, который находится в домене moscow.abc.ru. Сначала Алиса обращается за квитанцией доступа к почтовому серверу к серверу квитанций своего домена. Однако этот сервер не может ей выдать искомую квитанцию, так как ресурс находится за пределами его пространства имен. Поэтому он возвращает квитанцию-ссылку на сервер abc.ru, с которым у домена, которому принадлежит ресурс, есть прямые доверительные отношения.

Второй этап. Сервер TGS домена abc.ru возвращает квитанцию-ссылку на сервер TGS домена moscow.abc.ru, которому принадлежит ресурс mail.moscow.abc.ru.

Рис. 16.13. Неинтерактивная многодоменная Kerberos-аутентификация

Третий этап. Модуль Kerberos компьютера com33.orel.abc.ru отправляет запрос серверу TGS домена moscow.abc.ru на доступ к серверу mail.moscow.abc.ru. Сервер TGS может обслужить этот запрос и отправляет квитанцию на доступ.

Четвертый этап. Модуль Kerberos компьютера com33.orel.abc.ru отправляет серверу mail.moscow.abc.ru квитанцию доступа вместе с аутентификатором Алисы. Сервер mail.msocow.abc.ru обычным для Kerberos способом проверяет аутентичность Алисы.

При описании неинтерактивной аутентификации мы опустили подробности использования ключей, но читатель может дополнить этот пробел сам, пользуясь схемой интерактивного доступа.

Реализация Kerberos в системе Active Directory использует фирменное расширение Micorsoft этого протокола, помогающее авторизации пользователей ресурсными серверами. Для этого квитанция на доступ к ресурсу включает *сертификат привилегий пользователя* ***(Privilege Attribute Certificate, PAC),*** который содержит идентификаторы групп, которым принадлежит пользователь, а также права пользователя.

Вопросы к главе 16

1. Как соотносятся между собой системы аутентификации ОС и приложений?

а) система аутентификации ОС и система аутентификации серверной части приложения всегда независимы друг от друга;

б) система аутентификации ОС всегда берет на себя задачу аутентификации пользователей приложений;

в) в соответствии с принципом эшелонированной защиты система аутентификации ОС и система аутентификации серверной части приложения всегда работают совместно;

г) система аутентификации ОС и система аутентификации серверной части приложения могут работать независимо;

д) ни одно утверждение не верно.

2. Какие из приведенных ниже утверждений ошибочны?

а) локальная система аутентификации может быть использована для аутентификации удаленного пользователя;

б) «Горячие» клавиши CTRL-Alt-Del рекомендуется использовать для быстрого вызова панели логического входа;

в) доменная аутентификация может быть использована для аутентификации удаленных пользователей, принадлежащих одному DNS-домену;

г) ни одно из них.

3. Правила политики паролей ОС Windows позволяют задать:

а) максимальное требуемое число новых паролей до того, как пользователь может использовать старый пароль вновь;

б) максимальный срок действия пароля;

в) минимальный срок действия пароля;

г) минимальную длину пароля.

4. Какие средства могут быть использованы для аутентификации пользователей UNIX?

а) утилита login;

б) удаленная аутентификация по протоколу telnet;

в) многофункциональный пакет Secure Shell;

г) программная среда PAM;

д) сервис локальной безопасности SLA;

е) Kerberos.

5. Протокол SSH:

а) работает в архитектуре клиент-сервер;

б) выполняет защищенное копирование между удаленными хостами;

в) использует RSA;

г) обеспечивает целостность передаваемых данных;

д) является средством аутентификации;

е) передает данные в зашифрованном виде.

6. Вставьте слова из списка в предложение «В системе ... команда... используется для временного получения пользователем статуса суперпользователя .... а команда ... для получения права на выполнение некоторых привилегированных операций». В ответе перечислите термины в том порядке, в котором они следуют в предложении:

а) root;

б) Unix;

в) Windows;

г) su;

д) sudo.

7. К какому классу операций доступа ОС семейства Windows относится запрет установки системного времени?

а) индивидуальные разрешения;

б) стандартные разрешения;

в) права;

г) блокировки;

д) возможности;

е) ни к одному из перечисленных.

8. Какую информацию включает токен доступа пользователя в системе управления доступом ОС Windows?

а) идентификатор пользователя;

б) идентификаторы всех групп, в которые входит пользователь;

в) список возможностей пользователя;

г) список разрешений пользователя ACL по умолчанию;

д) список прав пользователя.

9. Основные функции системы Kerberos:

а) аутентификация;

б) авторизация;

в) шифрование.

10. Какие из следующих положений реализованы в системе Kerberos?

а) Kerberos является посредником во всех процедурах аутентификации, которые выполняются между клиентами серверами сети;

б) все клиенты сети доверяют системе Kerberos;

в) все серверы сети доверяют системе Kerberos;

г) в системе Kerberos клиенты должны доказывать свою аутентичность для доступа к серверу;

д) в системе Kerberos серверы также должны доказывать свою аутентичность для работы с клиентом;

е) в системе Kerberos все обмены данными в сети выполняются в защищенном виде с использованием симметричного алгоритма шифрования AES.

11. Каким образом выполняется аутентификация пользователя, если его учетные данные хранятся в одном домене Active Directory, а он выполняет логический вход с компьютера, принадлежащего другому домену?

а) учетные данные пользователя копируются в базу аутентификационных данных контроллера домена, с компьютера которого пользователь делает логический вход;

б) запрос на аутентификацию направляется в тот контроллер домена, где хранятся его учетные данные пользователя, а квитанцию на доступ к компьютеру выдает контроллер домена, к которому принадлежит компьютер;

в) учетные данные пользователя копируются в главный каталог, и аутентификация выполняется на контроллере центрального домена.

17 Аудит *СОБЫТИЙ, ИМЕЮЩИХ ОТНОШЕНИЕ К БЕЗОПАСНОСТИ*

Современные операционные системы имеют развитые средства аудита событий, которые позволяют регистрировать заданные классы и типы событий в системном журнале, а также анализировать зарегистрированные события с применением разнообразных фильтров. События, связанные с безопасностью ОС, представляют собой важный класс событий, а их регистрация является прямым способом реализации уже упомянутого принципа «предотвращение вторжений — это идеально, но их обнаружение — это абсолютная необходимость».

Важно, чтобы средства аудита ОС были настраиваемыми, т. е. позволяли администратору системы задавать аудит тех типов событий безопасности, которые представляют для него особый интерес, так как регистрация всевозможных событий может породить слишком большое количество записей, которое будет практически невозможно анализировать. Список событий, которые должна регистрировать система для того, чтобы претендовать на получение сертификата безопасности, определен документом Гостехкомиссии РФ: «Положение о сертификации средств и систем вычислительной техники и связи по требованиям безопасности информации».

Аудит событий в ОС Windows

ОС Windows регистрирует событий безопасности в журнале Security Log, который можно просматривать с помощью утилиты Event Viewer. Типы регистрируемых событий безопасности определяются параметрами политики аудита, которая определяется или для отдельного компьютера или же для домена в целом.

Политика аудита задается для различных типов событий безопасности, при этом для каждого типа администратор может отдельно задать регистрацию успешных и неуспешных событий. Например, если администратор задал аудит неуспешных операций удаления файла, то в случае его успешного удаления запись об этом событии в журнале создана не будет, а вот если попытка удаления файла оказалась неуспешной из-за отсутствия у пользователя прав на эту операцию, то соответствующая запись будет создана и запомнена.

Рассмотрим типы аудита событий безопасности, определяемые политикой аудита, более подробно. В исходном состоянии (т. е. после установки OG) все типы аудита событий безопасности не активны (не определены), однако для каждого типа имеется индивидуальное поведение по умолчанию, которое работает даже в том случае, когда аудит этого типа не активен.

Аудит входа в систему. Эта группа событий относится к логическому входу пользователя в отдельный компьютер (но не домен). Это может быть как интерактивный, так и не-интерактивный вход, например, сетевой доступ к разделяемому каталогу. Поведением по умолчанию является регистрация успешных событий входа.

Аудит доступа к объектам. Объектами являются файлы, каталоги файлов, принтеры, ключи Registry. Аудит попыток доступа можно выполнять только для тех объектов, для которых существует список управления доступом ACL, поэтому для файлов и каталогов разделов NTFS аудит возможен, а для файлов и каталогов FAT — нет. Для регистрации событий, связанных с доступом к конкретному объекту, недостаточно только активизировать этот вид аудита в списке политик аудита, нужно также активизировать его в списке типов аудита событий безопасности этого объекта. Например, если мы хотим выполнять аудит успешных и неуспешных операций удаления файлов и подкаталогов из каталога C:\Study Notes, а также удаления самого каталога, то для этого нужно открыть закладку Auditing в панели Advance Security каталога C:\Study Notes, и отметить опции Successful и Failed для операций Delete subfolders and files и Delete (рис. 17.1). После этого все попытки удаления самого каталога C:\Study Notes и его подкаталогов и файлов будут регистрироваться в журнале Security Log системы при условии, что аудит доступа к объектам активизирован.

Аудит доступа к справочной службе Active Directory. Аудит выполняется для операций доступа к любым объектам Active Directory: пользователям, группам пользователей, компьютерам, принтерам. Как и в случае аудита доступа к объектам, кроме общей активизации аудита доступа к службе Active Directory требует задания опций аудита для конкретного объекта.

Аудит изменения политики. Этот вид аудита фиксирует изменения политики прав пользователя, политики аудита и политики доверительных отношений.

Аудит использования привилегий. Фиксируются случаи использования пользователями специальных прав, например права выключать систему или изменять системное время.

Аудит отслеживания процессов. Этот вид аудита фиксирует события, связанные с программными процессами: создание или завер-

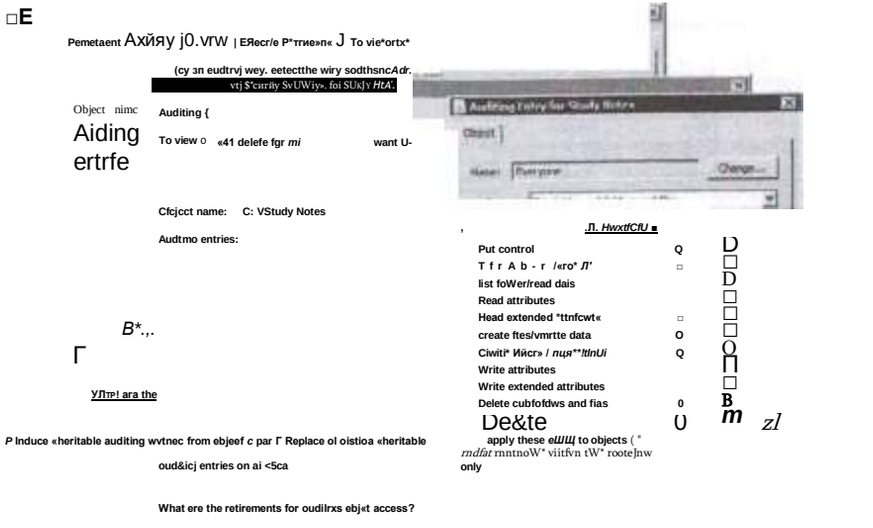


Рис. 17.1. Задание опций аудита для каталога

шение процесса, запуск приложения и т. п. Этот вид аудита рассчитан прежде всего на программистов, отлаживающих приложение.

Аудит системных событий. Фиксируются события, важные для операционной системы в целом: рестарт системы, очистка журнала безопасности Security Log, изменение системного времени и многие другие. Любое из этих событий может свидетельствовать об атаке, например очистка журнала безопасности может говорить о том, что злоумышленник пытался уничтожить следы атаки. Flo если аудит системных событий был активизирован, то после очистки журнала безопасности в него будет занесено сообщение о том, что очистка имела место, что очень полезно для обнаружения атак.

Аудит входа по учетной записи. Этот вид аналогичен аудиту логического входа за тем исключением, что фиксируются случаи логического входа пользователя с аутентификацией службой Active Directory.

Просмотр журнала безопасности Security Log выполняется с помощью утилиты Event Viewer — для этого просто нужно выбрать нужный журнал из ее меню (рис. 17.2). События в журнале безопасности по умолчанию упорядочены по времени, но их можно упорядочить и по любому параметру, отображаемому в окне Event Viewer.

Журнал безопасности может содержать несколько тысяч событий. Для того чтобы можно было быстро найти интересующее событие или события, Event Viewer оснащён функциями поиска события, а также фильтрации по

различным признакам (рис. 17.3).

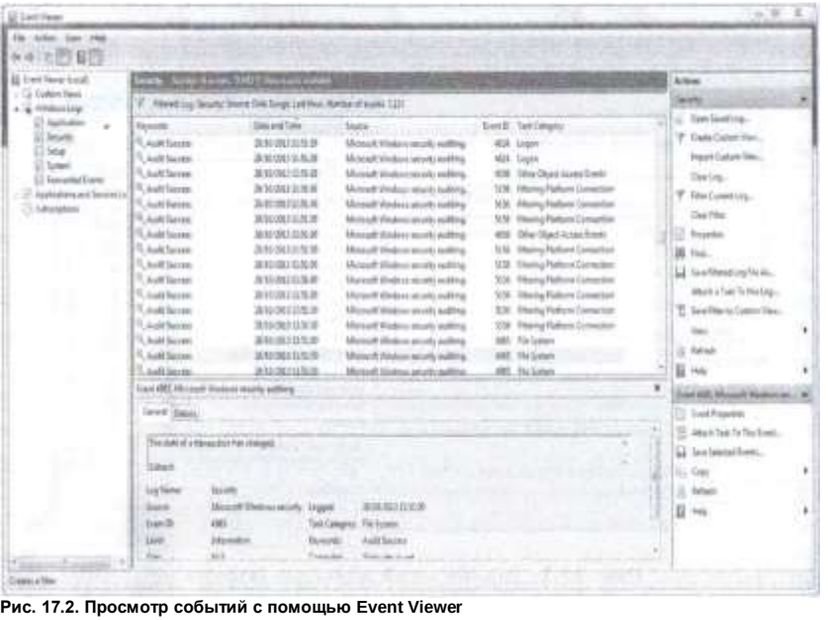


Рис. 17.2. Просмотр событий с помощью Event Viewer

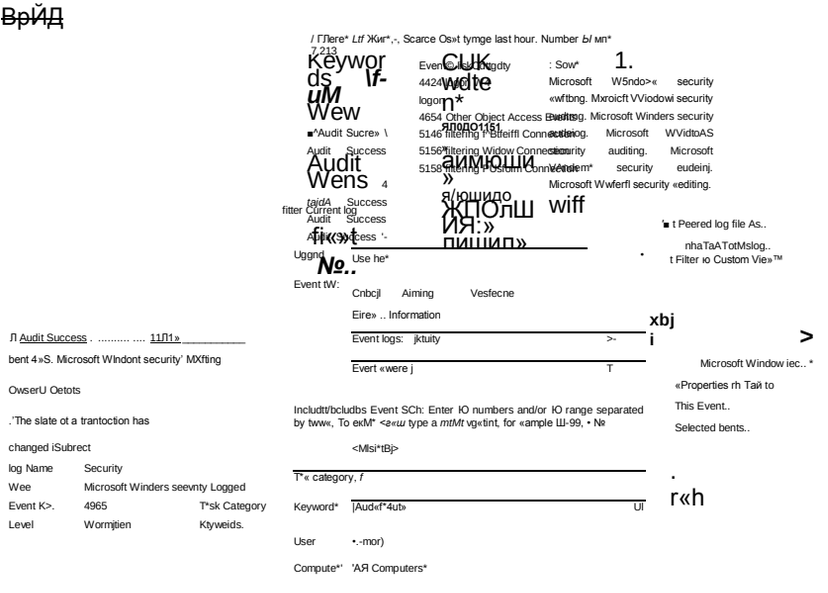


Рис. 17.3. Фильтрация событий журнала безопасности

Для повышения эффективности журнализации событий безопасности существуют несколько правил:

- активировать аудит нужно только для критических или чувствительных ресурсов;
- активировать аудит для операций Read и List (чтение и просмотр списка) с объектами нужно только в случае абсолютной необходимости, так как эти наиболее распространённые операции могут породить слишком большое количество событий малой значимости;
- аудит операции Execute (выполнить) для исполняемых файлов имеет смысл активировать только для системных утилит, которые могут использоваться при организации атаки;
- активировать аудит всех изменений системного каталога Windows, каталога Program Files, а также каталогов, важных для вашей работы.

Доступ к журналу безопасности должен быть ограничен, а сам журнал нужно периодически архивировать и создавать его резервные копии.

Аудит — это широкое понятие. До сих пор мы понимали его как журнализацию и анализ событий, связанных с различными аспектами безопасности системы. Существует также аудит безопасности в несколько ином смысле — как аудит настроек операционной системы, влияющих на степень безопасности системы. Цель такого аудита состоит в выявлении уязвимостей системы из-за ее некорректного конфигурирования.

Компания Microsoft разработала и свободно распространяет программу Microsoft Baseline Security Analyzer (MBSA), которая выполняет аудит настроек системы с точки зрения обеспечения ее безопасности. Эта программа создает отчет, где перечисляются найденные уязвимости в таких областях, как администрирование (например, проверяется, заблокирована ли учетная запись Guest, дающая доступ к компьютеру без предъявления пароля), достаточно ли стойкие пароли пользователей системы, правильно ли ведется администрирование веб-сервера I IS, если он установлен, и т. п.

Аудит событий безопасности в ОС Unix

В ОС Unix журнализация событий ведется многими сервисами, утилитами и приложениями; данные аудита событий помещаются в различные журналы. Журналы событий обычно помещаются в каталог /var/log, хотя отдельные приложения могут вести журналы и в других каталогах.

Типичный набор журналов, помещаемых в каталог /var/log различными сервисами и приложениями, выглядит следующим образом:

```
[rootSwwp log]#
-rw ----------1    root root      0 Oct 27  04:03 boot.log
-rv----------1     root root 109228 Oct 28  17:42 cron
-rw-r—r— 1         root root 146584 Oct 28  17:41 lastlog
drwxr-xr-x 2       root root   4096 Apr  3   2008 mail
-rw ----------1    root root 928269 Oct 28  17:42 maillog
-rw ---------- 1   root root    359 Oct 28  16:42 messages
-rw ----------1    root root   1221 Oct 28  17:42 secure
-rw ----------1    root root      0 Oct 27  04:03 spooler
-rw-rw-r— 1        rootutmp  37248 Oct 28  17:41 wtmp
drwx-------- 3     root root   4096 Feb 23   2012 xen
-rw-r--r-- 1 root root 51609 Oct 28 17:26 Xorg.O.log
```

Несколько журналов из этого списка ведется сервисом ***syslog,*** который собирает сообщения от других сервисов и помещает их в журнал в стандартной форме. Сообщения, относящиеся к событиям безопасности, сервис syslog помещает в журнал secure. Кроме того, интерес для анализа событий безопасности могут представлять записи файла messages, в которые попадают сообщения ядра системы. Заметим, что файлы сервиса syslog разрешено читать и изменять только суперпользователю root, остальные пользователи не могут даже прочитать содержимое этих журналов аудита.

Сервис syslog появился в системах Unix в 1980-е годы как часть проекта Sendmail и постепенно стал стандартом де-факто журнализации событий любого типа. Протокол syslog является теперь стандартом Интернета, описанном в RFC 5424.

Журнал secure, как и другие журналы сервиса syslog, состоит из текстовых записей, удобных для анализа, например:

```
Oct 28 17:04:38 wwp sshd[1299]: Invalid user adamb from 212.219.210.245 Oct 28 17:04:38 wwp
sshd[1300] : input_userauth_request: invalid user adamb Oct 28 17:04:42 wwp sshd[1300]: Connection
closed by 212.219.210.245 Oct 28 17:41:57 wwp sshd[1589]: Accepted password for victor from
193.62.83.186 port 55855 ssh2
Oct 28 17:41:57 wwp sshd[1589]: pam_unix(sshd:session): session opened for user victor by (uid=0)
Oct 28 17:42:02 wwp su: pam_unix(su:session): session opened for user root by victor(uid=500)
```

Из этих записей видно, что пользователь adamb пытался выполнить удаленный интерактивный вход с помощью сервиса ssh, но сессия не была успешной, так как такого пользователя нет в базе учетных данных системы (т. е. в файле passwd). Пользователь victor успешно вошел в систему с помощью ssh, и для него была открыта пользовательская сессия. Последняя запись говорит о том, что пользователь victor открыл для себя новую сессию под именем суперпользователя root (сообщения от утилиты su).

Сервис syslog можно настраивать, задавая уровень критичности сообщений, помещаемых в журнал. Для этого необходимо редактировать файл конфигурации сервиса /etc/syslog.conf. Параметры этого файла после

установки ОС выглядят так:

```
[rootSwwp etc]# cat syslog.conf
#    Log all kernel messages to the console.
#    Logging much else clutters up the screen.
#kern.* /dev/console
#    Log anything (except mail) of level info or higher.
#    Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none /var/log/messages
#    The authpriv file has restricted access, authpriv.*
/var/log/secure
#    Log all the mail messages in one place, mail.* -
/var/log/maillog
#    Log cron stuff cron.*
/var/log/cron
#    Everybody gets emergency messages *.emerg
*
#    Save news errors of level crit and higher in a special file, uucp.news.crit
/var/log/spooler
#    Save boot messages also to boot.log local7.*
/var/log/boot.log
```

Каждая строка файла syslog.conf задает способ обработки сообщений определенного типа. Например, строка authpriv.* /var/log/secure говорит сервису syslog, что сообщения об аутентификации пользователей (тип authpriv) любого уровня критичности (об этом говорит символ *) нужно помещать в файл /var/log/secure. В то же время сообщения этого типа не должны помещаться в файл messages, об этом говорит строка конфигурации

*.info;mail.none;authpriv.none;cron.none /var/log/messages

которая также диктует сервису syslog не помещать в этот файл почтовые сообщения (mail.none) и сообщения утилиты crone (crone.none). В то же время в файл messages направляются сообщения всех остальных типов с уровнем критичности info и выше.

Всего для сообщений syslog определено 8 уровней критичности:

- emerg (emergency, уровень 0): система стала неработоспособной, состояние «паники», о котором нужно предупредить всех пользователей;
- alert (alert, уровень 1): тревога, сообщение свидетельствует об очень серьезной проблеме в одной из основных систем ОС, требуется немедленная реакция администратора;
- crit (critical, уровень 2): критическое состояние одной из вторичных систем ОС, требуется немедленная реакция администратора;
- err (error, уровень 3): сообщение о некритичной ошибке, реакция может быть отложена на некоторое время;
- warning (warning, уровень 4): предупреждение о ситуации, которая в будущем может привести к ошибке;
- notice (notice, уровень 5): уведомление о необычной ситуации,

которая может создать проблему в будущем;

- info (informational, уровень б): информационное сообщение, возможно отчет об использвании ресурса или выполнении задачи;
- debug (debug, уровень 7): сообщение, полезное для отладки сервиса или программы, не используется в нормальных операциях. Используя инструкции файла syslog.conf, несложно изменить режим работы сервиса syslog и протоколировать больше или меньше сообщений в зависимости от ситуации.

Сервис syslog может направлять сообщения не только в локальные файлы, но и в файлы удаленного сервера. В больших корпоративных сетях часто выделяют специальный сервер для централизованной журнализации сообщений syslog остальных серверов системы, такое решение позволяет администратору сети просматривать все сообщения в одном журнале, что удобно. Для отправки сообщений на удаленный сервер на клиенте нужно вместо имена файла указать DNS имя сервера, например строка

```
authpriv.* Sdeimos.dev.ja.net
```

задает отправку всех сообщений аутентификации на сервер demos.dev.ja.net (символ © говорит о том, за ним следует имя удаленного сервера. На удаленном сервере также нужно добавить в файл конфигурации имя машины-клиента и файла-журнала для того, чтобы сообщения клиента принимались и журнализировались. Можно также направить сообщения syslog в другую программу, например программу анализа сообщений, с помощью какого-либо программного интерфейса, например с помощью механизма конвейера ОС.

Аудит логических входов пользователя можно проводить не только с помощью сервиса syslog, но и с помощью других сервисов, разработанных специально для этой цели. Популярны два таких сервиса — lastlog и last. Первый ведет журнал с тем же именем, а второй — журнал wtmp, оба журнала имеют специальный формат записей, поэтому читаются только с помощью команд lastlog и last.

Если lastlog дает информацию только о последнем входе каждого пользователя компьютера, то last ведет учет определенного количества последних входов в систему, так что, если кто-то входил в нее несколько раз за последние несколько дней, то last покажет все эти пходы:

```
[root®wwp log]# last
victor pts/5 193.62.83.186 Mon Qct 28 17:41 still logged in
root pts/4 :1.0 Mon Oct 28 17:36 still logged in
victor pts/1 deimos.dev.ja.ne Mon Oct 28 17:03 still logged in
root pts/1 :1.0 Fri Oct 18 17:34 — 16:41 (10+00:07)
victor pts/4 193.62.83.186 Thu Oct 3 15:08 -- 17:21 (02:12)
victor pts/4 193.62.83.186 Tue Sep 24 11:53 — 12:47 (00:54)
```

Ручная обработка журналов syslog и других сервисов ненадёжна и

отнимает много времени у администратора. Существует большое количество программ, автоматизирующих эту обработку за счет периодического просмотра заданных журналов и создания отчетов о событиях определенного типа. К таким программам относятся, например, logwatch, logcheck, logsurfer и этот список можно продолжить.

Классическая реализация протокола syslog передает сообщения на удаленный сервер с помощью протокола UDP в незашифрованном виде. Это, естественно, не очень хороший способ передачи, даже в пределах корпоративной сети. Существуют более современные реализации протокола syslog, например rsyslog, которые устраняют этот недостаток за счет использования протокола TCP и защищенных каналов SSL/TLS.

Вопросы к главе 17

1. Почему аудит событий локального доступа к файлам и каталогам файловой системы FAT невозможен?

а) потому что файловая система FAT не позволяет шифровать содержимое файлов;

б) потому что файловая система FAT не поддерживает механизм ACL;

в) потому что файловая система FAT не поддерживает многопользовательский режим.

2. Вы не активировали аудит логических входов в ОС Windows. Система аудита этой ОС:

а) не будет регистрировать логические входы пользователей в систему;

б) будет регистрировать только неуспешные попытки входа;

в) будет регистрировать только успешные попытки входа;

г) заблокирует вход пользователей в систему.

3. Вы активировали аудит доступа к объектам в политике аудита ОС Windows. Вы решили проверить правильность своих действий, открывая файл C:\notesl.doc, находящийся в разделе NTFS, и закрывая его в текстовом редакторе. Однако события, связанные с доступом к этому файлу, не были зафиксированы в журнале Security Log. Как вы думаете, почему это произошло?

а) вы проверили не тот журнал аудита;

б) вы не задали опцию аудита чтения данных для файла C:\notesl.doc;

в) текстовый редактор не поддерживает механизм ACL.

4. Чем отличается опция «Аудит входа в систему» от опции «Аудит входа по учетной записи» в ОС Windows?

а) это разные переводы одной и той же англоязычной опции;

б) опция «Аудит входа в систему» регистрирует входы в домен, а опция «Ауди1 входа по учетной записи» — в компьютер;

в) опция «Аудит входа в систему» регистрирует входы в компьютер, а опции «Аудит входа по учетной записи» — в домен.

5. Для программ какого вида имеет смысл активировать аудит операции «Выполнить» (Execute)?

а) для всех программ;

б) для всех системных утилит;

в) для всех программ, загруженных из Интернета;

г) для системных утилит, с помощью которых можно организовать атаку.

6. Чем использование сервиса аудита rsyslog ОС Unix/Linux предпочтительней использования сервиса syslog с точки зрения безопасности?

а) сервис rsyslog надежнее, так как передает данные на центральный сервер аудита с помощью протокола TCP;

б) сервис rsyslog быстрее, так как использует для передачи данных на центральный сервер аудита с помощью протокола UDP;

в) сервис rsyslog обеспечивает конфиденциальность и целостность данных аудита за счет использования защищенного канала IPSec;

г) сервис rsyslog обеспечивает конфиденциальность и целостность данных аудита за счет использования защищенного канала SSL.

7. В каком режиме из перечисленных ниже сервис аудита syslog будет фиксировать больше событий:

а) в режиме *.info;

б) в режиме *.crit;

в) в режиме *.alert.

8. Верно ли следующее утверждение: «Все события в ОС Unix/Linux журнали- зируются только сервисом syslog»?

а) да;

б) нет.

L8 *Стандарты безопасности и сертификация ОС и программных систем*

Стандарты безопасности операционных систем и программного обеспечения создают основу для сертификации ОС и программ, которая проводится независимыми организациями и дает администраторам сетей и пользователям некоторые объективные основания доверять свойствам безопасности этих продуктов. Кроме того, стандарты безопасности определяют перечень тех свойств и функций, наличие которых является необходимым для того, чтобы ОС или программная система обрабатывала информацию безопасным образом. Поэтому даже в том случае, когда ОС или программная система не является сертифицированной, стандарты безопасности играют полезную роль — они дают потребителю возможность проверить наличие описанных в стандартах свойств и самостоятельно оценить защищенность системы.

Оранжевая книга

Все, кто когда-либо занимался вопросами безопасности компьютерных систем, обязательно слышали об Оранжевой книге. Это самый заслуженный и популярный стандарт оценивает степень защищенности информационной системы, а также позволяет формализовать процедуру оценки, для чего в нем определяются формальные критерии отнесения системы к тому или иному классу безопасности. Формальное название этого стандарта: Министерство обороны США, ***Критерии оценки доверенных компьютерных систем (Department of Defence, Trusted Computer System Evaluation Criteria).***

mmsi

Рис. 18.1. Одно из изданий стандарта Министерства обороны США, известного под именем «Оранжевая книга»

Впервые этот стандарт был опубликован в 1985 году в составе так называемой

«Радужной» серии стандартов информационной безопасности, издававшейся в период с 1980 по 1990 гг. под эгидой Министерства обороны США. Все 37 книг этой серии имели обложки разнообразных цветов, одних только оттенков желтого цвета было несколько: желтая, светло-желтая, желто-коричневая, янтарная, оранжевая, ярко- оранжевая. Именно цвет обложки и дал второе, неформальное название «Оранжевая книга» стандарту «Критерии оценки доверенных компьютерных систем». Следует упомянуть также Красную книгу — еще один стандарт из «радужной» серии, который представляет собой интерпретацию Оранжевой книги для сетевых конфигураций.

Мы рассмотрим Оранжевую книгу более подробно, чем остальные стандарты, и сделаем это не только отдавая дань историческому значению данному стандарту, но и преследуя прагматические цели. Как в свое время появление модели OSI стало средством общения для сетевых специалистов, так публикация Оранжевой книги дала возможность разговаривать на одном языке специалистам в области безопасности (в основном — при обсуждении операционных систем). Новые стандарты, приходящие на смену Оранжевой книге, унаследовали интерпретацию многих основных терминов и важнейших концепций компьютерной безопасности в том виде, в котором они впервые были сформулированы при описании критериев оценки доверенных компьютерных систем. Знание основных положений Оранжевой книги полезно тем, что они показывают, на какие свойства и подсистемы ОС нужно обратить первоочередное внимание для усиления ее безопасности, какие механизмы защиты данных нужно применять, т. е. направляют усилия администратора безопасности в нужное русло.

Критерии сертификации вычислительных систем в области безопасности

Любые рассуждения о компьютерной безопасности неминуемо приводят к необходимости сформулировать требования к ней. Другими словами, надо ответить на вопрос: «Какие свойства должна иметь система, чтобы ее можно было назвать «безопасной?» Самым общим ответом на этот вопрос авторы стандарта считают следующее: ***Безопасная система*** должна контролировать доступ к информации, используя специальные механизмы, так, чтобы только авторизованные подобающим образом персоны или вычислительные процессы, работающие от их имени, могли читать, записывать, создавать или удалять информацию.

Таким образом, в качестве главного средства, обеспечивающего информационную безопасность, авторы стандарта рассматривают *управление доступом к информационным ресурсам.* Соответственно, интегральным критерием защищенности системы выступает качество механизмов управления доступом. В свою очередь критериями качеств механизмов доступа оцениваемой системы являются две группы

фебований:

- требования к функциональной полноте;
- требования гарантий качества реализации.

Представление о том, какой набор и характер функций (возможностей) должна иметь безопасная система, можно получить простым перечислением типичных функций управления доступом в существующих операционных системах широкого назначения. Однако к функциональности защитных механизмов систем специального назначения (встроенные системы, коммуникационные процессоры, управляющие системы и др.) могут быть предъявлены более строгие, специфические требования.

Заметим, что главная цель управления доступом — это обеспечение конфиденциальности и целостности, следовательно, именно на этих свойствах безопасной системы фокусируется данный стандарт, не *затрагивая проблему доступности.*

Предлагаемые в стандарте критерии предназначены для решения следующих задач:

- предоставить рекомендации производителям компьютерных систем относительно того, какие защитные механизмы они должны предусмотреть в своих новых коммерческих продуктах, чтобы производить широко доступные системы, удовлетворяющие требованиям безопасности (в основном предотвращению раскрытия данных) для чувствительных приложений;
- снабдить Министерство обороны США средствами оценки степени доверия к компьютерным системам, предназначаемым для обработки в защищенном режиме секретных и чувствительных данных;
- обеспечить основу для определения требований к безопасности в других стандартах.

В стандарте используется ряд специфических терминов: ***«Доверенная система»*** (*trusted system).* Для того чтобы подчеркнуть, что абсолютная безопасность является абстрактным, недостижимым понятием, авторы данного стандарта используют вместо терминов «безопасная система» термин «доверенная система», т. е. система, безопасности которой можно доверять в определённой степени.

«Требования доверия» (*trust requirements)* — требования, определяющие некоторую степень доверия к безопасности системы.

«Политика безопасности». Данный термин в Оранжевой книге используется в более узком смысле, чем обычно, и означает принципы и подходы к организации управления доступом (сравните с опре

делением этого термина в разделе «Политика безопасности» данной книги).

«Доверенная вычислительная база» (*ДВБ*) (***Trusted Computing Base, TCB***) — собокупность программных и аппаратных механизмов информационной системы, отвечающих за реализацию политики безопасности. Таким образом, оценка степени безопасности/доверия системы может быть сведена к оценке ее компактной и обозримой доверенной вычислительной базы. ДВБ является частью, хотя и принципиально важной, вычислительной системы.

Шесть базовых требований

Как было сказано выше, главным требованием к доверенной системе является качественно исполненное управление доступом. Из этого общего утверждения могут быть извлечены шесть требований, четыре из которых имеют дело с функциональностью, т. е. с тем, что именно должно быть сделано для того, чтобы контролировать доступ к информации, а оставшиеся два — с гарантиями, т. е. с возможностью удостовериться в том, что полученная в результате система действительно заслуживает доверия. Функциональные требования в свою очередь подразделяются на требования к политике безопасности и требования к подотчётности.

Требования к политике.

Первое требование — *политика безопасности.* Безопасная система должна иметь тщательно проработанную и хорошо описанную политику безопасности. Объекты и субъекты доступа должны быть идентифицированы, а также должен быть определён набор правил, на основании которых система должна решать, может ли данный субъект получить доступ к данному объекту. В стандарте кратко описываются Два вида политики безопасности, соответствующие двум основным способам контроля доступа: мандатного и дискреционного (мы рассмотрели их в части 2 «Базовые технологии компьютерной безопасности»), *Мандатный* подход к определению прав доступа заключается в том, что все информационные ресурсы (объекты) делятся на группы в зависимости от степени секретности, а все субъекты — на группы в соответствии с уровнем допуска к секретной информации. Управление доступом осуществляется сравнением уровня секретности объекта с уровнем допуска субъекта. То есть субъекты в принципе не имеют возможности влиять на принятие решения. *Дискреционный (избирательный)* метод управления доступом предполагает, что субъект- владелец информационного ресурса по своему усмотрению разрешает или запрещает выполнять определенные операции с данным ресурсом другим субъектам, явно указывая их имена-идентификаторы.

Второе требование — *маркировка.* ***Для*** того чтобы система могла контролировать доступ к хранящейся в компьютере информации в соответствии с правилами мандатной политики безопасности, с каждым

объектом в системе должна быть связана *метка безопасности.* Метка содержит информацию об уровне чувствительности (классе секретности) объекта или уровне допуска субъекта. Уровни секретности и уровни доступа организованы иерархически. Помимо уровней секретности и допуска метка безопасности как объекта, так и субъекта содержит перечень ***категорий,*** описывающих предметную область, к которой относится информация. Множество категорий является неупорядоченным, поэтому в стандарте используется термин «неиерархические категории».

Третье требование — *идентификация.* Каждый индивидуальный субъект (пользователь) должен быть идентифицирован. При каждой попытке доступа должны быть проанализированы данные о том, кто пытается получить доступ к информации и к какому классу информации у него есть доступ. Эта идентификационная и авторизационная информация должна надежно поддерживаться системой и ассоциироваться с каждым субъектом системы, выполняющим какое-либо действие, затрагивающее безопасность.

Четвертое требование — *подотчетность* (accountability). Система должна выборочно сохранять (протоколировать) данные, фиксирующие события в системе, с тем, чтобы в дальнейшем у ответственных лиц была возможность отследить действия, затрагивающие безопасность системы. Возможность селективной записи событий необходима для того, чтобы минимизировать стоимость хранения протоколируемых данных, а также повысить эффективность их анализа (аудита). Протоколированные данные должны быть надёжно защищены от модификации и разрушения неавторизованными субъектами.

Этими четырьмя требованиями стандарт устанавливает обязательную индивидуальную ответственность субъектов, для реализации которой система должна поддерживать функции аутентификации, авторизации и аудита (система должна собирать и сохранять информацию, достаточную для отслеживания тех действий пользователя, которые могут повлиять на безопасность системы).

Пятое требование — *гарантированность.* В компьютерной системе должны быть определены программные и аппаратные механизмы, которые можно было бы независимо оценивать для получения гарантий того, что система действительно удовлетворяет всем четырем требованиям, перечисленным выше. То есть должен быть идентифицирован и унифицирован, а также тщательным образом описан набор программно-аппаратных управляющих механизмов, которые исполняют функции политики безопасности, маркировки, идентификации и

подотчетности. Обычно такого рода механизмы встраиваются в операционную систему и выполняют указанные функции в защищенном режиме. Гарантии того, что система реализована корректно, дает тщательное тестирование доверенной вычислительной базы, т. е. всех частей системы, значимых с точки зрения безопасности.

Шестое требование — ***непрерывная защита*** (continuous protection). Доверенные механизмы должны быть постоянно защищены от неавторизованного вмешательства в их работу и неавторизованных изменений. Ни одна система не может считаться безопасной, если сами механизмы, реализующие политику безопасности, не защищены от подделок и других неавторизованных воздействий. То есть защита системы должна охватывать весь ее жизненный цикл: проектирование, производство, испытания. В данном стандарте этот термин используется в более узком смысле, чем обычно, и означает способ контроля доступа, продажу и техническую поддержку, так чтобы на каждом из этапов исключить угрозы злонамеренного вмешательства в эти механизмы, такого, например, как наделение системы недекла- рированными нелегальными возможностями.

Уровни и классы безопасности

В стандарте определены четыре уровня безопасных систем: D, C, B и A, образующих иерархию от самого низкого уровня D (неудовлетворительная безопасность) к самому высокому — A. Уровни C и B подразделяются на подуровни (классы), так что результирующая иерархия выглядит следующим образом: Cl, C2, Bl, B2, B3, A1.

Каждый класс характеризуется четырьмя наборами критериев/ требований, представляющих:

- политику безопасности;
- подотчетность;
- гарантированность;
- документированность.

При перемещении от нижнего к вышестоящему классу происходит аккумулирование требований. То есть каждый класс должен удовлетворять всем требованиям, предъявляемым к предыдущему классу, а также дополнительным требованиям, повышающим статус безопасности данного класса. Например, система класса B3 должна удовлетворять всем требованиям, предъявляемым к системам класса B2 или, что равнозначно, требованиям классов Cl, C2, B1 и B2 и сверх того, ряду дополнительных требований, которые становятся обязательными, начиная с этого уровня.

Системы класса C1 должны удовлетворять следующим критериям:

Политика безопасности. Доверенная вычислительная база систем данного класса должна реализовать дискреционный (избирательный) способ управления доступом именованных пользователей к именованным объектам.

Подотчетность. Доверенная вычислительная база должна требовать, чтобы пользователи сообщали ей свои идентификаторы прежде, чем они начнут выполнять какие-либо другие операции, которые могут потребовать санкционирования со стороны ДВБ. ДВБ должна использовать какой-либо защитный механизм (например, пароли), чтобы аутентифицировать пользователей. Все данные, используемые при аутентификации, должны быть надежно защищены от доступа неавторизованных пользователей.

Гэ*рантированность.* ДВБ должна поддерживать программную область для собственного выполнения, защищенную от внешнего вмешательства или подделки (например, модификацией ее кода или структур данных). Система должна быть снабжена программными и/или аппаратными средствами, позволяющими периодически проверять правильность функционирования аппаратных элементов, относящихся к доверенной вычислительной базе. Защитные механизмы должны быть тщательно протестированы на предмет их соответствия технической документации. Тестирование должно демонстрировать, что отсутствуют какие-либо явные возможности обойти или другим образом преодолеть защитные механизмы безопасности в ДВБ.

Документированность. Руководство пользователя должно включать описание защитных механизмов ДВБ, информацию о том, как их использовать, а также о том, как они могут взаимодействовать. Руководство системного администратора должно содержать предостережения о функциях и привилегиях, которые необходимо контролировать при работе защитных средств. Разработчик системы должен передать специалистам, выполняющим оценку системы, все документы, включая план и процедуры тестирования, которые показывали бы, как проводилось тестирование защитных механизмов и какие результаты были получены. Наконец, обязательно должна быть в наличии проектная документация.

Системы класса C2 должны удовлетворять всем требованиям, предъявляемым к системам класса C1, а также следующим дополнительным критериям:

Политика безопасности. Так же, как и в системах класса C1, доверенная вычислительная база системы класса C2 должна определять и контролировать доступ именованных пользователей к именованным объектам (например, файлам и программам). Однако ее исполнительный механизм (например, на основе списков доступа) должен приводить в исполнение более тонко гранулированные правила дискреционного доступа. Так, ДВБ должна предоставлять пользователям возможность устанавливать правила совместного использования указанных объектов именованными субъектами и/или группами субъектов, а также должна дать им средства ограничивать дальнейшее распространение прав доступа. Управляющий механизм дискреционного доступа должен либо по явному указанию пользователя, либо по умолчанию обеспечивать защиту каждого

объекта от несанкционированного доступа. Разрешения на доступ к объекту для пользователей, их еще не имеющих, должны назначаться только авторизованными пользователями.

Важным дополнительным требованием к средствам управления доступом является требование безопасного повторного использования объектов, которое состоит в следующем. При завершении использования любого объекта должны ликвидироваться все информационные следы его предыдущего использования, должны очищаться соответствующие области оперативной и дисковой памяти. Никакая информация, в том числе представленная в зашифрованном виде, не должна стать доступной следующему субъекту, использующему объект.

Подотчётность. Помимо требований данной группы, сформулированных для С1, в системах класса С2 должна быть обеспечена возможность индивидуальной подотчётности, т. е. возможность селективного аудита затрагивающих безопасность событий для каждого конкретного пользователя. Доверенная вычислительная база должна создавать, поддерживать и защищать от модификации, разрушения или иного неавторизованного доступа данные о попытках доступа к объектам, которые она защищает. Чтение данных аудита должно быть разрешено только тем лицам, которые авторизованы выполнять аудит. К событиям, подлежащим регистрации, относятся использование механизмов идентификации и аутентификации, перемещение объектов в пользовательское адресное пространство (например, открытие файла, запуск программы), удаление объектов, операции, проводимые оператором, администратором или ответственным за безопасность вычислительной системы. Запись о каждом зарегистрированном событии должна идентифицировать пользователя, тип события, дату и время события, успешность или неуспешность события. Для событий идентификации и аутентификации нужно указывать источник запроса, например идентификатор терминала. Если событие связано с действиями над объектом, то запись должна включать имя объекта.

Гарантированность. В дополнение к требованиям, предъявляемым к гарантированности систем класса С1, от систем класса С2 требуется, чтобы тестирование подтвердило отсутствие явных дефектов в механизмах изоляции ресурсов и защиты данных аудита и аутентификационных данных.

Документирование. В соответствии с дополнительными требованиями к аудиту в руководстве администратора для систем класса С2 должны присутствовать процедуры проверки и поддержки файлов с данными аудита, а также описание детальной структуры записей аудита для каждого типа событий.

Системы с наивысшим уровнем безопасности А и В получают свой статус за счет более развитой политики безопасности, качественной архитектуры, а также за счет более строгого анализа на этапе проектирования.

Системы класса В1 должны удовлетворять всем требованиям, предъявляемым к системам класса С2, а также следующим дополнительным критериям.

Политика безопасности. Кроме дискреционного доступа, доверенная вычислительная база системы класса В1 должна поддерживать мандатный способ управления доступом. Для этого в нее должны быть встроены функции назначения меток безопасности всем субъектам и хранимым объектам системы. Метки объектов характеризуют уровень секретности содержащейся в них информации, а метки субъектов — уровень допуска субъектов к классифицированной информации. ДВБ должна поддерживать как минимум два уровня секретности.

При любой попытке доступа должны проверяться следующие условия:

- субъект может читать объект только, если уровень допуска данного субъекта выше или равен уровню секретности данного объекта, а также список неиерархических категорий в метке субъекта включает все категории объекта;
- субъект может записывать в объект, если уровень допуска данного субъекта ниже или равен уровню секретности данного объекта, список категорий объекта субъекта содержатся в списке категорий объекта.

Подотчетность. Системный администратор должен иметь возможность проводить аудит селективно, основываясь на персональном идентификаторе и уровне секретности объекта.

Гарантированность. Важным средством систем класса В1 является использование структурных методов защиты. Доверенная вычислительная база должна поддерживать взаимную изоляцию процессов за счет предоставления им отдельных адресных пространств. Специалисты, полностью владеющие информацией о том, как устроена ДВБ, должны подвергнуть тщательному анализу и тестированию ее проек- шую документацию, исходные и объектные коды. Все обнаруженные

дефекты должны быть устранены, при этом необходимо удостоверить-ся, что в результате в систему не были внесены новые ошибки.

Документированность. Руководство администратора должно со-держать описание функций оператора и администратора, с помощью которых они могут изменять характеристики безопасности пользователя. Документация должна содержать рекомендации о том, как эффективно использовать средства безопасности системы, как они взаимодействуют, каким образом можно сгенерировать новую доверенную вычислительную базу.

Системы класса B2 должны удовлетворять всем требованиям, предъявляемым к системам класса B1, а также следующим допол-нительным критериям

Политика безопасности. ДВБ должна базироваться на ясно оп-ределённой и документированной *формальной модели политики без-опасности*,* в соответствии с которой порядок применения дискреционного и мандатного контроля доступа в системах класса B1 должен быть распространен на все субъекты и объекты вычислительной системы. ДВБ должна поддерживать метки не только для объектов и субъек1ов ДВБ, но и для всех ресурсов вычислительной системы (например, для постоянной памяти, карт памяти, съемных жестких дисков), которые прямо или косвенно доступны субъектам, внешним по отношению к ДВБ. ДВБ должна поддерживать назначение минимального и максимального уровней секретности для каждого из подключенных внешних устройств.

Подотчетность. Доверенная вычислительная база должна под-держивать доверенный (защищенный) коммуникационный канал, который должен использоваться исключительно по инициативе пользователя во время его начальной процедуры логического входа и аутентификации. Для всех событий, связанных с использованием доверенных каналов обмена, должна поддерживаться функция аудита.

Гарантированность. ДВБ должна быть тщательно структурирована с выделением функционально независимых модулей и критически важных с точки зрения защиты элементов. По возможности структурным модулям должны присваиваться атрибуты «только для чтения» или «только для записи». Каждый отдельный модуль должен наделяться тем минимальным уровнем привилегий, который необходимым для его выполнения. Интерфейс ДВБ должен быть четко определен, а ее архитектура и реализация должны быть таковы, чтобы ДВБ могла быть подвергнута еще более тщательному тестированию, чем ДВБ систем класса B1.

Формальные модели безопасности управления доступом рассматриваются в главе б.

Системный архитектор должен провести тщательный поиск име- м |щихся в системе скрытых каналов памяти и оценить (с помощью фи-жческих измерений или инженерных расчетов) максимальную пропускную

способность каждого обнаруженного канала. *Скрытый канал* — 1 го любой программно-реализуемый способ передачи информации от одного процесса другому, реализуемый в обход специально предусмотренных для этой цели законных безопасных механизмов обмена.

)та проблема более подробно рассматривается в главе 19 в разделе (крытые коммуникации и скрытые каналы».

Механизмы ДВБ должна быть протестированы с тем, чтобы показать ее относительную устойчивость к попыткам проникновения. И процессе разработки и сопровождения доверенной вычислительной базы должна использоваться система конфигурационного управления, обеспечивающая контроль изменений в описательных спецификациях верхнего уровня, иных архитектурных данных, реализационной документации, исходных текстах, работающей версии объектного кода, тестовых данных и документации.

Документирование. Документация должна включать план тести-рования, описание и результаты тестовых процедур, которым были подвержены механизмы безопасности системы. Результаты тестирования должны показывать эффективность мер по уменьшению пропускной способности скрытых каналов памяти. Должен быть предос- швлен перечень всех видов протоколируемых событий, которые могут косвенно указывать на использование выявленных скрытых каналов. Для выявленных скрытых каналов, использование которых не может быть отслежено протоколированием событий, должна быть указана пропускная способность.

Системы класса B3 должны удовлетворять всем требованиям, предъявляемым к системам класса B2, а также следующим допол-нительным критериям.

Политика безопасности. Механизм дискреционного доступа должен быть дополнен возможностью определять для каждого именованного объекта список доступа, т. е. перечень именованных субъектов или групп субъектов с указанием действий, которые им разрешены по отношению к данному объекту. Более того, для каждого именованного объекта должна быть предусмотрена возможность определения перечня именованных субъектов или групп субъектов, для которых доступ к данному объекту запрещен.

Подотчетность. ДВБ должна поддерживать доверенный канал связи (не путать со скрытым каналом) между ней и пользователями (например, при логическом входе или при изменении уровня допуска пользователя). Взаимодействие через этот доверенный канал связи должно активироваться исключительно по инициативе пользователя

ДВБ. ДВБ должна включать в себя механизм, способный регистрировать и аккумулировать события, имеющих отношение к безопасности системы. При превышении порога допустимости количества таких событий этот механизм должен немедленно оповестить администратора по безопасности о возможной угрозе безопасности системы. Если же накопление таких событий продолжится, система должна предпринять действия по пресечению этих событий.

Гарантированность. ДВБ должна иметь функционально полный и концептуально простой защитный механизм с точно определенной семантикой. В штате должна быть предусмотрена особая позиция — системный администратор по безопасности. Администратор может получить роль системного администратора по безопасности только в результате явных, запротоколированных службой аудита обращений к системе. Администратор по безопасности не должен выполнять действий, не связанных с безопасностью, кроме тех, которые ему необходимы для выполнения его основных функций обеспечения безопасности. Во время тестирования не должно быть найдено никаких архитектурных ошибок и лишь небольшое число поддающихся исправлению ошибок реализации, что давало бы основание для уверенности в том, что в системе скорее всего осталось мало ошибок. В результате тестирования доверенная вычислительная база должна показать устойчивость к проникновениям.

Документированность. Документация должна включать описание процедур, гарантирующих то, что система стартовала в режиме защиты. Должны быть включены также описание процедур, позволяющих восстанавливать работу системы без ослабления защиты после возникновения сбоев в результате каких-либо нарушений в работе системы. В документации должны быть также представлены результаты анализа скрытых каналов.

Уровень безопасности А характеризуется использованием формальных методов верификации защиты. Системы данного уровня снабжаются обширной документацией, которая призвана демонстрировать, что их ДВБ удовлетворяет требованиям безопасности во всех аспектах: архитектурном, проектном и реализации.

Системы класса А1 должны удовлетворять всем требованиям, предъявляемым к системам класса В3. Из числа дополнительных критериев авторы стандарта выделяют следующие пять основных требований.

1. Формальная модель политики безопасности должна быть ясно определена и документирована, включая математическое доказательство того, что модель согласуется с ее аксиомами и является достаточной, чтобы поддерживать политику безопасности.

2. Должны быть разработаны формальные спецификации верхнего уровня ДВБ, которые должны включать абстрактные определения функций, выполняемые доверенной вычислительной базой, а также абстрактные определения аппаратных и микропрограммных механизмов, используемых для поддержки раздельных областей памяти для выполнения процессов.

3. Должно быть показано, что эти формальные спецификации верхнего уровня ДВБ полностью согласуются с моделью. Это должно быть сделано формальными методами там, где это возможно (т. е. там, где существуют соответствующие средства верификации), и неформально в противном случае.

4. Должно быть неформально показано, что все компоненты реализации ДВБ (т. е. аппаратные, микропрограммные и программные средства) соответствуют ее формальным спецификациям верхнего уровня.

5. Для выявления и анализа скрытых каналов должны быть использованы формальные методы. Неформальные методы могут быть использованы для обнаружения ***скрытых временных каналов.*** Решение о дальнейшем существовании выявленных скрытых временных каналов должно быть обосновано.

Если максимально кратко сформулировать основные требования, предъявляемые к системам в Оранжевой книге, то получим следующие характеристики уровней безопасности:

A — верифицированная защита (verified protection);

B — мандатная защита (mandatory protection);

C — дискреционная защита (discretionary protection);

D — минимальная безопасность (minimal security).

Приведенное выше описание требований к системам очень близко соответствуют тексту стандарта, так что у читателя есть возможность получить общее представление о принципах сертификации систем по критериям безопасности Оранжевой книги. В настоящее время на смену данному стандарту пришли другие, но это не умаляет заслуг Оранжевой книги, заложившей принципиальные основы оценки безопасности информационных систем, справедливые и по сей день.

Оранжевая книга учитывает безопасность отдельно рассматриваемой системы, а сетевые аспекты безопасности рассматриваются в Красной книге, имеющей название «Интерпретация доверенных сетей» (Trusted Network Interpretation). Как следует из названия этого стандарта, в нем интерпретируются основные положения Оранжевой книги применительно к операциям передачи информации по сетям. Например, вводится понятие сетевой доверенной вычислительной базы (Network Trusted Computing Base), которое дополняет понятие доверенной вычислительной базы за счет средств безопасности сетевых коммуникаций.

Стандарт «общие критерии»

Общая структура и цели

Оранжевая и Красная книги хороши для понимания общего подхода к

стандартизации и сертификации операционных систем в отношении безопасности, но эти стандарты уже не являются сегодня общепринятым практическим руководством при оценке и сертификации безопасности информационных систем по нескольким причинам:

- это не международные стандарты, а стандарты Министерства обороны США;
- они разработаны уже достаточно давно;
- они не обладают той гибкостью, которая необходима для применения их к различным программным продуктам, а не только к операционным системам.

Международное информационное сообщество, понимая эти ограничения Оранжевой и Красной книг, разработало стандарт, известный под кратким (неофициальным) названием ***«Общие критерии».*** который в определенной степени преодолевает указанные ограничения: это международный стандарт, это постоянно обновляющийся стандарт и это стандарт, обладающей большой гибкостью, позволяющей оценивать и сертифицировать программные продукты различных классов.

Строго говоря, существует два варианта этого стандарта:

- стандарт Международной организации по стандартизации (ISO) «Критерии оценки безопасности информационных технологий (ISO/IEC 15408-1/2/3 — Information technology — Security techniques — Evaluation criteria for IT security);
- стандарт «Общие критерии оценки безопасности информационных технологий» (Common Criteria for Information Technology Security Evaluation) международного консорциума стран, подписавших «Соглашение о признании Общих критериев».

Эти стандарты очень близки, их можно даже считать одним стандартом, разница между ними состоит в том, что стандарт ISO является обычным «статическим» документом, в то время как стандарт международного консорциума — это «живой» документ, который постоянно обсуждается, корректируется и, что более важно, является теоретической основой международной сети сертификационных центров, которые занимаются практической работой по оценке и сертификации разнообразных информационных продуктов — операционных систем, файерволов, систем аутентификации, цифровой подписи и т. п. Процесс сертификации осуществляется в соответствии с документом-компаньоном «Общая методология оценки безопасности информационных систем». Консорциум «Общие критерии» (будем его для краткости называть так, поскольку другого названия это сообщество не имеет) публикует свои документы, а также список сертифицированных продуктов на сайте http://www.commoncriteriaportal.org.

Основной целью консорциума «Общие критерии» является исключение дублирования процесса сертификации информационных продуктов несколькими странами. Страны, подписавшие «Соглашение о признании Общих критериев», договорились о взаимном признании сертификатов безопасности, выданным любым из центров сертификации, находящимся в стране-участнице и перечисленным в списке действующих центров сертификации консорциума. Не все страны, подписавшие соглашение, имеют сертификационные центры на своей территории, поэтому страны делятся на авторизующие (сертифицирующие) и потребляющие (признающие сертификаты авторизующих стран).

На момент написания данной книги консорциум состоял из 17 авторизующих стран и 9 потребляющих (Россия пока это соглашение не подписала), а список сертифицированных продуктов насчитывал 1852 системы из них 102 — операционные системы.

Стандарт ISO «Критерии оценки безопасности информационных технологий» можно считать мгновенным «слепком» постоянно развивающего стандарта «Общие критерии», хотя он не совсем статичен и обновляется раз в 4-5 лет. Так, на момент написания книги действующим стандартом ISO была версия 15408-2009, то есть 2009 года, в то время как предыдущая версия была опубликована в 2005 году. В то же время действующей версией «Общих критериев» была версия «September 2012, Version 3.1 Revision 4», т. е. намного более «свежая».

В РФ действует ГОСТ Р ИСО/МЭК 15408-1-2008 «Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий», который, как написано в его вводной части, «идентичен международному стандарту ИСО/МЭК 15408-1:2005», т. е. повторяет (точнее — переводит на русский язык) версию стандарта ISO от 2005 года.

Далее мы достаточно компактно рассмотрим основные положения «Общих критериев».

Стандарт состоит из трех частей.

- ***Первая часть*** является вводной, она описывает цели стандарта,

термины и общую модель построения набора релевантных критериев безопасности ***оцениваемого объекта*** (СЮ).

- ***Вторая часть*** «Функциональные требования безопасности» описывает свойства и механизмы, характеризующие безопасную систему. Это тип требований является аналогом требований политики безопасности из Оранжевой книги.
- ***Третья часть*** «Требования доверия к безопасности» содержит описание требований, выполнение которых порождает доверие к тому, что функции безопасности оцениваемой системы реализованы качественно и могут успешно противостоять предполагаемым угрозам. В каком-то приближении этот тип требований соответствует требованиям гарантированности и документирован- ности из Оранжевой книги.

«Общие критерии» не описывают самой методики оценки ОО на соответствие функциональным требованиям и требованиям доверия. Этот пробел восполняет документ «Общая методология оценки безопасности информационных технологий». Сертификационные центры должны руководствоваться этим документом, чтобы результаты сертификации различных центров были сопоставимыми. Стандарт ISO «Критерии оценки» (и, соответственно, российский ГОСТ 15408) ограничивается повторением только трех частей «Общих критериев», оставляя методологию оценки на усмотрение оценивающих органов и лиц.

Функциональные требования безопасности

Функциональные требования безопасности, описанные во второй части «Общих критериев», представляют собой обширную библиотеку/каталог функциональных механизмов безопасности, которые используются в информационных системах для защиты от угроз. Это — механизмы аутентификации, авторизации, аудита, разобранные на крупные и мелкие составляющие таким образом, чтобы из них можно было сконструировать набор функций ОО, который производитель ОО намерен повергнуть сертификации, т. е. всесторонней оценке. Гибкость конструирования функциональных требований является отличительной чертой философии «Общих критериев», которая резко контрастирует с предопределенным подходом Оранжевой книги, где все функции безопасности жестко закреплены за каждым уровнем безопасности.

Функциональные требования безопасности имеют иерархическую структуру, они разбиты на три уровня: классы требований, семейства требований, компоненты требований.

Всего определено 11 классов функциональных требований:

- FAU — аудит;
- FCO — коммуникации (неотказуемость коммуникаций «отправитель-получатель»);

- FCS — криптографическая поддержка (обслуживает управление ключами и операциями);
- FDP — защита данных пользователя;
- FIA — идентификация и аутентификация;
- FMT — администрирование безопасности;
- FPR — приватность;
- FPT — защита функций безопасности объекта оценки;
- FRU — использование ресурсов (например, отказоустойчивость);
- FTA — доступ к объекту оценки (управление сеансами работы пользователей);
- FTP — доверенный маршрут/канал.

Каждый класс включает несколько семейств требований, а каждое семейство может состоять из нескольких компонент.

Например, класс FAU включает б семейств требований, относящихся к функциональности системы аудита ОО. Среди них имеется семейство FAILARP, предусматривающее автоматический ответ ОО на обнаружение события, нарушающего безопасность ОО. В этом семействе имеется только один компонент FAU_ARP.l, который требует, чтобы ответ ОО состоял в выполнении некоторых автоматических действий из списка, точный состав которого не определен. Этот неопределенный список является примером параметрической настройки функциональных требований «Общих критериев», когда значение параметра определено не стандартом, а производителем при составлении так называемого ***задания по безопасности*** (Security Target, ST).

Задание по безопасности является главным (и индивидуальным) документом, который производитель предоставляет вместе со своей системой в сертификационный центр. Мы еще вернемся к этому документу, а пока скажем, что одним из обязательных разделов этого документа является раздел «Требования к безопасности», куда и должны быть помещены функциональные требования, выбранные производителем для сертификации и имеющие конкретные параметры, если они не определены во второй части «Общих критериев».

Требования доверия к безопасности

Требования доверия к безопасности изложены в третьей части «Общих критериев». Эти требования выражают философию «Общих критериев», которая заключается в том, что доверие к заявленными производителем функциям безопасности, которыми (якобы) обладает его система, может появиться только в результате тщательной и всесторонней оценки этих функций. ***«Доверие через оценку»*** — это основной девиз «Общих критериев». Поэтому требования доверия стараются оценить все аспекты ОО — его цели безопасности, спектр и серьезность угроз, а также соответствие функций безопасности, имеющихся у ОО, существующим угрозам, качество процесса проектирования системы, качество процессов

поддержки жизненного цикла системы, включая ее руководства администратора и пользователя, а также качество процессов тестирования функций безопасности 00.

Таким образом, в соответствии с «Общими критериями» тестирование функций безопасности не является единственным типом процедур сертификации 00. Предполагается, что производитель оцениваемого объекта предоставляет на сертификацию свидетельства, характеризующие процесс его проектирования и поддержки жизненного цикла, а сертифицирующий центр должен провести оценку этих свидетельств.

Как и функциональные требования безопасности, требования доверия к безопасности имеют иерархическую структуру, включающую классы, семейства и компоненты требований.

Определено 8 классов требований:

- APE — оценка профиля защиты;
- ASE — оценка задания по безопасности;
- ADV — оценка проектирования 00;
- AGD — оценка руководств 00 (администратора и пользователя);
- ALC — оценка поддержки жизненного цикла;
- ATE — тестирование 00;
- AVA — оценка уязвимостей;
- ACO — композиция (оценка составных 00, включающих другие 00).

Названия большинства классов говорят сами за себя, и они прямо соответствуют основным областям, определяющим доверие к безопасности системы.

Пользуясь каталогом требований доверия к безопасности, производитель системы может сконструировать индивидуальный набор требований, который, на его взгляд, приведет к сертификации его системы с достаточным уровнем доверия к ее безопасности и в то же время соблюдет баланс между уровнем доверия и затраченными усилиями на сертификацию — понятно, что для более высокого уровня доверия нужно предоставить больше свидетельств и провести больше тестов.

Ввиду важности получения сравнимых результатов сертификации для различных продуктов разработчики «Общих критериев» определили семь ***стандартных*** наборов критериев доверия к безопасности, которые называются ***оценочными уровнями доверия*** (Evaluation Assurance Levels, EAL). Уровень EAL 1 обеспечивает самую низкую степень доверия к свойствам безопасности оцениваемого объекта, так как он включает минимальное число (а именно, 13) компонент требований доверия к безопасности. Самую высокую степень доверия обеспечивает уровень EAL 7, который включает уже 26 компонент почти из всех семейств, определенных в части 3 «Общих критериев», причем эти компоненты более строгие, чем соответствующие компоненты из тех же семейств, используемые при определении уровня EAL 1. Каждый более высокий

уровень включает все компоненты, входящие в предыдущий уровень, и добавляет к ним свои компоненты — что аналогично способу построения уровней безопасности Оранжевой книги. Можно добавлять к некоторому уровню EAL компоненты безопасности, не входящие в этот уровень, но удалять компоненты из стандартного набора уровня нельзя.

Несмотря на предоставляемую стандартом возможность индивидуального определения набора компонент доверия к безопасности, производители систем предпочитают пользоваться наборами стандартных оценочных уровней безопасности. Если вы зайдете на сайт http://www.commoncriteriaportal.org, то на страницах сертифицированных продуктов вы найдете, что практически все они были сертифицированы по стандартным уровням доверия EAL (как правило, от EAL2 до EAL 5) с добавлением одного-двух-трех компонент «сверх нормы».

Задание по безопасности и профили защиты

Как уже было отмечено, основным документом, который производитель оцениваемого объекта должен разработать и представить в сертификационный центр, является ***Задание по безопасности.*** Его структура и содержание каждого раздела определено семействами и компонентами требований доверия к безопасности класса ASE. Руководствуясь этими требованиями, сотрудники сертификационного центра должны оценить качество Задания по безопасности данного 00, и эта оценка является одним из этапов сертификации.

Для некоторых наиболее распространенных типов информационных систем, например для универсальных операционных систем, файерволов, систем аутентификации и некоторых других, членами консорциума «Общих критериев» разработаны ***Профили защиты*** (Protection Profiles). Профиль защиты является своего рода типовым «Заданием по безопасности» для определенного типа информационной системы, он содержит некоторые базовые описания, обоснования и требования, которые релевантны для всех 00 этого типа. Структура и содержание разделов «Профиля защиты» определяется семействами и компонентами требований доверия к безопасности класса APE, и они весьма

близки к структуре «Задания по безопасности», определяемой классом требований ASE. Предполагается, что разработчик «Задания по безопасности» может сослаться на соответствующий «Профиль безопасности» или даже несколько профилей безопасности в своем документе и описать только отличия своего «Задания по безопасности» от «Профиля защиты».

Рассмотрим структуру «Задания по безопасности».

Введение (семейство ASEJNT). В этом разделе должно содержаться описание 00: объект должен быть уникально идентифицирован, т. е. должно быть указано имя продукта и его версия, например Red Hat Enterprise Linux on IBM Hardware for Power and System Architectures Version 6.2. Кроме этого, введение должно содержать обзор ОО — его назначение, условия применения и основные функции безопасности, реализованные в 00.

Утверждения соответствия. Здесь указывается, каким документам «Общих критериев» соответствует данное «Задание по безопасности». Как правило, здесь должен быть указан «Профиль защиты», на котором основано данное Задание по безопасности, или несколько профилей, если такой случай имеет место. Если Задание по безопасности не основано на «Профиле безопасности», то ссылка должна указывать на те части «Общих критериев», которым соответствует данный документ.

Определение проблем безопасности. В этом разделе должны быть описаны:

- основные информационные активы 00, которые должны быть защищены;
- типы злоумышленников (называемых агентами угроз);
- типы угроз;
- предположения о среде, в которой будет работать 00. Таким предположением, например, может являться наличие администратора, который реализует политики безопасности в соответствии с руководством администратора и обнаруживает любые нарушения в конфигурации защитных механизмов 00; или наличие физической защиты 00; или соответствие уровня защищенности доверенных внешних систем, с которыми 00 взаимодействует, уровню защищенности данного ОО и т. п.

Цели безопасности. В этом разделе в достаточно общем виде должны быть сформулированы цели механизмов защиты ОО. Например, здесь может быть сказано, что ОО должен поддерживать функции аудита и дискреционного доступа, что только успешно аутентифицированные пользователи могут выполнять какие-либо действия с 00, что 00 должен предоставлять защищенные каналы для внешнего взаимодействия. В этом разделе также должны быть приведены обоснования целей, поясняющие, от каких угроз объект будет защищен в результате достижения объектом той или иной заявленной цели безопасности.

Расширенные определения компонент. Автор «Задания по безопасности» может в этом разделе привести описания собственных компонент функциональных требований к безопасности и требований доверия к безопасности, не перечисленных в частях 2 и 3 «Общих критериев» или же определенных в «Профиле защиты», на основе которых разработано данное Задание по безопасности.

Требования к безопасности. Это один из основных разделов Задания по безопасности, так как в нем перечислены те требования из каталога части 2 (т. е. функциональные требования к безопасности) и части 3 (требования доверия к безопасности) «Общих критериев», которым удовлетворяет 00. В случае, когда компонент имеет настраиваемые параметры, автор должен указать их явные значения. Для требований доверия к безопасности обычно указывается один из стандартных оценочных уровней доверия EAL 1-7 с указанием дополнительных требований, не вошедших в этот уровень, если они имеются.

Резюме безопасности. В этом заключительном разделе автор должен суммировать основные свойства и механизмы безопасности 00, а также пояснить, каким образом они отражают потенциальные атаки на 00.

Структура «Профиля защиты» аналогична структуре «Задания по безопасности», основное отличие «Профиля защиты» состоит в том, что он носит более общий характер, так как он должен относится ко всем системам определенного типа. Поэтому, например, описания требований безопасности «Профиля защиты» занимают промежуточное положение между аналогичными описаниями частей 2 и 3 «Общих критериев», с одной стороны, и описаниями требований «Задания по безопасности» — с другой. Они более конкретны, чем описания «Общих требований», так как относятся не к информационной системе вообще, а к информационной системе определенного типа, например к файерволу или операционной системе, но менее конкретны, чем описания требований «Задания по безопасности», так как последние относятся к конкретной системе, работающей в определенной среде.

В настоящее время различными участниками консорциума «Общих критериев» разработано довольно большое количество «Профилей защиты», в том числе 17 профилей для продуктов цифровой подписи, 9 профилей для файерволов, 2 профиля для систем контроля доступа и 1 профиль для универсальных операционных систем.

Для ОО, прошедших сертификацию в соответствии с документом «Общая методология оценки безопасности информационных тех

нологий», на сайте http://www.commoncriteriaportal.org публикуются «Задание по безопасности» и «Отчет о сертификации», в последнем приводятся результаты исследования свидетельств безопасности 00, а также условия и результаты тестирования механизмов безопасности.

Вопросы к главе 18

1. Стандарты сертификации безопасности ОС важны по следующим причинам:

а) они создают основу для системы сертификации ОС, результаты которой дают пользователям объективные основании доверять свойствам безопасности конкретной ОС;

б) они дают администратору сети подробные инструкции для достижения безопасной конфигурации ОС;

в) они описывают свойства, которыми должна обладать ОС для того, чтобы быть безопасной.

2. Что из перечисленного правильно характеризует понятие «Доверенная вычислительная база»?

а) вычислительная система, которой можно доверять в определенной степени;

б) компьютер, которому доверяет сертифицируемая система;

в) совокупность аппаратных и программных механизмов сертифицируемой системы, отвечающих за реализацию политики безопасности.

3. Начиная с какого уровня и класса стандарта «Оранжевая книга» к ОС начинает предъявляться требование предоставления пользователям возможности устанавливать правила совместного использования именованных объектов именованными субъектами и/или группами субъектов?

а) A1;

б) C2;

в) B1.

4. Верно ли следующее утверждение о спецификациях «Оранжевой книги»: «При перемещении от вышестоящего класса А к нижестоящему классу С происходит аккумулирование требований безопасности»

а) да;

б) нет.

5. С какого уровня и класса стандарта «Оранжевая книга» начинает предъявляться требование мандатного способа управления доступом?

а) C3;

б) B2;

в) B1.

6. Что из перечисленного правильно характеризует свойства системы класса A1 «Оранжевой книги»:

а) поддержка мандатного доступа;

б) поддержка дискреционного доступа с возможностью определять для каждого именованного объекта список доступа, т. е. перечень именованных субъектов или групп субъектов с указанием действий, которые им разрешены по отношению к данному объекту;

в) для выявления и анализа скрытых каналов должны быть использованы формальные методы.

7. Что из перечисленного является общими характеристиками стандартов «Общие критерии» и «Оранжевая книга»?

а) это международные стандарты;

б) это стандарты, описывающие критерии безопасности любых информационных систем;

в) эти стандарты включают две группы требований: требования у функциональной полноте свойств безопасности системы и требования гарантий качества реализации функций безопасности (называемых также требованиями доверия).

8. Верно ли следующее утверждение: «Стандарт «Общие критерии» не описывает самой методики оценки 00 на соответствие функциональным требованиям и требованиям доверия»?

а) да;

б) нет.

9. Верно ли следующее утверждение: «Требования доверия «Общих критериев» включают оценку «Задания по безопасности» и «Профиля защиты»»?

а) да;

б) нет.

19 Уязвимости программного кода и вредоносные программы

Использование уязвимостей программных кодов

Программная система, состоящая из десятков тысяч строк кода, всегда имеет уязвимости, которые может использовать злоумышленник. Эти уязвимости могут быть результатом ошибок программистов — в соответствии с исследованием CyLab Университета Карнеги Мэллона в среднем каждые 1000 строк кода содержат 20-30 ошибок, из которых 5 % влияют на безопасность системы и 1 % открывает возможность для взлома системы. Уязвимости могут быть и результатом плохого проектного решения или плохого управления проектом. Недаром стандарты безопасности информационных систем, такие как «Оранжевая книга» и «Общие критерии», уделяют столько внимания оценке проектных решений и процесса проектирования. Для того чтобы качество программного кода было высоким, необходимо следовать современным методикам разработки, отладки и тестирования программ.

Существуют известные типичные уязвимости программного кода, которые полезно знать разработчикам для того, чтобы не допускать их появления в своих программах.

Уязвимости, связанные с нарушением защиты оперативной памяти

Области оперативной памяти отдельных процессов защищены друг от друга, а также от области памяти ядра за счет архитектурных решений операционной системы, которые мы рассмотрели в главе 15 «Архитектурная безопасность ОС», поддерживаемых механизмами привилегированного режима процессора. Однако, несмотря на указанные меры, некорректное использование областей памяти все же может происходить в пределах адресного пространства ***одного процесса*** или в пределах ***ядра*** операционной системы. В последнем случае это особенно опасно, так как при этом крах может потерпеть вся система, а не отдельное приложение, как это бывает в первом случае.

Переполнение буфера памяти является, наверно, наиболее часто используемой при атаках уязвимостью, связанной с нарушением защиты оперативной памяти. Мы уже знаем одну такую атаку, приводящую к краху всей системы, которая использует буфер, расположенный в памяти ядра, — это атака Ping смерти. Точнее будет сказать, приводила, так как ошибка в

операционных системах, приводящая к краху при превышении размера IP-пакета размера 65535 байтов, уже давно устранена. Тем не менее, механизм, который эксплуатируется атакой Ping смерти, очень типичен — он использует отсутствие контроля над вводимой из внешнего мира информацией, в данном случае не контролируется длина помещаемого в буфер пакета.

Переполнение стека является частным случаем переполнения буфера памяти. Этот вид уязвимости часто используется злоумышленниками, чтобы заставить ОС выполнить код злоумышленника. Напомним, что стеком является область памяти с реализацией стратегии записи LIFO (Last In First Out) — «последним пришел, первым вышел». Этот способ записи удобен при многократном вызове функций (подпрограмм), так как он обеспечивает экономичный возврат из вызванной функции в вызывающую.

Типичная структура стека, который растет в сторону меньших адресов (архитектура Intel x86), показана на рис. 19.1. Здесь мы видим стек, содержащий данные одной функции ***fi(A\,A2).*** В стек помещены аргументы этой функции, за которым идет адрес возврата в функцию, ее вызвавшую — в данном случае это функция main, т. е. основное тело программы, написанной на языке С. За адресом возврата идет локальная память функции Д, которая используется для хранения ее локальных переменных и массивов. Указатель стека содержит адрес первого слова свободной области стека.

Если функция Д вызывает другую функцию Д, то ее аргументы, адрес возврата в функцию Д и ее локальная память будут размещаться над областью памяти, выделенной в стеке функции Д (рис. 19.2).

При завершении функция Д должна выполнить специальную инструкцию RET, которая вернет управление по адресу возврата в функцию Д и очистит стек от данных функции Д, вернув указатель стека на прежнее место.

Переполнение стека может произойти в случае, когда в область локальной памяти функции помещаются данные, длина которых оказывается больше длины этой области. В таком случае эти данные могут наложиться на адрес возврата и после завершения вызванной функции произойдет переход на некоторый адрес, который может быть случайным или специально сформированным злоумышленником. Многие атаки основаны на том, что в область локальных данных стека помещается вредоносный код, которому затем передается

Рис. 19.1.

Свободная область стека _____
Свободная область стека
Локальные данные функции /!
Локальные данные функции /,
Локальные данные функции /]
Адрес возврата в функцию main
$Л_2$ аргументы функции //Л,. $А_2$) Л,
аргументы функции $f\(А_b Л_2)$

Структура стека до вызова функции /2 из функции

Указатель стека

Свободная область стека
Свободная область стека
Локальные данные функции /2
Локальные данные функции /2
Локальные данные функции /2
Адрес возврата в функцию /,
$А_2$ аргументы функции/2(Л,, Л2)
$А,$ аргументы функции/2Ц, $Л_2$)
Локальные данные функции/!
Локальные данные функции f
Локальные данные функции f
Адрес возврата в функцию main
$Л_2$ аргументы функции/,$(А,, А_2)$
Л, аргументы функции/!(Л|, Л,)

Указатель стека

Рис. 19.2. Структура стека после вызова функции /г из функции f_x

управление за счет подмены содержимого поля «адрес возврата» адресом начала вредоносного кода.

Технику использования переполнения стека для организации атаки мы хотим проиллюстрировать ставшим уже классическим примером программы на языке С из книги Ховарда и Лебланка «Защищенный код»*. Этот пример (в котором английские комментарии заменены на русские) состоит из тела программы StackOverun.c (функция main) и двух функций foo и bar. Функция foo является атакуемой программой, а функция bar — вредоносной программой, которую злоумышленник пытается выполнить.

```
/*
StackOverrun.c
Эта программа является примером того, как переполнение стека может быть использовано для выполнения произвольного кода.
Для достижения этой цели нужно найти вид строки исходных данных, ввод которой приведет к выполнению функции bar */
ifinclude <string.h>
ifinclude <stdio.h>
void foo(const char* input)
{
char buf[10];
printf ("Мой стек выглядит так:\п */,р\п */,р\п У,р\п У,р\п '/,р\п */, р\п\п"); strcpy(buf, input);
printf(" */,s\n", buf);
```



```
printf ("Теперь мой стек выглядит так:\п '/,р\п */.р\п */,р\п У.р\п */.р\п %р\п\п");
}
void bar(void)
{
    printf("Ааа! Меня взломали!\п");
}
int main(int argc, char* argv[])
{
    printf ("Address of foo = "/,p\n", foo); printf
    ("Address of bar = */,p\n", bar); if (argc != 2)
    {
        printf("Пожалуйста введите строку как аргумент!\n"); return -1;
    }
foo(argv[l]); return 0;
}
```

Функция foo (const char* input) выполняет полезную работу — она печатает строку, передаваемую ей в качестве аргумента. Для этого она сначала копирует аргумент в локальный массив символов buf[10], который вмещает не более 10 символов. При этом функция foo не проверяет длину строки-аргумента, чем воображаемый злоумышленник и воспользуется в нашем примере. Функция bar нигде явно не вызывается, задача злоумышленника состоит в вызове ее за счет переполнения стека. Если это удастся, то на экране должен появиться текст «Ааа! Меня взломали!»

Для целей демонстрации в main и foo добавлены строки, печатающие на экране вспомогательную информацию: адреса функций foo и bar, а также состояние стека до копирования строки аргумента в буфер и после. Информация выводится с использованием библиотечной функции printf, которая при задания формата %р печатает адрес имени переменной или функции, а не ее значение, например

printf ("Address of foo = 7,p\n", `foo`); печатает

адрес функции foo.

Та же функция printf, вызванная без второго параметра, распечатывает содержимое стека при работе функции foo — первый раз распечатывается б первых слов (в 32-разрядной машине это соответствует 4 однобайтным символам кода ASCII) стека до копирования строки-аргумента в локальный массив buf[10], а второй раз — после.

Ниже показаны результаты работы программы StackOverrun при задании ей в качестве аргумента строки «Hello». Эта строка состоит из 5 символов, так что она умещается в массив buf[10] без проблем. Мы видим результат такого копирования в первых двух словах стека,

распечатанного функцией foo — первое слово содержит шестнадцатеричные коды 48 (H), 65 (e), 6C (l), 6C (l), а второе содержит код 6F (o).

```
C:\Secureco2\Chapter05>StackOverrun.exe Hello Address of foo = 00401000 Address of
bar = 00401045 Мой стек выглядит так:
00000000
00000000
7FFDF000
0012FF80
0040108A <- адрес возврата из foo, который мы пытаемся переписать
00410EDE
Hello
А сейчас мой стек выглядит так:
6C6C6548 <- Сюда скопирована строка "Hello"
0000006F
7FFDF000
0012FF80
0040108A
00410EDE
```

Адрес возврата из foo в main равен 0040108A, и между ним и кодами «Hello» имеется достаточно безопасный зазор. Адрес функции bar равен 00401045, и наша цель будет достигнута, если мы сможем заменить адрес возврата на это значение.

Сначала проверим, можно ли нарушить нормальную работу программы, если длина строки-аргумента будет больше 10 символов. Вызовем ее с аргументом «AAAAAAAAAAAAAAAAAAAAAAAA», который состоит из 24 символов с шестнадцатеричным кодом 41(A). Результат такого вызова показан ниже. Видно, что весь стек оказался заполнен кодом 41, в том числе таким оказался и адрес возврата из функции foo. После распечатки результатов программа аварийно завершается операционной системой с ошибкой недопустимого обращения к адресу 414141.

```
C:\Secureco2\Chapter05>Stack0verrun.exe AAAAAAAAAAAAAAAAAAAAAAAA
Address of foo = 00401000 Address of bar = 00401045 Мой стек выглядит так:
00000000
00000000
7FFDF000
0012FF80
0040108A <- адрес возврата из foo, который мы пытаемся переписать 00410EDE
AAAAAAAAAAAAAAAAAAAAAAAA А сейчас мой стек выглядит так:
41414141 <- Сюда скопирована строка "AAAAAAAAAAAAAAAAAAAAAAAA"
41414141
41414141
41414141
41414141 <- адрес возврата из foo
41414141
```

Теперь, когда видно, что программа действительно уязвима из- за переполнения стека, осталось только подобрать нужную строку- аргумент, чтобы в нужном месте стека оказался адрес функции bar (00401045). Такой строкой будет, например строка "ABCDEFGHIJKL

MNOP\Ox40\OxlO\Ox45", в которой последние три символа заданы в шестнадцатеричном виде. Мы опустим детали задания строки, содержащей спецсимвол 0x10 (авторы примера делают это с помощью простого perl-скрипта) и ниже покажем результат работы StackOver- run с такой строкой-аргументов.

```
C:\Secureco2\Chapter05>Stack0verrun.exe "ABCDEFGHIJKLMN0P\0x40\0xl0\0x45"
Address of foo = 00401000 Address
of bar = 00401045 Мой стек
выглядит так:
00000000
00000000
7FFDF000
0012FF80
0040108A <- адрес возврата из foo, который мы пытаемся переписать
00410EDE
ABCDEFGHIJKLMN0PS E A сейчас мой
стек выглядит так:
44434241 <- Сюда скопирована строка "ABCDEFGHIJKLMN0P\0x40\0x10\0x45"
48474645
4C4B4A49
504F4E4D
00401045< - адрес возврата из foo
00410EDE
Ааа! Меня взломали!
```

Так как функция bar распечатала свою строку «Ааа! Меня взломали!», значит, мы добились нужного результата — смогли заменить адрес возврата из функции foo в главную программу main на адрес функции bar. Об этом говорит и адрес возврата, распечатанный функцией foo.

Защитить свою программу от переполнения стека можно разными способами. Одним из них является использование языков программирования, которые автоматически контролируют защиту памяти, например C# и Java делают это, в то время как C и C+ + оставляют это на откуп программисту. Можно не полагаться на компилятор, а просто проверять корректность ввода (и не только его длину) каждый раз после получения аргумента от пользователя или другой функции.

Существуют также расширения компиляторов C, такие как Stack Guard, которые предпринимают дополнительные меры для обнаружения искажения адреса возврата входе выполнения функции. Для этого в стек между адресом возврата и областью локальных переменных помещается дополнительное слово, которое называют «канарейкой» (canary word), поскольку оно выполняет ту же функцию что и канарейка, которую когда-то шахтеры запускали в шахту, чтобы проверить, не скопился ли там газ. Аналогичным образом оставшееся неповрежденным к моменту возврата из функции слово-канарейка, показывает, что переполнения буфера не было и можно безопасно использовать адрес возврата, следующий за этим словом. Слово-канарейку выбирают каждый раз случайным образом.

Некоторые операционные системы помечают область стека как неисполняемую, что предотвращает выполнение вредоносного кода в случае, когда он попадает в стек.

Более сложно для злоумышленника использовать переполнение «кучи» (heap) — памяти, динамически выделяемой программе по запросу malloc. Переполнение буфера, находящегося в куче, не вызывает краха программы, но разрушает структуры данных, что приводит к неверным результатам, производимым программой.

Уязвимости контроля вводимых данных

Переполнение буфера является частным случаем уязвимостей, являющихся следствием слабого контроля вводимых данных. В более общем случае специальный вид вводимых данных может вызвать совершено непредвиденные разработчиком последствия и этот факт может использовать злоумышленник. Обобщенно такой тип атаки называют ***внедрением кода (code injection).*** «Любой ввод данных — это зло!» — любят повторять специалисты по разработке безопасного кода.

Внедряемый код может представлять собой программный код в классическом смысле этого термина, например это может быть код Java, который выполняется виртуальной Java-машиной атакуемого компьютера. А может представлять и некоторую последовательность символов, неверно обрабатываемую программой. Это происходит в тех случаях, когда разработчик программы рассчитывает, что пользователи всегда будут вводить только «правильные» данные, т. е. данные только такого типа и из такого диапазона, который разработчик имел в виду.

Тривиальным примером является веб-форма, в которой пользователю предлагается ввести номер статьи, выбранный из списка, включающего 10 статей. Если разработчик не предвидел, что вместо ожидаемого положительного числа из диапазона от 1 до 10 пользователь может ввести «-1», то его приложение может повести себя совсем не так, как он

планировал, например выдать конфиденциальный документ вместо публично доступной статьи. Многие системы программирования пытаются исключить такие ситуации за счет того, что заставляют разработчика явно описывать тип вводимых переменных и их возможные значения, но это не всегда оказывается достаточным. Например, тип переменной «текстовая строка» не сможет предохранить приложение от подмены имени пользователя строкой скрипта или строкой htmi-тега, так как все эти переменные являются строками и нужен более детальный анализ содержимого строки, чтобы распознать попытку «подделки» вводимых данных.

Специальные символы, встречающиеся в строке ввода, могут вызывать непредвиденные эффекты в поведении программы, если разработчик не учел такой возможности ввода. Например, пусть некоторый веб-сервис принимает запросы на загрузку файлов удаленных пользователей из их домашних каталогов. Разработчик этого сервиса предусмотрел проверку имени запрашиваемого файла и передает это имя системному вызову операционной системы для чтения файла только в случае положительной проверки. Однако, если разработчик использует свой собственный парсер имен38, который не учитывает наличие точек в составном имени, то он может посчитать имя /home/bob/files/../../../etc/passwd легальным, принимая во внимание только его начальную часть /home/bob/files и передать его операционной системе. Парсер операционной системы работает правильно, и в результате файл паролей системы будет передан пользователю.

Существует большое количество различных типов атак внедрения кода в зависимости от атакуемого приложения и применяемого трюка. Очень распространены атаки на веб-сервисы и базы данных, что естественно, так как это наиболее популярные на сегодняшний день приложения.

Для несанкционированного доступа ***к базам данных*** злоумышленники пытаются использовать лазейки для внедрения кода с помощью специального вида SQL-запросов к базам данных, например очень часто используются кавычки в запросе, что существенно изменяет отработку запроса SQL сервером и может привести к выдаче не отдельной записи таблицы базы данных — что было бы санкционированным ответом, а всех записей таблицы, к которым злоумышленник доступа не имеет.

---

38 Парсер — программа синтаксического анализа строки, от parse — анализировать, разбирать.

Поясним это на примере. Запрос к записям SQL-базы данных может выглядеть так:

```
string sql = "select * from client where name = '" + name + "'"
```

По этому запросу выбираются записи из таблицы client, у которых имя пользователя совпадает с именем, получаемым в результате ввода данных в программу, строку из которой мы привели.

Разработчик этой программы предполагает, что пользователь будет вводить имя как последовательность алфавитно-цифровых символов, например Alica. Тогда запрос программы к базе данных будет выглядеть так:

```
select * from client where name = 'Alica'
```

Однако, если пользователь введет такое имя:

```
Alica' or 1=1 - -
```

то запрос к базе данных будут уже выглядеть иначе:

```
select * from client where name = 'Alica' or 1=1 - -
```

Такой запрос вернет пользователю все записи из таблицы client, относящиеся к имени Alica (первая часть имени) плюс все записи из этой же таблицы, так как условие 1= 1 всегда истинно. Этот трюк выглядит очень простым, но без надлежащего контроля входной информации он может оказаться эффективным средством несанкционированного доступа к данным.

Разработчик программы может попытаться проверить введенное имя с помощью операторов SQL, например так:

```
string sql = "select * from client where name = '" + name + and name
like 'a ".
```

разрешая запрос записей имен пользователей, начинающихся с буквы «а».

Однако такую проверку отсекает использование в имени пользователя двух дефисов (т. е. "- -"), которые являются признаком комментария, поэтому база данных воспримет оператор проверки like как комментарий и снова все записи таблицы будут выданы по такому запросу.

Так как язык запросов к базам SQL очень гибок, то и способов обмануть базу данных так же много. Мы не будем в них углубляться, надеясь что приведенный простой пример дает хорошее представление о технике таких вредоносных запросов.

Приемы для внедрения кода в ***веб-серверы*** злоумышленники используют тот факт, что многие серверы строят динамические страницы с помощью скриптов — PHP, Java, ASP и других. Очевидно,

Рис. 19.3. Периметр доверия программы

что наиболее естественный способ обмануть такой веб-сервер — это «подсунуть» ему свой скрипт под видом обычного текста, а когда такой текст попадет в HTML-тело веб страницы, то сервер начнет выполнять его инструкции, выполняя чужую волю. Мы рассмотрим такие примере в разделе, посвященном безопасности веб-сервисов.

Метод борьбы с внедрением вредоносного кода при вводе данных имеется только один — любые данные, которые программа получает от источника, не вызывающего доверия, требуют тщательной проверки перед их использованием. Этот подход аналогичен принципу защиты периметра сети, рассмотренного в предыдущей части: вся информация, которая приходит извне доверенного периметра, должна тщательно фильтроваться.

Этот принцип иллюстрирует рис. 19.3. Здесь показана программа P1, работающая на внутреннем веб-сервере предприятия. Периметр доверия этой программы охватывает программы, работающие на том же сервере, а также программы, работающие на внутреннем SQL сервере. Поэтому данные, поступающие в программу P1 от программы P2, не подлежат тщательной проверке. Гораздо более строгий подход должен применяться к вводу данных от пользователя этого предприятия и от программы P3, работающих за пределами периметра доверия.

Фильтрацию вводимых данных может делать как сама программа, так и файервол, работающий на прикладном уровне, а также система обнаружения вторжений (ISD) хоста. Хорошо, когда фильтрацию выполняют все три компонента. Сама программа лучше всего знает специфику вводимых данных и возможные угрозы, в то время как файервол и ISD могут выполнять более общие проверки для определенного типа угроз.

Скрытые коммуникации и скрытые каналы

Скрытый канал (covert channel) — это коммуникационный канал, пересылающий информацию методом, который изначально для этого не был предназначен.

Персонажи детективных и шпионских романов пользуются скрытыми каналами постоянно — журнал «Огонек» в левой руке, сообщающий, что операция отменяется, цветок на подоконнике, предупреждающий профессора Плейшнера об опасности, количество сальто, исполненное цирковым артистом для передачи информации о количестве танков на полигоне, — все это примеры скрытых каналов.

Техника скрытых каналов основывается на том простом факте, что любое изменение состояния какого-то объекта несет информацию. И обнаружить закономерность в изменениях состояния объекта, которое изначально никак не было предназначено для передачи информации, бывает очень сложно. За высокую степень скрытности многие каналы такого типа расплачиваются низкой скоростью передачи информации. Например, наличие или отсутствие цветка на подоконнике несет в себе 1 бит информации. Количество сальто в цирковом номере более информативно, но и с помощью такого способа много данных не передашь.

Теперь рассмотрим несколько примеров из практики информационных технологий. Пусть, например, имеются два процесса: процесс А, который имеет высокий уровень допуска, и процесс В, имеющий низкий уровень допуска. Злоумышленник имеет возможность получать информацию *только от* процесса В. В соответствии с мандатным способом управления процесс В не имеет легальной возможности прочитать некие данные с высоким уровнем секретности. Но к ним имеет доступ процесс А, который однако в соответствии с тем же правилом мандатного способом доступа не имеет права передавать данные на нижележащий уровень процессу В. Возникает вопрос, существует ли для процесса А какой-нибудь обходной, не вызывающий подозрений способ передать секретные данные процессу В?

Ответ — да. Таким способом может быть, например, варьирование степенью заполнения буфера при выводе легальных данных. Пусть процесс А, выполняя легальную операцию вывода, изменяет объем выводимых данных в соответствии с каким-либо условным кодом, известным процессу-адресату В. И пусть процесс В имеет возможность анализировать состояние буфера. Таким образом процесс А, выполняя казалось бы вполне законные действия, может передать процессу В в закодированном виде секретные данные. Подчеркнем, что эти данные, не имеют ничего общего с легально выводимой информацией. Здесь можно провести аналогию с передачей информации модулированным синусоидальным сигналом, когда информация кодируется с помощью варьирования амплитуды. Именно такая идея положена в основу скрытого канала.

Скрытый канал — это механизм для передачи информации, не предусмотренный разработчиком информационной системы. Естественно, что передача данных по скрытому каналу не контролируется обычными

39 http://www.infosecurity.ru/-gazeta/content/060922/article01.shtml

http://www.cs.umd.edu/~jkatz/TEACHING/comp_sec_F04/downloads/confinement.pdf

механизмами безопасности ОС, такими как аутентификация и авторизация, именно поэтому наличие скрытых каналов очень опасно. Недаром все стандарты безопасности информационных систем, такие как «Оранжевая книга» и «Общие критерии», уделяют скрытым каналам большое внимание и требует анализа системы на наличие таких каналов при сертификации.

Кроме скрытых каналов, существуют также ***скрытые коммуникации.*** Скрытые коммуникации используют для передачи сообщений *легальные* каналы, однако действуют таким образом, что эти сообщения незаметны для легальных пользователей, так как они скрыты внутри других сообщений, используемых как контейнеры. Наиболее популярным примером скрытых коммуникаций является помещение секретного сообщения в биты цифровой фотографии, внешний вид которой от такой операции практически не отличается от исходного. Скрытые коммуникации также называют subliminal channel, что дословно переводится как *подсознательный канал,* канал, находящийся ниже уровня восприятия; есть также предложение называть такие каналы *потайными39.*

Общей особенностью скрытых каналов и скрытых коммуникаций является то, что здесь скрывается не только содержание сообщения, но и сам *факт* коммуникации. Насколько сам факт коммуникации может оказаться ценным для разведки говорит программа PRIZM, о наличии которой мир узнал из утечек Эдварда Сноудена. Эта программа собирает *метаданные* об электронных коммуникациях, т. е. данные о том, кто с кем, где и когда переписывался. И хотя собственно содержание сообщений не включается в собираемую информацию, метаданные сами по себе могут помочь раскрыть заговор или выявить агентов. Да и в личной жизни просто факт переписки одного из супругов с третьим лицом может привести к неожиданным последствиям.

Скрытыми коммуникациями занимается *стеганография,* поэтому такой способ передачи секретной информации иногда называется *сте- гоканалом.* Стеганография, как и криптография, имеет древнюю историю, и человечество придумало достаточно много остроумных способов помещения секретных данных в невинные с виду сообщения. В информационных системах не всегда можно провести четкую грань между техникой скрытых каналов и скрытых коммуникаций, мы увидим это немного дальше.

Примеры скрытых каналов

Существуют два типа скрытых каналов: ***скрытый канал памяти*** (*covert storage channel*) и ***скрытый временной канал (****covert timing*

channel). В первом случае для скрытной передачи информации используется модулирование характеристик памяти, таких как адреса, объем свободной памяти и др. Процесс использует скрытый временной канал, если для кодирования информации он варьирует временные характеристики, связанные с его собственным выполнением, например, в мультипрограммном режиме, это может быть использованная доля кванта процессорного времени.

Большое количество работ по анализу скрытых каналов в операционных системах посвящено системам, реализующим мандатную модель доступа. Возможно, это связано с тем, что мандатная модель ставит казалось бы непреодолимые барьеры на пути распространения конфиденциальной информации сверху вниз, которые тем не менее преодолеваются секретными каналами. Рассмотрим несколько примеров скрытых каналов в ОС с мандатным доступом, где процесс/пользователь более высокого уровня допуска (High) пытается нелегально передать конфиденциальную информацию процессу/пользователю более низкого уровня допуска (Low), обходя систему контроля доступа.

Чтение имен файлов или каталогов. Если в системе пользователю Low he разрешено читать содержимое файлов с более высоким уровнем допуска, но разрешено читать содержимое каталога, содержащего такие файлы, то скрытый канал может быть организован очень простым способом. Для этого пользователь High может, например:

- давать файлу имя, несущее информацию;
- просто помещать в каталог файл с определённым именем или удалять его, информируя о некотором событии с двумя состояниями (аналог цветка на подоконнике);
- помещать в каталог определенное количество файлов — именно это количество является сообщением;
- придумать еще много способов использовать такую вольность с чтением содержимого каталога для передачи информации на нижний уровень.

Использование факта блокировки файла. Этот вариант скрытого канала был описан в статье Батлера Лэмпсона «Заметка о проблеме ограничения» (A Note on confinement problem), опубликованной в 1973 году в журнале Communications of the ACM*, в которой впервые было введено понятие скрытого канала, а также были рассмотрены несколько типов таких каналов и пути их предотвращения.

В примере Лэмпсона используется некоторый абстрактный* системный вызов открытия файла open(file) со следующими свойствами. Когда какой-либо процесс открывает файл вызовом open(file), то для всех других процессов этот файл становится недоступным — заблокированным. Все попытки других процессов открыть данный файл являются неудачными. Файл становится доступным только после того, когда использующий его процесс выполнит системный вызов закрытия файла — close(file).

Этим системным вызовом могут пользоваться как High-, так и Low-процессы. Лэмпсон показал, что с помощью трех несложных процедур, использующих вызовы open(file) и close(file), а также трех файлов с низким уровнем секретности, которые процедура более высокого уровня допуска High может открывать только для чтения, а процедура с более низким допуском Low для чтения и для записи, можно организовать передачу данных от High к Low, в обход запретов мандатной системы доступа ОС.

На рис. 19.4 приведена блок-схема процедуры settrue(file), которая циклически пытается открыть файл file. В случае успеха — файл разблокирован и открыт — процедура завершает работу.

Процедура setfalse(file) (рис. 19.5) выполняет закрытие файла file. Предполагается, что она всегда успешна.

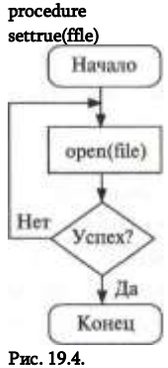


Рис. 19.4. Процедура

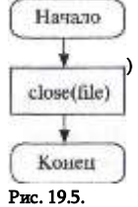


Рис. 19.5. Процедура setfalse(file)

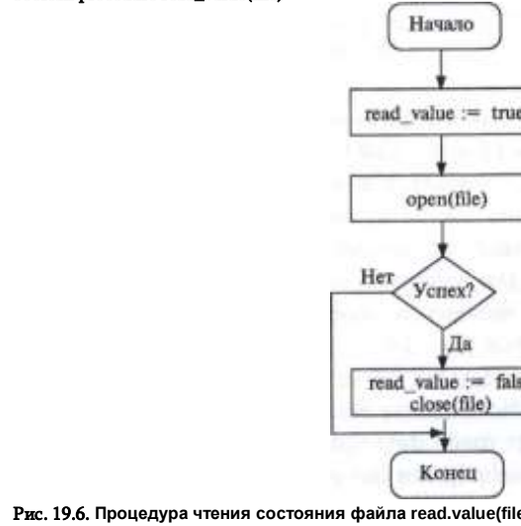


Рис. 19.6. Процедура чтения состояния файла read.value(file

То есть не относящийся к определенной ОС и языку программирования.

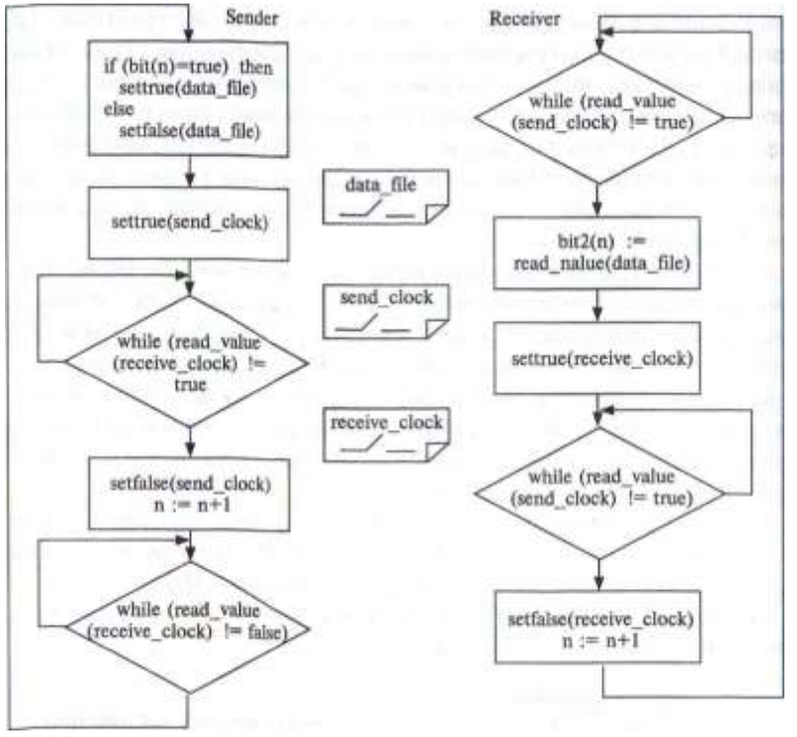


Рис. 19.7. Передача данных между процессом Sender (высокий уровень допуска) и Receiver (низкий уровень допуска)

Третья процедура (рис, 19.6) устанавливает значение бита в значение true (1), если файл заблокирован (попытка открыть файл оказалась неуспешной), и значение false (0) — в противном случае.

Нетрудно показать, как можно передавать данные с верхнего уровня на нижний, используя три описанные процедуры и три файла: read-value для передачи данных, send_clock и receive.clock для синхронизации процессов передачи и чтения данных. Этот процесс иллюстрируется рис. 19.7.

Процесс Sender читает данные из массива bit(n), которые он хочет передать процессу Receiver, используя технику скрытого канала, и открывает файл data.value, если бит равен 1 (true), или закрывает файл data.value, если бит равен 0 (false). Затем Sender сообщает процессу Receiver о том, что очередной бит был передан, устанавливая *сигнализирующий файл* send_clock в открытое состояние (true).

Процесс Receiver ждет, пока файл send-dock не перейдет в открытое состояние, после чего считывает состояние файла read-value и присваивает соответствующее значение своему массиву bit2(n). После этого он устанавливает *свой сигнализирующий файл* receiver-dock в открытое

состояние (true), оповещая процесс Sender о том, что очередное значение массива bit(n) прочитано.

Процесс Sender, дождавшись сигнала о том, что очередной бит считан, устанавливает файл send_clock в начальное состояние false, ждет, пока процесс Receiver установит свой сигнализирующий файл receive_clock в начальное состояние false и затем передает аналогичным образом следующий бит массива bit(n).

Скрытый временной канал может быть построен на основе синхронизации обращений к некоторому системному ресурсу, например к сокету TCP или UDP. Процесс-получатель считывает периоды занятости определенного сокета и получает таким образом информацию. В другом случае процесс может кодировать информацию, посылая пакеты на удаленный хост в определенные моменты времени.

Сетевые средства и протоколы предоставляют большой простор для создания скрытых каналов. Например, в пакете TCP SYN имеется *параметр* ISN — начальный номер последовательности (Initial Sequence Number), — которому присваивается произвольное значение при запросе установления TCP-соединения. Вполне логично использовать этот параметр для кодирования скрытой информации, передавая в то же время по установленному соединению легальные сообщения. То же самое можно сказать и о *номерах портов* клиентской части протоколов TCP или UDP, которые также выбираются произвольным образом. Источником информации может также служить *размер пакета,* если дополнять пакеты с легальной информации заполнителем до размера, соответствующего кодовому значению. В последнем случае канал может быть отнесен как к скрытому, так и к стегоканалу, поскольку при передаче скрытой информации обычный информационный поток используется как контейнер.

Ну и нельзя не сказать чуть-чуть подробнее о классическом ***стеганографическом канале,*** который использует модификацию битов цифрового изображения. Существует большое количество программ, в том числе и бесплатных (например, EZStego, Xiao Steganography, Steganography Studio), которые умеют делать это для различных форматов цифровых изображений — JPEG, TIFF, BMP, PNG и других. Изменение одного младшего бита цвета RGB-пиксела с 24-битовой палитрой не изменяет восприятие изображения, а информация предается внутри фотографии достаточно большая. Для того чтобы такой канал стал действительно тайным, сама по себе передача фотографий не должна вызывать подозрений, так, отправка фотографий кошек настолько популярна в социальных сетях, что их можно использовать

как очень хороший контейнер — только надо делать это периодически, создавая для себя устойчивую легенду члена клуба ми-ми-ми.

Можно встроить скрытую информацию и в текстовые файлы. Например, популярный редактор MS Word использует служебные метки «Начало текста» и «Конец текста», при этом служебная информация, не попадающая в область текста между метками начала и конца, не отображается на экране или печати. Поэтому байты скрытого сообщения можно встроить в служебную информацию незаметным для пользователя способом. Вообще, развитые текстовые редакторы используют большое количество служебных меток, таких как начало нового стиля, метка индекса и т. п., которыми можно кодировать скрытое сообщение.

Борьба со скрытыми каналами. Самым эффективным способом борьбы со скрытым каналом является его уничтожение. Эту идею проще всего проиллюстрировать на примере стегоканала, использующего цифровые изображения. При конвертировании формата скрытые биты скорее всего будут потеряны и сообщение автоматически уничтожено. Поэтому любое конвертирование цифровых изображений является эффективным способом борьбы с этом видом скрытых каналов. То же самое относится и к архивированию, оно применимо и к текстовым файлам.

Уничтожить скрытый временной канал можно сглаживанием трафика за счет внесения задержек пакетов. Такая техника, известная как шейпинг (shaping), давно используется для уменьшения очередей пакетов в буферах маршрутизаторов и коммутаторов.

Скрытый канал, использующий номера портов TCP/UDP, можно разрушить, целенаправленно изменяя эти номера.

Существует теоретический подход, позволяющий выявлять потенциальные скрытые каналы на основе анализа моделей. В обзоре «Скрытые каналы»40 упоминаются две таких модели: модель зависимостей и модель матрицы ресурсов.

Некоторые специалисты по безопасности высказываются скептически в отношении практической важности борьбы со скрытыми каналами, считая, что большой интерес к скрытым каналам объясняется только необходимостью сертификации систем и естественным любопытством ученых, а на практике злоумышленники не используют такие экзотические средства41.

Внедрение в компьютеры вредоносных программ

Многочисленная группа атак связана с внедрением в компьютеры ***вредоносных программ (malware),*** к числу которых относятся троянские и

40 http://www.jetinfo.ru/Sites/new/Uploads/2002.ll.DF9C812FFBD9496BAE9694E27F2D9DlD.pdf

41 В.А. Галатенко «О каналах скрытых, потайных, побочных. И не только», www.jetinfo.ru/stati/o-kanalakh-skrytykh-potajnykh-pobochnykh-i-ne-tolko

шпионские программы, черви, вирусы, спам, логические бомбы и некоторые другие типы программ, нацеленные на нарушение безопасности. Вредоносный код чаще всего классифицируют по способу проникновения кода в чужой компьютер, а также по целевому назначению, так что один и тот же код может иметь по крайней мере два названия, например червь и шпионская программа.

Самый простой способ проникновения — «самодоставка», когда пользователь загружает файлы из непроверенных источников (съемных носителей или веб-сайтов) либо беспечно открывает подозрительный файл, пришедший к нему по электронной почте. Существуют и более сложные представители вредоносных программ, обладающие собственными механизмами «размножения», копии таких программ распространяются по компьютерам сети без участия пользователей.

Ущерб, наносимый вредоносными программами, может выражаться не только в уничтожении, искажении или похищении информации, приведении в нерабочее состояние программного обеспечения, а значит, и компьютера в целом, но и в значительных затратах времени и сил администраторов на обнаружение и распознавание атак, фильтрацию внешних сообщений, тестирование и перезагрузку систем. Вредоносные программы в начале этого десятилетия были одной из основных причин нарушения безопасности компьютерных сетей. Однако, как показала статистика, в последние два года суммарный ущерб, нанесенный вредоносными программами предприятиям, резко снизился. Это связывают, в том числе, с улучшением качества антивирусных средств и ужесточением наказаний за такого рода преступления.

На практике злоумышленники часто сочетают в одной и той же программе различные типы угроз. Например, некоторые черви способны маскироваться как троянские программы или подобно вирусам заражать исполняемые файлы на локальном диске, а некоторые вирусы наделены способностями червей самокопироваться на другие компьютеры. Кроме того, вы можете встретить и другую классификацию вредоносных программ, где, скажем, троянские программы и черви рассматриваются как разновидности вирусов.

Троянские программы

Троянские программы, или ***трояны*** (***trojan***) — это разновидность вредоносных программ, которые наносят ущерб системе, ***маскируясь*** под какие-либо полезные приложения.

Троянские программы могут применять в качестве прикрытия знакомые пользователю приложения, с которыми он работал и раньше, до появления в компьютере «троянского коня». При другом подходе в полном соответствии с древней легендой троянская программа принимает вид *нового* приложения, которое пытается заинтересовать пользователя-жертву какими-то своими якобы полезными функциями.

Однако суть троянской программы и в том, и в другом случаях остается вредительской: она может уничтожать или искажать информацию на диске, передавать данные (например, пароли) с «зараженного» компьютера на удаленный компьютер хакера, приводить в неработоспособное состояние установленное на атакованном компьютере программное обеспечение, участвовать в проведении DoS-атак на другие удаленные компьютеры. Так, одна из известных троянских программ AIDS TROJAN DISK7, разосланная нескольким тысячам исследовательских организаций на дискете, при запуске перемешивала символы в именах всех файлов и заполняла все свободное пространство жесткого диска. После этого программа от имени злоумышленника предлагала помощь в восстановлении диска, требуя взамен вознаграждение для автора этой программы. (Злоумышленники могут также шантажировать пользователя, зашифровывая его данные.) Кстати, описанное компьютерное преступление завершилось поимкой хакера-шантажиста.

Троянские программы могут быть отнесены к самому простому по реализации виду вредоносных программ.

Сетевые черви

Сетевые черви (worm) — это программы, способные к *самостоятельному распространению* своих копий среди узлов в пределах локальной сети, а также по глобальным связям, перемещаясь от одного компьютера к другому без всякого участия в этом процессе пользователей сети.

Поскольку большинство сетевых червей передаются в виде файлов, основным механизмом их распространения являются сетевые службы, основанные на файловом обмене. Так, червь может рассылать свои копии по сети в виде вложений в сообщения электронной почты или с помощью размещения ссылок на зараженный файл на каком-либо веб-сайте. Однако существуют и другие разновидности червей, которые для своей экспансии используют более сложные приемы, например, связанные с ошибками («дырами») в программном обеспечении.

Главная цель и результат деятельности червя состоит в том, чтобы передать свою копию на максимально возможное число компьютеров. При этом для поиска компьютеров — новых потенциальных жертв — черви задействуют встроенные в них средства. Типичная программа-червь не удаляет и не искажает пользовательские и системные файлы, не перехватывает электронную почту пользователей, не портит

содержимое баз данных, а наносит вред атакованным компьютерам потреблением их ресурсов, например для рассылки спама или проведения массированной атаки в составе ботнета.

При создании типичного сетевого червя хакер прежде всего определяет перечень сетевых уязвимостей, которые он собирается использовать для проведения атак средствами создаваемого червя. Таким уязвимостями могут быть как известные, но не исправленные на некоторых компьютерах ошибки в программном обеспечении, так и пока неизвестные никому ошибки, которые обнаружил сам хакер. Чем шире перечень уязвимостей и чем более они распространены, тем больше узлов может быть поражено данным червем.

Червь состоит из двух основных функциональных компонентов — атакующего блока и блока поиска целей.

Атакующий блок состоит из нескольких модулей (*векторов атаки),* каждый из которых рассчитан на поражение конкретного типа уязвимости. Этот блок открывает «входную дверь» атакуемого хоста и передает через нее свою копию.

Блока поиска целей (локатор) собирает информацию об узлах сети, а затем на основании этой информации определяет, какие из исследованных узлов обладают теми уязвимостями, для которых хакер имеет средства атаки.

Эти два функциональных блока являются *обязательными* и присутствуют в реализации любой программы-червя. Некоторые черви нагружены их создателями и другими вспомогательными функциями, о которых мы скажем позже.

Упрощенно жизненный цикл червя может быть описан рекурсивной процедурой, состоящей из циклического запуска локатора и атакующего блока на каждом из последующих заражаемых компьютеров (рис. 19.8).

В начале каждого нового цикла червь, базирующийся на захваченном в результате предыдущей атаки компьютере, запускает локатор для поиска и формирования списка узлов-целей, пригодных для проведения каждой из специфических атак, а затем, используя средства атакующего блока, пытается эксплуатировать уязвимости узлов из этого списка. В результате успешной атаки червь копирует все свои программы на «новую территорию» и активирует локатор. После этого начинается новый цикл. На рисунке показано, как червь лавинообразно распространяется по сети. Заражение тысяч компьютеров может занять всего несколько минут. Некоторые виды червей не нападают на уже зараженные и/или подвергающиеся атаке в данный

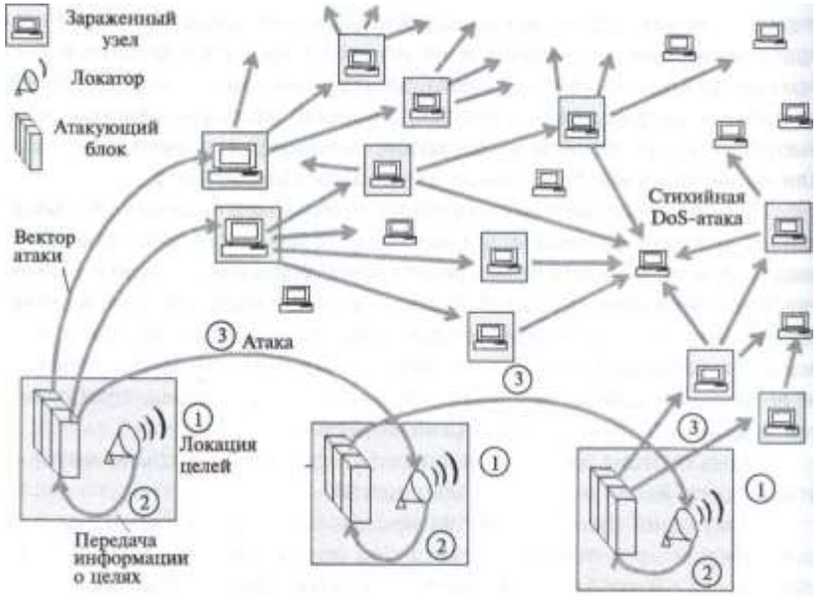


Рис. 19.8. Экспансия червя в сети

момент узлы. Если же такая проверка не предусмотрена в алгоритме работы червя, то в сети случайным образом могут возникать очаги стихийных DoS-атак.

Локатор идентифицирует цели по адресам электронной почты, IP-адресам, характеристикам установленных на хостах операционных систем, номерам портов, типам и версиям приложений.

Для сбора информации локатор может предпринимать действия, связанные как с поисками интересующих данных на захваченном им в данный момент хосте, так и зондированием сетевого окружения. Простейший способ получить данные локально — прочитать файл, содержащий адресную книгу клиента электронной почты*. Помимо почтовых адресов, локатор может найти на узле базирования другие источники информации, такие как таблицы конфигурационных параметров сетевых интерфейсов, ARP-таблицы и таблицы маршрутизации. Зная IP-адреса хоста базирования и шлюзов, локатор достаточно просто может определить IP-адреса других узлов этой сети. Для идентификации узлов локатор может также использовать ICMP- сообщения или запросы ping, указывая в качестве адресов назначения

Для коллекционирования почтовых адресов локатор может прибегать и к более интеллектуальным методам, которые используют в своей работе спамеры (о спаме см. далее).

все возможные IP-адреса. Для определения того, какие приложения работают на том или ином хосте, локатор сканирует различные ***хорошо известные*** номера TCP- и UDP-портов. Определив тип приложения, локатор пытается получить более детальные характеристики этого приложения.

Например, пусть некоторая программа-червь имеет в своем арсенале средства для атаки на некоторые версии веб-сервера Apache. Для поиска потенциальных жертв локатор этого червя зондирует узлы сети, посылая умышленно ошибочные запросы к веб-серверу:

```
GET / HTTP/1.l\r\n\r\n
```

Узел, на котором установлен сервер Apache, отвечает на такой запрос так, как и рассчитывал разработчик червя, т. е. сообщением об ошибке, например это может быть сообщение такого вида:

```
HTTP/1.1 400 Bad Request
Date:  Mon, 23 Feb 2004 23:43:42 GMT
Server:  Apache/1.3.19 (UNIX) (Red-Hat/Linux) mod.ssl/2.8.1
0penSSL/0.9.6 DAV/1.0.2 PHP/4.0.4pll mod.perl/1.24.01
Connection:  close
Transfer-Encoding:  chunked
Content-Type:  text/html; charset=iso-8859-l
```

Из этого ответа локатор узнает о том, что на узле установлен веб-сервер Apache версии 1.3.19. Для червя этой информации может быть достаточно, чтобы внести данный узел в число целей.

Собрав данные об узлах сети, локатор анализирует их подобно тому, как это делает хакер при поиске уязвимых узлов. Для атаки выбираются узлы, удовлетворяющие некоторым условиям, которые говорят о том, что данный узел ***возможно*** обладает уязвимостями нужного типа (для них в атакующем блоке есть средства нападения). Понятно, что при таком «предположительном» способе отбора целей не всякая предпринятая атака обязательно приводит к успеху. Неудача рассматривается атакующим блоком червя как штатная ситуация, он просто сворачивает все свои действия, направленные на не поддавшийся атаке узел, и переходит к атаке следующей цели из списка, подготовленного локатором.

Рассмотрим более подробно, как работает атакующий блок червя. Среди механизмов, позволяющих червю передать свою копию на удаленный узел, наиболее длинную историю имеет уязвимость ***ошибки переполнения буфера.*** Мы рассмотрели эту технику внедрения и активизации вредоносной программы выше, в разделе «Уязвимости, связанные с нарушениями защиты оперативной памяти».

Помимо локатора и атакующего блока червь может включать некоторые дополнительные функциональные компоненты.

Блок удаленного управления и коммуникаций служит для передачи сетевым червям команд от их создателя, а также для взаимодействия червей между собой. Такая возможность позволяет хакеру координировать

работу червей для организации распределенных атак отказа в обслуживании. Сетевые черви могут быть также использованы для организации параллельных вычислений при решении таких требующих большого объема вычислений задач, как, например, подбор секретного ключа шифрования или пароля.

Блок управления жизненным циклом может ограничивать работу червя определенным периодом времени.

Блок фиксации событий используется автором червя для оценки эффективности атаки, для реализации различных стратегий заражения сети или для оповещения других пользователей о повреждениях, нанесенных их компьютерам. Результатом работы данного блока может быть, например, список IP-адресов успешно атакованных машин, посланный хакеру в виде файла или сообщения электронной почты.

Вирусы

Вирус (virus) — это вредоносный программный фрагмент, который может внедряться в другие файлы.

Стремление злоумышленника сделать код вируса как можно более коротким часто ограничивает логику работы вируса очень простыми решениями, которые, однако, иногда приводят к весьма разрушительным последствиям. Так, например, один из реально существовавших вирусовд состоящий всего из 15 (!) байтов, записывал свою копию поверх других файлов в начало каждого сектора диска, в результате система очень быстро терпела крах. Некоторым утешением в таком и подобных ему случаях является то, что одновременно с крахом компьютера прекращает свое существование и вирус.

Вирус может внедрять свои фрагменты в разные типы файлов, в том числе в файлы исполняемых программ (рис. 19.9). При этом возможны самые разные варианты: замещение кода, когда размер инфицированного файла не меняется, вставка вирусного кода целиком в начало или конец исходной программы, замена фрагментов программного кода фрагментами вируса с перестановкой замещенных фрагментов и без перестановки, и т. д., и т. п. Более того, код вируса может быть зашифрован, чтобы затруднить его обнаружение антивирусными программами.

В отличие от червей вирусы (так же, как и троянские программы) не содержат в себе встроенного механизма активного распространения по сети, они способны размножаться своими силами только в ***пределах одного компьютера.*** Как правило, передача копии вируса на другой компьютер происходит с ***участием пользователя.*** Например,

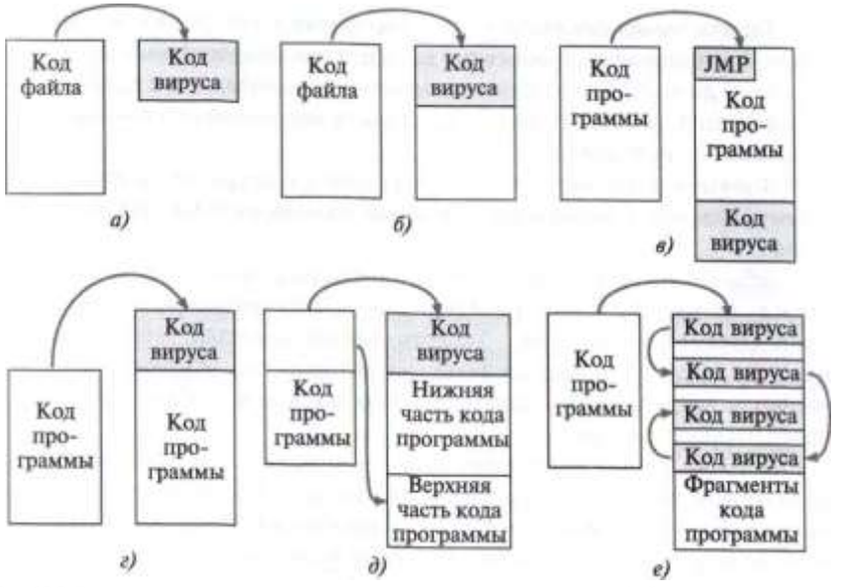


Рис. 19.9. Различные варианты расположения кода вируса в зараженных файлах: *а* — замещение с изменением размера инфицированного файла; *б* — наложение с сохранением размера инфицированного файла: *в* — добавление в конец программы; *г* добавление в начало программы; *д* — добавление с перестановкой частей кода программы; *е* — фрагментарное добавление вируса в тело программы

пользователь может записать свой файл, зараженный вирусом, на сетевой файловый сервер, откуда тот может быть скопирован всеми пользователями, имеющими доступ к данному серверу. Пользователь может также передать другому пользователю съемный носитель с зараженным файлом или послать такой файл по электронной почте. То есть именно пользователь является главным звеном в цепочке распространения вируса за пределы своего компьютера. Тяжесть последствий вирусного заражения зависит от того, какие вредоносные действия были запрограммированы в вирусе злоумышленником. Это могут быть мелкие, но раздражающие неудобства (замедление работы компьютера, уменьшение размеров доступной памяти, трата рабочего времени на переустановку приложений) или серьезные нарушения безопасности, такие как утечка конфиденциальных данных, разрушение системного программного обеспечения, частичная или полная потеря работоспособности компьютерной сети.

Программные закладки

В ГОСТ Р 51275-2006 «Защита информации. Объект информатизации. Факторы, воздействующие на информацию. Общие положения» дается такое определение:

Программная закладка — это внесенные в программное обеспечение функциональные объекты, которые при определенных условиях (входных данных) инициируют выполнение не описанных в документации функций, позволяющих осуществлять несанкционированные воздействия на информацию.

Функции, описание которых отсутствует в документации на программу, называют ***недекларированными возможностями*** (***НДВ***) программ.

Обычно понятие «программная закладка» несет отрицательный смысл, на что указывает прилагательное «несанкционированные» в приведенном определении. То есть подразумевается, что закладка наносит какой-то ущерб системе, на которой она установлена, т. е. является вредоносным кодом, замаскированным в глубине полезного программного продукта.

В то же время недекларированные возможности программы не обязательно являются вредоносными, они могут быть просто дополнительными функциями, которые разработчик программы решил включить для отладки, но не стал их описывать для рядовых пользователей. Это могут быть и просто забытые функции, особенно если речь идет о большой программной системе, в разработке которой участвовали десятки программистов. Примером НДВ, не являющейся закладкой, может служить способность библиотечной функции printf языка С распечатывать содержимое стека при использовании ее с одним параметром — мы рассмотрели это применение выше в разделе «Уязвимости, связанные с нарушением защиты оперативной памяти».

Существует также класс НДВ программы, внедряемых в нее для развлечения пользователя (и самих программистов тоже) — это так называемые «пасхальные яйца», Easter Eggs. «Пасхальное яйцо» прерывает нормальную работу пользователя, который, возможно, устал рассматривать ячейки таблицы своего документа, и радостно приветствует его интересной картинкой или сообщением, а то и приглашением поиграть в игру. Например, версия Microsoft Excell 95 при выполнении определенной последовательности действий вызывала игру Hall of Tortured Souls, в которой вдоль холла были развешаны портреты программистов, разработавших эту программу. Авторы также наблюдали однажды «пасхальное яйцо» в заставке экрана «Трубы» (3D Pipes) ОС Microsoft Windows — на экране на несколько минут среди труб появился очень симпатичный чайник. Появился, исчез, и больше никогда мы его не видели, хотя «Трубы» долго работали на наших компьютерах.

Программные закладки могут выполнять различную вредоносную работу:

- шпионить за действиями пользователя и передавать эту информацию на определенный сервер (так называемое ***spyware*** — шпионские программы);
- **получать доступ к конфиденциальной информации;**
- **искажать и разрушать данные, и**

этот список можно продолжить.

Антивирусные программы

Антивирусные программы давно стали необходимым атрибутом жизни любого пользователя. На домашних компьютерах работают индивидуальные пакеты антивирусной защиты, а на предприятиях — корпоративные пакеты, состоящие из клиентской программы и сервера, рассылающего обновления.

Несмотря на такие меры, в 2012 году около 30 % компьютеров во всем мире были заражены тем или иным видом вируса (данные компании Panda Security). Неизвестно, каким бы был этот процент, если бы на компьютерах не работали антивирусные программы, но хочется верить, что они эффективны и отражают большое количество угроз.

Антивирусные программы используют различные методы для обнаружения вредоносных кодов в файлах, сообщениях электронной почты или HTML-страницах.

Метод сигнатур. Вирус (будем так обобщенно называть далее любой вредоносный код) определенного типа имеет характерную последовательность программных кодов, которая его с какой-то степенью вероятности идентифицирует. Эта последовательность кодов называется *сигнатурой* (подписью) вируса. Для того чтобы обнаружить вирус, антивирусная программа должна иметь библиотеку сигнатур. Постоянное обновление этой библиотеки является одной из самых главных проблем любой компании, выпускающей антивирусное программное обеспечение. Злоумышленники постоянно изобретают новые вирусы, поэтому разработчики антивирусных программ стараются ненамного отстать от злоумышленников и своевременно пополнять библиотеку сигнатур. Сервер корпоративной антивирусной системы периодически рассылает обновленные версии такой библиотеки своим клиентам.

Метод сигнатур является основным методом обнаружения вирусов, но он обладает принципиальным недостатком — ***неспособностью обнаружить новый тип вируса.***

Кроме того, разработчики вирусов используют различные приемы, с помощью которых вирусы могут маскировать свои сигнатуры, что приводит к нераспознаванию вируса. Изменить сигнатуру вируса можно применив:

- *полиморфический код,* когда код изменяет сам себя во время выполнения, что, естественно, приводит к тому, что у него нет постоянной сигнатуры;
- *олигоморфический код,* который собирает вирус из большого количества частей кода; в этом случае все же можно распознать вирус, составив библиотеку сигнатур каждой части;
- *метаморфический код,* когда код транслирует свой двоичный код в некоторое иное представление, например язык высокого уровня, изменяет его, а затем транслирует это представление опять в двоичный код. При этом изменяется также та часть кода, которая выполняет метаморфические изменения, что делает такой код более сложным для распознавания, чем полиморфический, у которого код, выполняющий изменения вируса, сам остается неизменным.

Эвристические методы. Эта группа методов является более интеллектуальной, она не занимается простым сравнением инспектируемого кода с большим количеством заранее отобранный сигнатур, а пытается выявить вирус на основе структуры его кода или его поведения, не имея точной сигнатуры кода, но используя некоторые обобщенные признаки подозрительной структуры кода или подозрительного поведения.

Анализ структуры кода является *статическим* эвристическим анализом кода, а поведения — *динамическим.*

Для безопасного анализа поведения анализируемой программы она помещается в изолированную виртуальную среду, например в среду отдельной виртуальной машины или же созданной программной «песочйицы»42, ограждающей систему от опасных действий программы. В этом случае действия вируса не могут причинить вред основной операционной среде компьютера.

Помещение анализируемой программы в специальную защищенную среду является затратным как по ресурсам, так и по времени. Существует более эффективный хотя и более рискованный подход, когда анализируемой программе разрешают пробное выполнение в рабочей среде, но при этом антивирусное программное обеспечение следит за всеми ее действиями и, в случае необходимости, блокирует их, не давая нанести ущерб рабочей среде.

При обнаружение вируса антивирусная программа помещает зараженную программу в карантин и уведомляет пользователя об этом, который принимает решение об удалении зараженной программы или же, если это возможно, удалении из нее вируса.

Антивирусные программы работают в пространстве ядра, поэтому сами могут причинить ущерб операционной системе из-за своих ошибок. Зафиксированы случаи, когда под видом

42 Песочница (sandbox) — представляет собой механизм жёсткого контроля набора ресурсов (оперативной памяти, места на диске и др.) и системных сервисов, доступных подозрительной программе.

антивирусной программы пользователям предлагалось вредоносное программное обеспечение.

Ботнет

Бот — это программа, которая выполняет некоторые автоматические (часто — интеллектуальные) действия по командам от удаленного центра управления. Бот — это программный робот, который старается отреагировать на возникающую ситуацию некоторыми действиями — протоколированием сообщений (полезный бот, ведет архив чатов), отправкой сообщений — например, поддержанием «разговора» с удаленным собеседником или же участием в DDOS-атаке на какой-то сайт или сеть. Бот может, например, распознавать определенный, заданный ему «хозяином» контекст в дискуссии пользователей социальных сетей Интернета (Livejournal, Facebook) и стать ее участником, выдавая те или иные сообщения. Бот обычно находится в следящем режиме, анализируя сообщения и ожидая команды из центра управления или возникновения заранее определенной ситуации. Есть много похожего у программного бота и механикоэлектронного робота — оба запрограммированы на восприятие информации из окружающего мира, анализа ее на предмет обнаружения определенной ситуации и выполнения заранее запрограммированных действий, только набор действий бота ограничен отправкой сообщений либо в сеть, либо операционной системе, в среде которой он находится.

Боты проникают в удаленные компьютеры нелегально, как вирусы, черви или троянские кони. Пользователь может не знать, что его компьютер заражен ботом, если он не использует хорошее антивирусное программное обеспечение, потому что самому компьютеру бот не причиняет вреда, его цели находятся где-то в Интернете. Обычно злоумышленник заражает кодом бота несколько компьютеров, используя различные известные уязвимости ОС и приложений, а затем код бота, подобно сетевому червю, пытается заразить как можно больше машин. Зараженная ботом машина обычно называется *«зомби».*

Боты чаще управляются централизовано, из одного или нескольких центров, являющихся серверами сети ботов. Такую сеть называют ***ботнетом*** *(botnet).* Возможны и более сложные зависимости между ботами одной сети, с иерархическими или одноранговыми схемами взаимодействия.

Для управления ботами центр управления использует различные протоколы, одним из наиболее распространенных является протокол ***IRC*** (Internet Relay Chat), позволяющий передавать мгновенные сообщения (чат).

Так как «хозяин» ботнета точно не знает, какие именно машины оказались зараженными кодом бота, то для распознавания компью- теров-

зомби, используются методы сетевого сканирования, например сканирование портов, если код бота слушает определенный порт TCP.

Результатом заражения кодом бота может оказаться внушительная армия, включающая десятки тысяч зараженных машин и способная организовать очень мощную DDoS-атаку, распространить большое количество спама или собрать персональные данные, включая банковские реквизиты, большого количества пользователей.

Ботнет может работать как наемная армия — ее хозяин может предоставлять услуги своей сети третьим лицам.

Одним из наиболее свежих инцидентов, связанных с пресечением вредоносной деятельности ботнета, была операция, проведенная компанией Microsoft совместно с ФБР в июне 2013 года по разрушению центров управления ботнетами, зараженными вирусом Citadel. Этот вирус фиксирует нажатия клавиш на компьютере и передает информацию в свой центр управления. В результате проведенной операции было выявлено и разрушено 1462 центра управления, каждый из которых контролировал свой ботнет Citadel. По предварительным данным, эти сети причинили ущерб около 5 миллионам пользователей на сумму свыше полмиллиарда долларов.

Вопросы к главе 19

1. Сколько ошибок, открывающих возможность для взлома системы, может содержать 300 000 строк кода программы согласно данных статистики?

а) от б до 9;

б) 3;

в) более 1000;

г) от 100 до 200;

Д) 30.

2. Атака Ping смерти использует уязвимость:

а) переполнение стека;

б) страничное прерывание;

в) переполнение буфера памяти.

3. Стек реализует стратегию записи:

а) первым пришел, первым вышел;

б) последним пришел, первым вышел;

в) последним пришел, последним вышел.

4. Какую цель обычно преследует злоумышленник, используя эффект переполнения стека:

а) испортить локальные переменные функции ядра ОС;

б) поместить в стек вредоносный код и передать ему управление;

в) исчерпать оперативную память компьютера.

5. Каким образом можно защитить ОС от атак, эксплуатирующих переполнение стека?

а) применять языки программирования, автоматически контролирующие защиту памяти, такие как Java или C#;

б) применять расширения компиляторов, встраивающих коды проверки целостности стека;

в) применять антивирусные программы.

6. Для какой цели злоумышленник добавляет «- -» к концу следующей строки атаки на SQL базу данных:

select * from client where name = 'Alica' or 1=1

а) для подавления операторов контроля вводимых данных;

б) для инструктирования базы данных о том, что запрос поступает от анонимного пользователя и не требует ввода пароля;

в) для вызова из стека кода вредоносной программы.

7. Какие источники входных данных должны входить в периметр доверия программы, работающей на внутреннем сервере предприятия?

а) данные от программ, работающих на других внутренних серверах предприятия;

б) данные от пользователей, работающих на компьютерах предприятия, подключенных к внутренней сети предприятия;

в) данные от программ, работающих в сетях предприятий-смежников;

г) данные от внешних веб-серверов.

8. Чем скрытый канал (covert channel) отличается от стеганографического канала?

а) скрытый канал использует для передачи данных механизм, не предусмотренный разработчиком информационной системы для передачи информации, в то время как стеганографический канал «прячет» данные внутри легального канала;

б) скрытый канал «прячет» данные внутри легального канала, в то время как стеганографический канал использует для передачи данных механизм, не предусмотренный разработчиком информационной системы для передачи информации;

в) скрытый канал шифрует данные, в то время как стеганографический канал нет.

9. Какой механизм ОС эксплуатируют многие типы скрытых каналов?

а) механизм разделения ресурсов между процессами;

б) механизм виртуальной памяти;

в) механизм прерываний.

10. Какой легальный канал чаще всего используют современные стеганографические сообщения?

а) защищенный канал SSL;

б) защищенный канал IPSec;

в) передачу цифровых изображений.

11. Справедливо ли следующее утверждение: «Сетевой червь, попав каким-то образом в компьютер, старается заразить как можно больше других компьютеров, проникая в них по сети, в то время как вирус старается заразить как можно больше файлов в пределах того же компьютера»?

а) да;

б) нет.

12. Вредоносны ли программные закладки?

а) да, все программные закладки вредоносны;

б) некоторые из них вредоносны;

в) закладки Easter Eggs никогда не вредоносны.

13. Справедливо ли следующее утверждение: «Вредоносный код типа «Троянский конь» выполняет полезную работу»?

а) да;

б) да, но при этом он также выполняет работу, целью которой является нанесение ущерба;

в) нет.

14. Антивирусная программа, работающая по методу сигнатур, может:

а) выявлять только известные вредоносные коды;

б) выявлять только вирусы, но не черви;

в) выявлять троянские кони;

г) выявлять только новые виды вредоносных кодов.

15. Ботнет — это;

а) сеть управления телекоммуникационными устройствами;

б) сеть зараженных компьютеров, подчиняющаяся удаленному центру управления;

в) сеть, объединяющая микропроцессорные устройства группы роботов.

20 *Безопасность веб-сервиса*

Веб-сервис занимает особое место в семействе сервисов Интернета, так как именно благодаря этому сервису, изобретённому Тимом Бернерс-Ли и Робертом Кайо в 1989 году, Интернет стал тем, чем мы его знаем сегодня.

Кроме того, веб-браузер с его графическим интерфейсом является основным средством доступа пользователя к большинству сервисов Интернета: сайтам новостей, разнообразным справочникам, библиотекам, Интернет-магазинам, онлайн-банкам, социальным сетям, таким как Facebook, Twitter, LiveJournal, Вконтакте, облачным хранилищам информации и приложениям. Даже такие консервативные устройства, как сетевые маршрутизаторы и коммутаторы, стали поддерживать административный доступ с помощью веб-интерфейса, хотя справедливости ради нужно сказать, что это в основном относится к домашним маршрутизаторам, которые рассчитаны на администрато- ранеспециалиста, которому веб-интерфейс представляется гораздо более удобным, чем командная строка.

Поэтому справедливым будет сказать, что основная часть информации поступает в клиентский компьютер через веб-браузер, а как мы уже ранее отмечали, именно вводимые данные представляют собой главную угрозу для программного обеспечения компьютера. Через веб-браузер попадают в ваш компьютер большинство вредоносных кодов, таких как вирусы, черви и троянские кони, а также назойливые программки, размещающие рекламные объявления на просматриваемой странице без вашего согласия. Вредоносную информацию поставляют браузеру веб-сайты, которые вы посещаете. При поиске информации, когда нужно быстро просмотреть большое количество сайтов, трудно бывает понять, какой из веб-сайтов заслуживает доверия, а какой — нет. Может случится и так, что веб-сервер, на котором работает вполне респектабельный и заслуживающий доверия веб-сайт, оказывается взломанным и его посещение грозит неприятными последствиями.

Из сказанного ясно, что для защиты программного обеспечения защита веб-службы является делом первостепенной важности.

Мы рассмотрим сначала основные элементы веб-службы, а затем обсудим их безопасность.

Организация веб-сервиса

Веб- и HTML-страницы

Миллионы компьютеров, связанных через Интернет, хранят невообразимо огромные объемы информации, представленной в виде веб-страниц.

Веб-страница, или ***веб-документ,*** как правило, состоит из основного HTML-файла и некоторого количества ссылок на другие объекты разного типа: JPEG- или GIF-изображения, другие HTML-файлы, аудио- или видеофайлы, скрипты, написанные на каком-либо языке программирования.

HTML-страница, или HTML-файл, или ***гипертекстовая страница,*** содержит текст, написанный на языке ***HTML (HyperText Markup Language*** — язык разметки гипертекста). История появления этого языка связана с попытками программистов разработать средство, которое бы позволяло им программным путем создавать красиво сверстанные страницы для просмотра на экране. Другими словами, красивая картинка появляется на дисплее только в результате ее интерпретации специальной программой, а в исходном виде она представляет собой однообразный текст с множеством служебных пометок. Вместо применения различных приемов форматирования, таких как выделение заголовков крупным шрифтом, а важных выводов — курсивным или полужирным начертанием, создатель документа на языках этого типа просто вставляет в текст соответствующие указания о том, что данная часть текста должна быть выведена на экран в том или ином виде. Служебные пометки такого рода в исходном тексте выглядят, например, как <Ь> < /Ь> (начать и закончить вывод текста полужирным начертанием) и называются ***тегами (teg).*** Язык HTML не является первым языком разметки текста, его предшественники существовали задолго до появления веб-службы, например в первых версиях ОС Unix существовал язык troff (с помощью этого языка отформатированы страницы электронной документации Unix, известные как man-страницы).

В язык HTML включены разные типы тегов, команд и параметров, в том числе для вставки в текст изображений (тег < img src= '...'). Чтобы HTML-страница выглядела так, как задумал программист, она должна быть выведена на экран специальной программой, способной интерпретировать язык HTML. Такой программой является уже упоминавшийся веб-браузер.

Существует особый тип тега, который имеет вид <a href="..." . . . < / a > и называется ***гиперссылкой.*** Гиперссылка содержит информацию об веб-

43 URL-адреса с самого начала предназначались не только для веб-

странице или объекте, который может находиться как на том же компьютере, так и на других компьютерах Интернета. Отличие гиперссылки от других тегов состоит в том, что элемент, описываемый ею, не появляется автоматически на экране, вместо этого на месте тега (гиперссылки) на экран выводится некоторое условное изображение или особым образом выделенный текст — имя гиперссылки. Чтобы получить доступ к объекту, на который указывает эта гиперссылка, пользователь должен «щелкнуть» на ней, дав тем самым команду браузеру найти и вывести на экран требуемую страницу или объект. После того как новая веб-страница будет загружена, пользователь может перейти по следующей гиперссылке — такой «веб-серфинг» может продолжаться теоретически сколь угодно долго. Все это время веб-браузер будет находить указанные в гиперссылках страницы, интерпретировать все размещенные на них указания и выводить информацию на экран в том виде, в котором ее спроектировали разработчики этих страниц.

Адрес URL

Браузер находит веб-страницы и отдельные объекты по адресам специального формата, называемым ***URL (Uniform Resource Locator — унифицированный указатель ресурса).*** URL-адрес может выглядеть, например, так: http://www.olifer.co.uk/books/books.htm.

В URL-адресе можно выделить три части:

Тип протокола доступа. Помимо HTTP, здесь могут быть указаны и другие протоколы, такие как FTP, telnet, также позволяющие осуществлять удаленный доступ к файлам или компьютерам43. Тем не менее, основным протоколом доступа к веб-страницам является HTTP (как в нашем примере), и мы поговорим о нем немного позже.

DNS-имя сервера. Имя сервера, на котором хранится нужная страница. В нашем случае — это имя сайта www.olifer.co.uk.

Путь к объекту. Обычно это составное имя файла (объекта) относительно главного каталога веб-сервера, предлагаемого по умолчанию. В нашем случае главным каталогом является /books/books.htm. По расширению файла мы можем сделать вывод о том, что это HTML-файл.

Веб-клиент и веб-сервер

Как мы уже отмечали, сетевая веб-служба представляет собой распределенную программу, построенную в архитектуре клиент-сервер.

служб, но и для других сервисов доступа к информации через Интернет.

Клиент и сервер веб-службы взаимодействуют друг с другом по протоколу HTTP.

Клиентская часть веб-службы, или ***веб-клиент,*** называемый также ***браузером,*** представляет собой приложение, которое устанавливается на компьютере конечного пользователя. Одной из важных его функций является поддержание графического пользовательского интерфейса. Через этот интерфейс пользователь получает доступ к широкому набору услуг, главная из которых, конечно, — «веб-серфинг», включающий поиск и просмотр страниц, навигацию между уже просмотренными страницами, переход по закладкам и хранение истории посещений. Помимо средств просмотра и навигации, веб-браузер предоставляет пользователю возможность манипулирования страницами: сохранение их в файле на диске своего компьютера, вывод на печать, передача по электронной почте, контекстный поиск в пределах страницы, изменение кодировки и формата текста, а также множество других функций, связанных с представлением информации на экране и настройкой самого браузера.

К числу наиболее популярных сейчас браузеров можно отнести Microsoft Internet Explorer, Mozilla Firefox компании Mozilla, Google Chrome и Apple Safari. Веб-браузер — это не единственный вид клиента, который может обращаться к веб-серверу. Эту роль могут исполнять любые программы и устройства, поддерживающие протокол HTTP.

Значительную часть своих функций браузер выполняет в тесной кооперации с веб-сервером. Как уже было сказано, клиент и сервер веб-службы связываются через сеть по протоколу HTTP. Это означает, что в клиентской части веб-службы присутствует клиентская часть HTTP, а в серверной — серверная часть HTTP.

Веб-сервер — это программа, хранящая объекты локально в каталогах компьютера, на котором она запущена, и обеспечивающая доступ к этим объектам по URL-адресам. Наиболее популярными вебсерверами сейчас являются Apache и Microsoft Internet Information Server.

Как и любой другой сервер, веб-сервер должен постоянно находиться в активном состоянии, прослушивая TCP-порт 80, который является назначенным портом протокола HTTP. Как только на этот порт приходит запрос от клиента, сервер устанавливает TCP-соединение и получает от клиента имя объекта, например, в виде /books/books.htm, после чего находит в своем каталоге этот файл, а также другие связанные с ним объекты и отсылает их по TCP-соединению клиенту. Получив объекты от сервера, веб-браузер отображает их на экране (рис. 20.1). После отправки клиенту всех объектов страницы сервер

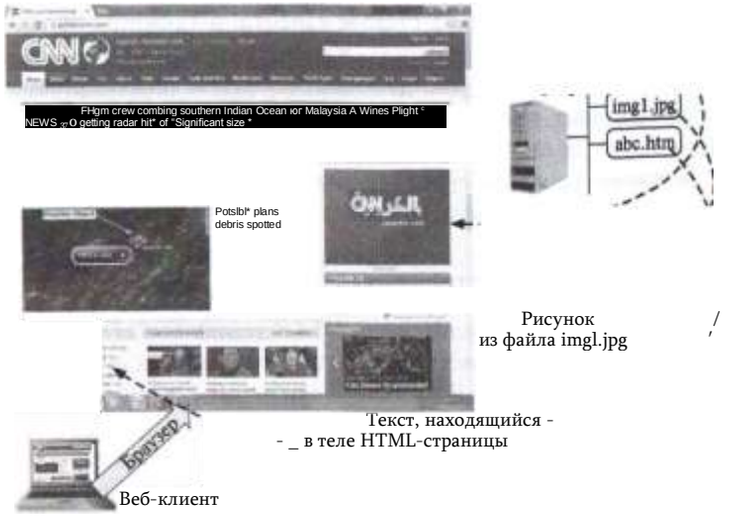


Рис. 20.1. Отображение веб-страницы

разрывает с ним TCP-соединение. В дополнительные функции сервера входят также аутентификация клиента и проверка прав доступа данного клиента к данной странице.

Веб-сервер в отношении сессии с веб-браузером является сервером stateless — без сохранения состояния. Это означает, что на сервере не хранится информации, отслеживающей состояние сессии — какие страницы пользователь уже посетил и какая информация ему была передана. Такой режим общения с клиентом упрощает организацию сервера, которому необходимо отвечать на большой поток запросов различных пользователей, так что запоминание состояния сессий пользователей существенно увеличило бы нагрузку на веб-сервер. Вместо этого веб-сервер рассматривает каждый запрос изолированно, отвечая на него и забывая про данного пользователя сразу после ответа. Кроме упрощения организации сервера, такой режим работы является более устойчивым, так как он не требует восстановления сессии, если по той или иной причине она потерпела крах, оставляя пользователю решать эту проблему. Недостатком данного режима является замедление работы клиента и увеличение трафика в сети из-за частого выполнения процедуры установления TCP-соединений.

Для повышения производительности некоторые веб-серверы прибегают к кэшированию последних наиболее часто используемых страниц в своей памяти. Когда приходит запрос на какую-либо страницу, сервер, прежде чем считывать ее с диска, проверяет, не находится ли она в буферах более «быстрой» оперативной памяти. Кэширование

страниц осуществляется и на стороне клиента, а также на промежуточных серверах (прокси-серверах). Кроме того, эффективность обмена данными с клиентом иногда повышают путем компрессии (сжатия) передаваемых страниц. Объем передаваемой информации уменьшают также за счет того, что клиенту передается не весь документ, а только та часть, которая была изменена. Все эти приемы повышения производительности веб-службы реализуются средствами протокола HTTP.

Протокол HTTP

Протокол передачи гипертекста HTTP (HyperText Transfer Protocol) — это протокол прикладного уровня, во многом аналогичный протоколам FTP и SMTP. В настоящее время используются две версии протокола HTTP/1.0 и HTTP/1.1.

Обмен сообщениями идет по обычной схеме «запрос-ответ». Клиент и сервер обмениваются ***текстовыми*** сообщениями стандартного формата, т. е. каждое сообщение представляет собой нескольких строк обычного текста в кодировке ASCII.

Для транспортировки HTTP-сообщений служит протокол TCP. При этом TCP-соединения могут использоваться двумя разными способами:

- ***долговременное соединение —*** передача в одном TCP-соединении нескольких объектов, причем время существования соединения определяется при конфигурировании веб-службы;
- ***кратковременное соединение*** — передача в течение одного TCP-соединения только одного объекта.

Долговременное соединение в свою очередь может быть использовано двумя способами:

- ***последовательная передача запросов с простоями*** — новый запрос посылается только после получения ответа;
- ***конвейерная передача*** — это более эффективный способ, в котором следующий запрос посылается до прибытия ответа на один или несколько предыдущих запросов. По умолчанию степень параллелизма устанавливается на уровне 5-10, но у пользователя имеется возможность изменять этот параметр при конфигурировании клиента.

В версии HTTP 1.0 поддерживается только режим кратковременных соединений, когда после передачи одного запроса и получения ответа соединение TCP закрывается. Такой режим полностью соответствует концепции сервера без сохранения состояния, а это, как было сказано, приводит к замедлению работы браузера и увеличению трафика из-за частого выполнения процедуры трехэтапного установления TCP-соединения.

В версии HTTP 1.1 по умолчанию применяются постоянные соединения и конвейерный режим. Соединение разрывается по инициативе либо браузера, либо сервера за счет отправки специального токена разрыва соединения в HTTP-пакете. Веб-сервер обычно использует таймер неактивности пользователя для того, чтобы разорвать соединение по тайм-ауту и не тратить ресурсы памяти на неактивные соединения.

Формат HTTP-сообщений

В протоколе HTTP все сообщения состоят из текстовых строк. Сообщения HTTP бывают двух типов — запросы и ответы. Запросы и ответы имеют единую обобщенную структуру, состоящую из трех частей: обязательной стартовой строки, а также необязательных заголовков и тела сообщения. В табл. 20.1 приведены форматы и примеры стартовых строк и заголовков для запросов и ответов.

Как видно из таблицы, запросы и ответы имеют разные форматы стартовой строки. Каждая из них состоит из трех элементов, вклю-

Таблица 20.1

Форматы стартовых строк и заголовков

Обобщенная структура сообщения	HTTP-запрос	HTTP-ответ
Стартовая строка (всегда должна быть первой строкой сообщения; обязательный элемент)	Формат запроса Метод/ URL HTTP/l.x. Пример: GET /books/books.htm HTTP/1.1	Формат ответа: HTTP/l.x Код состояния Фраза. Пример: HTTP/1.0 200 OK
Заголовки (следуют в произвольном порядке; могут отсутствовать)	Заголовок о DNS-имени компьютера, на котором расположен веб-сервер. Пример: Host: www.olifer. co.uk Заголовок об используемом браузере. Пример: User- agent: Mozilla/5.0 Заголовок о предпочтительном языке. Пример: Accept-language: ru Заголовок о режиме соединения. Пример: Connection: close	Заголовок о времени отправления данного ответа. Пример: Date: 1 Jan 2009 14:00:30 Заголовок об используемом веб-сервере. Пример: Server: Apache/1.3.0 (Unix) Заголовок о количестве байтов в теле сообщения. Пример: Content-Length: 1234 Заголовок о режиме соединения. Пример: Connection: close
Пустая строка		
Тело сообщения (может отсутствовать)	Здесь могут быть расположены ключевые слова Для поисковой машины или страницы для передачи на сервер	Здесь может быть расположен текст запрашиваемой страницы

чающих поле ***версии протокола HTTP.*** В примере и в запросе, и в ответе указана версия HTTP/1.1.

Стартовая строка ***запроса*** включает в себя поле ***метода*** — это название операции, которая должна быть выполнена.

Чаще всего в запросах используется метод GET, т. е. запрос объекта. Именно он включен в наш пример запроса. Еще одним элементом стартовой строки является ***URL-ссылка*** на запрашиваемый объект — здесь это имя файла /books/books.htm.

Помимо этого метода в запросах протокол предусматривает и другие методы, такие как HEAD, POST, PUT, DELETE и некоторые другие.

- Метод HEAD аналогичен методу GET, но запрашиваются только метаданные заголовка HTML-страницы.
- Метод POST используется клиентом для отправки данных на сервер: сообщения электронной почты, ключевых слов в запросе поиска, веб-формы.
- Метод PUT используется клиентом для размещения некоторого объекта на сервере, на который указывает URL.
- Метод DELETE указывает серверу на то, что некоторый объект на сервере, определяемый URL, необходимо удалить.

Методы GET и HEAD считаются безопасными для сервера, так как они только передают информацию клиенту. В то же время методы POST, PUT и DELETE считаются опасными, так как они передают информацию на сервер, при этом наиболее опасными являются два последних метода, потому что они непосредственно указывают на объект на сервере, а значит, при их злоумышленном использовании можно заменить или удалить некоторые объекты на сервере, т. е. атаковать его.

В стартовой строке ***ответа,*** помимо уже упоминавшегося указания на версию протокола HTTP, имеются поле ***кода состояния*** и поле ***фразы*** для короткого текстового сообщения, поясняющего данный код для пользователя.

В настоящее время стандарты определяют пять классов кодов состояния:

- 1xx — информация о процессе передачи;
- ***2xx*** — информация об успешном принятии и обработки запроса клиента (в таблице в примере стартовой строки ответа приведен код и соответствующая фраза 200 ОК, сообщающий клиенту, что его запрос успешно обработан);
- 3xx— информация о том, что 1цля успешного выполнения операции нужно произвести следующий запрос по другому URL-адресу, указанному в дополнительном заголовке Location;
- 4xx — информация об ошибках на стороны клиента (читатель наверняка не раз сталкивался с ситуацией, когда при указании адреса несуществующей страницы браузер выводил на экран сообщение 404 Not Found);
- 5xx — информация о неуспешном выполнения операции по вине сервера (например, сообщение 505 http Version Not Supported говорит о том, что сервер не поддерживает версию HTTP, предложенную клиентом).

Среди кодов состояния имеется код 401, сопровождаемый сообщением authorization required. Если клиент получает такое сообщение в ответ на попытку доступа к странице или объекту, это означает, что доступ к данному ресурсу ограничен и требует авторизации пользователя. Помимо поясняющей фразы сервер помещает в свой ответ дополнительный заголовок www-Authenticate:< ...>, который сообщает клиенту, какую информацию он должен направить серверу для того, чтобы процедура авторизации могла быть выполнена. Обычно это имя и пароль. Веб-клиент с момента получения такого ответа сервера начинает добавлять во все свои запросы к ресурсам данного сервера дополнительный заголовок Authorization: < ***имя, парольУ*** , который содержит информацию, необходимую для авторизации доступа.

Динамические веб-страницы

До сих пор подразумевалось, что содержание страницы не изменяется в зависимости от действий пользователя. Когда пользователь щелкает по гиперссылке, то он переходит на ***новую*** страницу, а если выполняет команду возвращения обратно, то на экране появляется предыдущая страница всегда в ***неизменном*** виде. Такие страницы называются ***статическими.***

Однако в некоторых случаях было бы очень желательно, чтобы содержание страницы изменялось в зависимости от действий пользователя, например при наведении указателя мыши на определенную область страницы там появлялся бы рисунок вместо текста или значка. Динамическое воспроизведение состояния базы данных также является типичным примером ситуации, когда статическая стрэни- ца не может решить задачу. Например, многие Интернет-магазины поддерживают базу данных продаваемых товаров, и вывод количества оставшихся в наличии товаров требует динамического обновления соответствующего поля веб-страницы.

Веб-страницы, которые могут генерировать выводимое на экран содержание, меняющееся в зависимости от некоторых внешних условий, называются ***динамическими.***

Динамика страницы достигается путем ее программирования, обычно для этого используются программные языки сценариев, такие как Perl, PHP или JavaScript.

Различают два класса программ, предназначенных для создания динамического содержания веб-страниц:

- программы, работающие на стороне клиента (т. е. на том компьютере, где запущен веб-браузер, воспроизводящий страницу на экране);
- программы, работающие на стороне сервера.

В том случае, когда программа работает на стороне клиента, код страницы передается веб-сервером веб-браузеру как обычный статический объект, а затем браузер выполняет этот код, с его помощью создает динамическое содержание страницы и выводит ее на экран.

Существуют различные способы создания динамического содержания страницы на стороне клиента.

Расширения браузера. Механизм ***надстроек браузера*** (***add-on***) позволяет расширить его функциональные возможности за счет динамического вызова из браузера дополнительных программ, установленных на клиентском компьютере. Программа-надстройка обрабатывает объекты веб-страницы определенного типа, например программа- надстройка Adobe Acrobat NPAPI Plugin вызывается браузером Firefox для показа пользователю документа в формате PDF в окне браузера, а программа-надстройка Shockwave Flash вызывается для проигрывания видеоклипов в формате Flash или для интерактивной анимации, написанной на языке ActionScript. P1ужно отметить, что термин надстройка (add-on) может считается обобщенным названием различных видов надстроек браузера, как это, например делает браузер Firefox, который различает несколько видов add-ons:

- ***расширения (extensions***), которые встраиваются в браузер (т. е. становятся его частью);
- ***темы,*** которые изменяют внешний вид окна браузера и также встраиваются в него;
- ***вставки*** (***plug-ins***) — программы, оформленные чаще всего в виде библиотек и вызываемые через стандартный для браузера интерфейс, такой как, например, NPAPI (Netscape Plugin Application Programming Interface). Вставки являются внешними по отношению к браузеру программами.

Java-апплеты представляют собой скомпилированные программы на языке Java, которые динамически загружаются браузером с веб-сервера и выполняются виртуальной Java-машиной (JVM) клиентского компьютера. Передача Java-dnплета Java-машине выполняется вставкой Java Applet Plugin.

JavaScript. Динамическое содержание страницы, созданное с помощью скриптов языка ***JavaScript,*** созданного компанией Netscape. Этот способ отличается от предыдущего тем, что JavaScript — это язык интерпретирующего типа, который интерпретирует сам браузер.

JavaScript хотя и имеет общую часть в названии с языком Java, но это отдельный язык со своим синтаксисом, большинство браузеров его поддерживают.

ActiveX. Динамическое содержание страницы может быть также созданы управляющими элементами Microsoft ActiveX, которые могут вызывать внешние объекты и встраивать результаты их работы в страницу. Объекты ActiveX являются двоичными исполняемыми файлами, которые должны иметь цифровую подпись. Ограничением элементов ActiveX является то, что их выполнение возможно только в среде ОС Windows на процессоре Intel x8б или же при их эмуляции. На практике это означает, что страницы с элементами ActiveX будут правильно отображаться только браузером Internet Explorer.

При программировании содержания страницы на стороне сервера процесс выглядит немного сложнее, так как программный код страницы создает содержание на сервере, следовательно, здесь нужен дополнительный этап — передача этого содержания по протоколу HTTP на клиентскую машину браузеру. Популярными языками сценариев для серверной части являются Perl, ASP, JSP и PHP. Существует также стандартный программный интерфейс между веб-сервером и программами, генерирующими динамическое содержание, — это общий шлюзовой интерфейс (Common Gateway Interface, CGI).

Безопасность веб-браузера

Первые браузеры не уделяли большого внимания таким проблемам, как приватность пользователя при посещении веб-сайтов Интернета и безопасность его коммуникаций и данных. За время, прошедшее с появления первого браузера, разработчики браузеров накопили большой опыт в борьбе со злоумышленниками и меню приватности и безопасности современного браузера включает много опций, позволяющих уменьшить риски пользователя (правда, за счет создания некоторых неудобств при просмотре сайтов, так как некоторые из них могут быть по ошибке заблокированы или же почти любой щелчок по элементу страницы будет требовать явного разрешения от пользователя).

Рассмотрим, что же стоит за опциями меню безопасности браузеров, какие риски они стараются уменьшить и за счет каких средств.

Приватность и куки

Популярность Интернета существенно снизила приватность его пользователей. Потенциально все действия пользователя в Интернете— посещенные сайты, просмотренные страницы, запросы поиска — могут быть зафиксированы и проанализированы, и антитеррористи- ческие службы, а также службы маркетинга торговых предприятий

активно этим занимаются. Веб-серверы ведут журналы посещений своих сайтов с запоминанием IP-адресов клиентов и предоставляют эти данные владельцам сайтов в удобной форме. Однако анонимность в этих журналах до какой-то степени сохраняется, особенно если адрес назначен провайдером динамически.

Браузеры также ведут журналы посещения сайтов и страниц. И если на веб-сервере данные о ваших посещениях скорее всего растворились бы в общей статистике, то на вашем компьютере (если он только не разделяется с другими сотрудниками или посетителями кафе или гостиницы) сохраняется история именно ваших посещений и интересов (последнее — в виде запросов к поисковым машинам). Поэтому конфискация компьютера и просмотр журнала истории браузера является теперь одним из первых действий следователя при расследовании дела в отношении подозреваемой личности.

Современный браузер позволяет пользователю достаточно детально управлять журналом истории посещений, который хранит как адреса посещенных сайтов и страниц, так и кэшированные страницы этих сайтов. Например, браузер Firefox v.24 позволяет пользователю полностью отключить ведение истории посещений, а также разрешать/запрещать ведение каждого из двух отдельных разделов журнала (рис. 20.2):

- истории посещений сайтов и загрузки файлов;
- истории запросов и форм.

Кроме того, Firefox, как, впрочем, и другие браузеры, позволяет пользователю управлять приемом куки.

Куки (cookies — печенье)— это небольшой фрагмент текстовых данных, которым обмениваются веб-сервер и браузер. Куки, относящийся к некоторой сессии браузера с сервером, содержит информацию о текущем состоянии этой сессии, аутентификационные данные и персональные настройки клиента, а также уникальный для сервера номер сессии. В течение всей сессии куки сохраняется на стороне браузера.

При установлении соединения сервер генерирует содержимое куки и передает его браузеру. Веб-браузер, получив текст куки от вебсервера, сохраняет его в виде файла. В течение всей сессии пользователя, а, возможно, и при всех повторных обращениях данного пользователя к данному сайту,браузер передает куки серверу в том же виде, в каком он его получил в последнем ответе сервера. Таким образом достигается эффект запоминания состояния сессии, причем состояние запоминается на стороне клиента.

Веб-сервер обычно использует данные из куки пользователя для его же, пользователя, удобства — например, Интернет-магазины обыч-

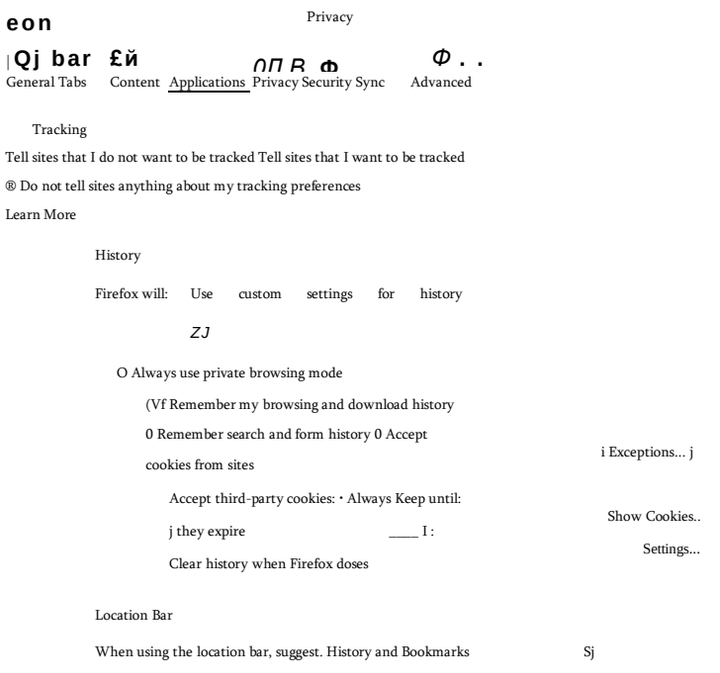


Рис. 20.2. Опции защиты приватности браузера Firefox

но хранят в куки шоппинг-карту пользователя, в них также может храниться история навигации пользователя по страницам сайта.

Типичной информацией, помещаемой веб-сервером в куки, является идентификатор сессии пользователя SID, на основе которого связываются воедино отдельные запросы пользователя. Даже в случае работы по протоколу HTTP 1.1, который использует длительные сессии TCP, эти сессии могут прерываться из-за временной не- активности пользователя, так что соединение отдельных фрагментов сессии, которая представляется пользователю единой, полезно для индивидуального обслуживания пользователя.

На рис. 20.3 показаны куки сайта www.google.co.uk, сохраненные браузером Safari б, — их можно просматривать с помощью опции Web Inspector этого браузера.

Все куки в этом примере — постоянные, они хранятся в файловой системе ОС и имеют длительные сроки действия (показанные в колонке Expires). Существуют также временные куки, которые браузер хранит в оперативной памяти и удаляет после своего закрытия.

Куки имеют не только срок, но и область действия — она задается доменным именем сайта, который создал куки. Браузер не передает

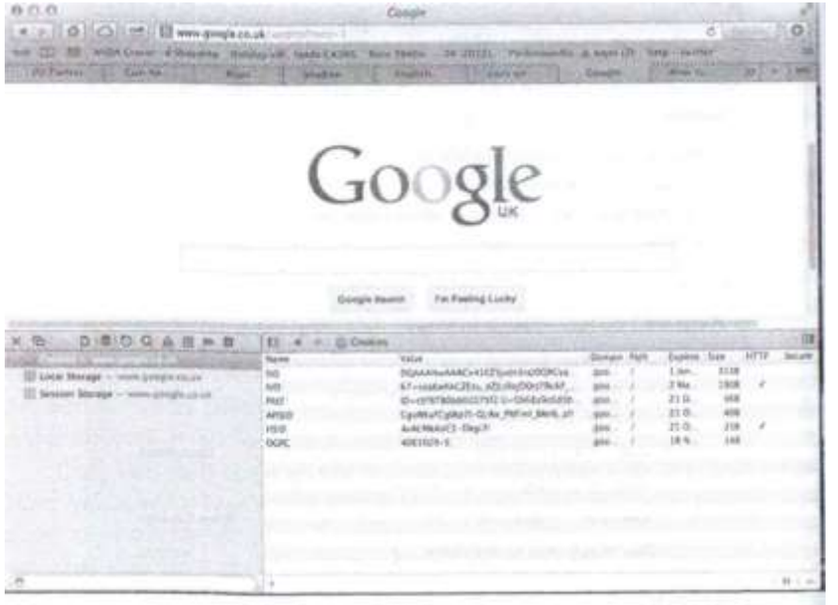


Рис. 20.3. Куки, полученные от сайта google.co.uk

куки сайту с другим доменным именем, но так как доменное имя может быть задано не для конкретного сайта, а для некоторого домена, т. е., например, не для www.cisco.com, а для cisco.com, то куки могут иметь более широкую область действия, чем один сайт.

Так как куки представляют собой текстовые файлы, то угрозы безопасности для пользователя они не представляют (за исключением того случая, когда в них содержится аутентификационная информация пользователя, — этот случай будет рассмотрен в следующем разделе). Вирусы и другие вредоносные коды с помощью куки не распространяются, так что существующее мнение, что куки могут заразить компьютер клиента, неверно.

В то же время куки могут повредить вашей ***приватности,*** особенно если в них помещается чувствительная личная информация — данные ваших шоппинг-карт, формы запросов с вашим именем и фамилией, адресом и так далее. Некоторые сайты используют куки третьих сторон, например рекламных компаний. Браузер запоминает такой куки и предъявляет его не тому сайту, который его сгенерировал, а тому, у которого доменное имя было задано исходным сайтом. Таким образом с помощью куки ваши предпочтения становятся известны большому количеству сайтов.

Самым простым способом защиты своей приватности является полный запрет на прием куки от любых сайтов. Однако при этом вы

Рис. 20.4. Предупреждение сайта о политике использования куки

можете лишиться некоторых сервисов, основанных на использовании куки, например услуг некоторых (не всех) Интернет-магазинов. Поэтому браузеры разрешают задавать индивидуальные списки сайтов, от которых вы запрещаете принимать куки, разрешая это для всех остальных сайтов.

С некоторых пор многие сайты стали предупреждать пользователей о том, что они используют куки. Это связано с изменениями в законодательстве стран Европейского союза и США, ужесточающих защиты личных данных пользователей Интернета. В Евросоюзе такие правила были введены в 2009 году в виде директивы 2009/136/ЕС, называемой также директивой е-приватности. В соответствии с ней все страны Евросоюза должны принять соответствующие поправки к своему законодательству, запрещающие сайтам использовать куки без согласия пользователя.

Поэтому компании начали помещать на своих сайтах предупреждения об использовании куки, сопровождаемые подробные пояснения своей политики в отношении их использования (например, обязательство не передавать данные о клиенте третьей стороне). Например, компания Philip Morris International на своем российском сайте выдает при открытии его страниц предупреждение, показанное на рис. 20.4.

Это предупреждение дает ссылку на объяснение политики использования куки, а также просит нажать на кнопку Cookie Consent, которая показывает, какие виды куки вы разрешаете использовать.

PHILIP MORRIS INTERNATIONAL

Our website uses cookies so that we can remember you and understand how visitors use bur alto. This tool gives you control over the information ws collect, if you would like to read more about our use of cookies, please view our Cc®Ш.ЩМ-

Cookie Name	Cookie Purpose	Opt-out option LssI ySSCv*
First Party and Essentials		
GCWLanguage	This cookie will remember which language you chose to view the site in.	No opt-out
Third Party (Analytics)		
Google Analytics	Analytics and Measurement.	Access opt-out site

Рис. 20.5. **Форма выбора типа куки, на которые пользователь дает согласие**

Нажатие этой кнопки показывает вам форму, описывающую типы куки, которые использует сайт и на использование которых вы может дать согласие (рис. 20.5). Данный сайт использует только два типа куки — собственные куки и куки третьей стороны Google Analytics, которые Google собирает для анализа посещаемости сайтов. Из экранной формы видно, что пользователь может отказаться от приема Google Analytics куки, но не может отказаться от приема куки данного сайта, если он решит продолжать его использовать.

Безопасность коммуникаций браузера и протокол HTTPS

Веб-браузер для взаимодействия с веб-сервером по умолчанию использует протокол HTTP без дополнительных мер по обеспечению основных свойств безопасных коммуникаций, т. е. аутентификации сторон, а также конфиденциальности, доступности и целостности данных. Естественно, это создает значительные риски безопасности при работе с сайтами Интернета.

Атаки вида «человек посередине» вполне возможны при перехвате злоумышленником незащищённых HTTP пакетов, циркулирующих между веб-браузером и веб-сервером.

Атака «захвата сессии» является разновидностью атаки «человек посередине»; она возможна в том случае, когда пользователь аутентифицируется на веб-сервере с помощью своего имени и пароля, а затем веб-сервер использует куки, передаваемые в сообщениях браузера, как свидетельство того, что очередной запрос пришел от аутентифицированного пользователя, и продолжает сессию без повторного

запроса пароля. Понятно, что такой способ аутентификации пользователя в случае множественных сессий протокола HTTP 1.0 или разрыва по какой-то причине длительной сессии протокола HTTP 1.1 предоставляет злоумышленнику хорошую возможность для захвата сессии. Для этого ему достаточно перехватить запрос HTTP, содержащий куки, и затем посылать свои запросы от имени легального пользователя на соответствующий веб-сервер.

Другой разновидностью атаки «человек посередине» является ***атака «повторение»,*** когда злоумышленник повторяет перехваченные запросы легального пользователя, возможно несколько модифицируя их. Например, перехватив запросы сессии пользователя с его банком злоумышленник может инициировать повторный перевод денег, но теперь уже на свой счет.

В предыдущих примерах злоумышленник использовал уязвимости процесса аутентификации пользователя. Очевидно, что прослушивание открытого трафика между браузером и веб-сервером может также нарушить конфиденциальность данных, а также их целостность, если злоумышленник по какой-то причине внесет какие-то изменения в них. Злоумышленник может также нарушить доступность данных, просто отбрасывая ответы веб-сайта.

Основным способом обеспечения надежной аутентификации пользователей веб-сайтов, а также конфиденциальности, целостности и доступности данных, циркулирующих между веб-браузером и вебсайтом, является использование протокола ***безопасный протокол передачи гипертекста HTTPS*** (***Hypertext Transfer Protocol Secure***) вместо HTTP.

Написав «протокол HTTPS», мы погрешили против истины — такого протокола нет, а есть комбинация протоколов HTTP и протокола SSL (или TLS в его стандартном названии). Другими словами, в этом варианте протокол HTTP работает поверх протокола SSL, который и обеспечивает безопасность соединения. Тем не менее, название HTTPS прижилось и пользователь должен его употреблять, когда он хочет инициировать защищенное соединение с веб-сервером, например, задавая

https://www.cisco.com.

Соединение HTTPS использует по умолчанию порт 443 вместо порта 80 соединения HTTP. При использовании соединения HTTPS сам протокол HTTP, работающий поверх протокола SSL, остается неизменным, все свойства безопасности коммуникаций обеспечиваются протоколом SSL, который был рассмотрен в части II.

Аутентификация. Как вы помните, аутентификация в протоколе SSL основана на цифровых сертификатах. При обращении веб-

браузера к веб-серверу по протоколу HTTPS каждая из сторон должна иметь подписанный центром сертификации сертификат, достоверность которого можно проверить по цепочке доверия, ведущей к одному из доверенных корневых центров сертификации.

Производители с каждой копией своего браузера поставляют действительный цифровой сертификат, который может быть использован для аутентификации данного браузера. Этот сертификат, называемый ***встроенным,*** не аутентифицирует пользователя, работающего с браузером, он используется только для создания защищенного канала при передаче данных между браузером и веб-сервером.

Пользователь может запросить ***личный цифровой сертификат*** у некоторого центра сертификации и установить его соответствующим образом в своей операционной системе, указав, что он должен использоваться для логического входа. В таком случае вход в веб-сервер, требующий аутентификации, может происходить не на основе имени и пароля пользователя, а с помощью его личного сертификата, поставляемого браузером серверу по запросу последнего (сервер должен быть сконфигурирован соответствующим образом для того, чтобы запрашивать у браузера личный сертификат пользователя). Но основной практикой работы браузера с сервером по протоколу HTTPS является использование встроенных сертификатов браузера, которые пользователя не аутентифицируют.

Аутентификация сервера при установлении HTTPS соединения всегда выполняется на основе его индивидуального сертификата, получаемого владельцем сервера. Этот сертификат свидетельствует о том, что он выдан для веб-сервера с определенными (одним или несколькими) доменными именами.

Веб-браузер может отказаться устанавливать HTTPS соединение с веб-сервером, если проверка сертификата сервера показала его недействительность. Сертификат сервера может быть признан недействительным, если:

- выдавший его сертификационный центр не является доверенным, т. е. не входит в иерархию ни одного из доверенных корневых центров;
- срок действия сертификата истек;
- имя сервера, набранного пользователем, не совпадает ни с одним из доменным именем сервера, фигурирующим в сертификате. Браузер обычно уведомляет пользователя о том, что сертификат сервера по какой-то причине является недействительным, оставляя на усмотрение пользователя окончательное решение — отказаться от соединения или все же установить его. Иногда сложный механизм проверки аутентичности сервера работает вхолостую, потому что пользователи недооценивают угрозы со стороны «невыясненных»

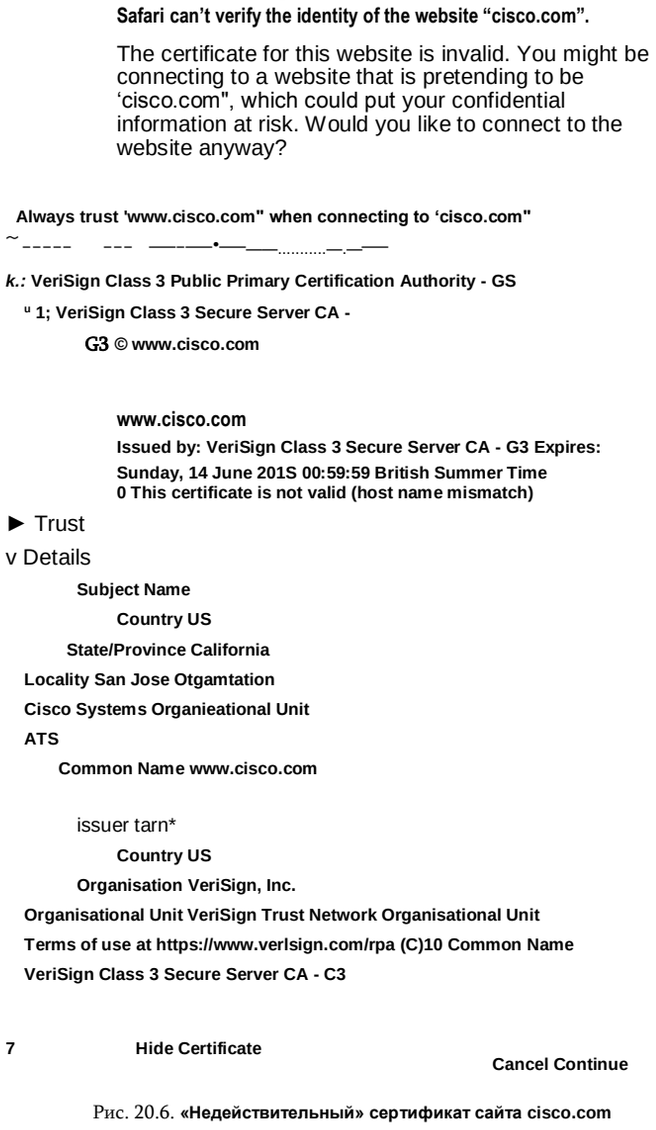


Рис. 20.6. **«Недействительный» сертификат сайта cisco.com**

веб-серверов и предпочитают действовать на свой страх и риск, лишь бы получить заветную программу или информацию.

Ситуацию с недействительным сертификатом веб-сервера легко промоделировать в учебных целях, чтобы посмотреть на поведение своего браузера в таких ситуациях. Вы можете проделать это, задав имя какого-нибудь веб-сервера, поддерживающего HTTPS соедине-

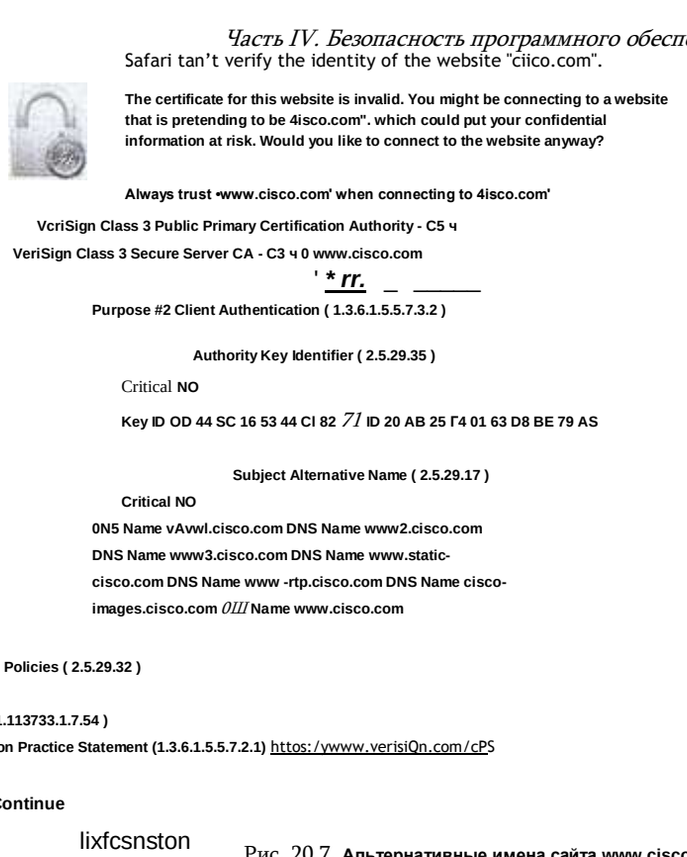


Рис. 20.7. **Альтернативные имена сайта www.cisco.com**

ния, в усеченном виде, т. е. без имени самого сайта, а только его старшую часть, относящуюся к имени домена. Например, мы сделали это с сайтом компании Cisco, обратившись к нему так: https://cisco.com.

Результат такого запроса показан на рис. 20.6. Браузер Safari предупреждает, что сайт претендует на то, чтобы быть сайтом «cisco, сот», хотя и имеет другой имя, www.cisco.com. Действительно, видно, что компания Cisco получила сертификат на имя www.cisco.com, которое фигурирует в параметре Common Name. Дальнейший просмотр сертификата показывает, что вообще-то специалисты Cisco понимали, что пользователи могут интересоваться различными сайта

ми Cisco, имеющими имена, отличные от www.cisco.com, и поэтому поместили их в расширение сертификата, описывающее альтернативные имена сайта (рис. 20.7). В этот список попали: wwwl.cisco.com, www2.cisco.com, www3.cisco.com, www.static-cisco.com, www.rtp.cisco, com и cisco-images.cisco.com, однако имени cisco.com среди них нет, поэтому браузер и не признал сертификат за действительный.

Конфиденциальность и целостность. После фазы аутентификации сервер и браузер вырабатывают разделяемый ключ сессии, который используется для обеспечения конфиденциальности данных (шифрование) и их целостности (цифровая подпись).

Обмен данными между браузером и сервером по протоколу HTTPS намного снижает риск атак «человек посередине» и любых попыток перехватить или исказить конфиденциальные данные на пути между клиентом и сервером. Поэтому все веб-сайты, требующие высокой степени конфиденциальности и целостности коммуникаций, — банковские сайты, сайты Интернет-магазинов, сайты агентств по продаже туров и авиабилетов и т. п. — в ходе сессии с пользователем в определенной ее точке, когда такого рода свойства сессии становятся критически важными, переходят на HTTPS-соединение.

Вообще, хорошей практикой безопасных коммуникаций является использование протокола HTTPS по умолчанию в расчете на то, что нужный вам сайт его поддерживает. Некоторые сайты переходят на него автоматически, даже если пользователь и набрал http:// в своем запросе. Так, например, все сайты компании Google переадресовывают пользователя с http://www.google.ru на https://www.google.ru.

Безопасность средств создания динамических страниц

Современные браузеры поддерживают разнообразные средства создания динамических страниц. Все они представляют собой программные коды, полученные извне, и, следовательно, несут риски, связанные с несанкционированным воздействием на клиентский компьютер, начиная с чтения конфиденциальных данных и удаления файлов пользователя до разрушения операционной системы.

Вставки и надстройки. По сути это программы, которые вы устанавливаете на своем компьютере. Поэтому нужно очень серьезно относится к любому предложению веб-сайта установить новую надстройку или вставку для того, чтобы, например, лучше проигрывать видеоклип определенного формата или же быстрее загружать файлы. Очень может быть, что помимо своей основной функции такая программа будут заниматься еще и какой-то побочной деятельностью, наносящей вред вашей вычислительной среде, например фиксировать нажатия клавиш клавиатуры и передавать их злоумышленнику.

Менее страшны вставки и надстройки, которые изменяют настройки и внешний вид браузера так, чтобы заставить пользователя посещать определенные сайты, обращаться к определенным поисковым системам, страницы которых полны рекламными объявлениями, и пользоваться определенными программами, на которые ведут новые кнопки, которые были добавлены к стандартным меню и панелям браузера. Этот вид вредоносного программного обеспечения получил название ***AdWare (Adversary Ware),*** т. е. рекламных вирусов. Избавится от паразитов бывает не очень просто, так как они глубоко встраиваются в операционную систему и часто не удаляются обычными средствами браузера. Поэтому добавляйте только те надстройки, которые подписаны цифровой подписью известных производителей. Проверка их антивирусными программами также необходима.

ActiveX объекты. Они представляют собой наибольшую опасность для браузера, потому что их действия не ограничены никакими рамками, они могут читать, создавать и удалять файлы и выполнять любые системные действия. Компания Microsoft снабжает свои объекты ActiveX цифровой подписью, так же делают и другие производители программного обеспечения, поэтому браузер должен принимать только те объекты ActiveX, которые подписаны вызывающим доверие разработчиком.

JavaScript. Разработчики этого языка встроили в него средства безопасности, что значительно уменьшает риски, связанные с использованием скриптов этого типа. Скрипт JavaScript не может получить доступ к файлам компьютера, а также к его сетевым интерфейсам, но имеет доступ к объектам браузера (например, читать куки) и делать HTTP-запросы.

Java-апплеты. В языке java также имеются средства безопасности, в частности поведение апплета контролируется объектом «менеджер безопасности», который обычно не разрешает апплету выполнять такие опасные действия, как вызов произвольных функций, чтение файлов за пределами пользовательского каталога, обращения к веб-сайтам, отличным от того, который загрузил апплет, и т. п.

Браузеры позволяют пользователям управлять процессом обработки элементов создания динамического содержания страницы. Так, пользователь может запретить выполнять объекты ActiveX или Java-апплеты или разрешить их выполнения только от доверенных сайтов, список которых он составляет сам.

Вопросы к главе 20

X. Основным источником заражения веб-браузера являются:

а) данные, вводимые пользователем этого веб-браузера;

б) данные, поступающие от посещаемых зараженных веб-сайтов;

в) скрипты и другие исполняемые коды динамических веб-страниц;

2. Можно ли в адресе URL указать пароль пользователя?

а) да;

б) нет.

3. В каких целях веб-сервер не хранит данные о состоянии сессии пользователя?

а) для более устойчивой работы в условиях ненадежных сетевых соединений;

б) для упрощения организации веб-сервера;

в) для ускорения работы веб-сервера.

4. В какой версии протокола HTTP поддерживается конвейерная передача запросов?

а) HTTP 1.0;

б) HTTP 2.0;

в) HTTP 1.1.

5. Какие команды протокола HTTP из перечисленных ниже считаются наиболее уязвимыми для веб-сервера:

а) GET;

б) POST;

в) PUT;

г) HEAD;

д) DELETE.

6. Что означает код ошибки 505?

а) страница не найдена (Page Not Found);

б) для успешного выполнения операции нужно произвести следующий запрос по другому URL-адресу, указанному в дополнительном заголовке Location;

в) данная версия протокола HTTP не поддерживается.

7. Веб-страница может создаваться динамически:

а) только веб-сервером;

б) только веб-браузером;

в) как веб-сервером, так и веб-браузером.

8. Какого типа надстройкой для браузера Mozilla Firefox является программа Adobe Acrobat NPAPI:

а) расширением (extensions);

б) темой (Theme);

в) вставкой (plug-in).

9. Код какого типа несет наибольшую угрозу безопасности веб-браузера Microsoft Internet Explorer:

а) JavaScript;

б) Java;

в) ActiveX.

10. Могут ли куки, переданные веб-серверами вашему веб-браузеру, заразить ваш компьютер вирусом?

а) да;

б) нет.

11. С какой целью веб-сервер передает куки веб-браузеру клиента?

а) для создания динамический веб-страниц;

б) для сохранения состояния сессии с пользователем;

в) для повышения защищенности сессии.

12. Какое утверждение относительно протокола HTTPS является правильным.

а) протокол HTTPS использует защищенный канал SSL для предотвращения атак на сессию между веб-браузером и веб-сервером;

б) протокол HTTPS не является протоколом в строгом смысле этого термина,

в) протокол HTTPS использует цифровые сертификаты веб-браузера и веб-сервера для образования защищенного канала.

13. По каким причинам сертификат веб-сервера может быть признан веб-браузером недействительным?

а) выдавший его сертификационный центр не является доверенным, т. е. не входит в иерархию ни одного из доверенных корневых центров;

б) срок действия сертификата истек;

в) имя сервера, набранного пользователем, не совпадает ни с одним доменным именем сервера, фигурирующим в сертификате;

г) все приведенные выше ответы неверны.

21 Безопасность электронной почты

Организация почтового сервиса

Сетевая почтовая служба, или ***электронная почта,*** — это распределенное приложение, главной функцией которого является предоставление пользователям сети возможности обмениваться электронными сообщениями.

Как и все сетевые службы, электронная почта построена в архитектуре клиент-сервер. Почтовый клиент всегда располагается на компьютере пользователя, а почтовый сервер, как правило, работает на выделенном компьютере.

Почтовый клиент (называемый также *агентом пользователя)* — это программа, предназначенная для поддержания пользовательского интерфейса (обычно графического), а также для предоставления пользователю широкого набора услуг по подготовке электронных сообщений. В число таких услуг входит создание текста в различных форматах и кодировках, сохранение, уничтожение, переадресация, сортировка писем по разным критериям, просмотр перечня поступивших и отправленных писем, грамматическая и синтаксическая проверка текста сообщений, ведение адресных баз данных, автоответы, образование групп рассылки и прочее и прочее. Кроме того, почтовый клиент поддерживает взаимодействие с серверной частью почтовой службы.

Почтовый сервер выполняет прием сообщений от клиентов, для чего он постоянно находится в активном состоянии. Кроме того, он выполняет буферизацию сообщений, распределение поступивших сообщений по индивидуальным буферам (почтовым ящикам) клиентов, управляет объемами памяти, выделяемой клиентам, выполняет регистрацию .клиентов и регламентирует их права доступа к сообщениям, а также решает много других задач.

Электронные сообщения

Почтовая служба оперирует *электронными сообщениями* — информационными структурами определенного стандартного формата.

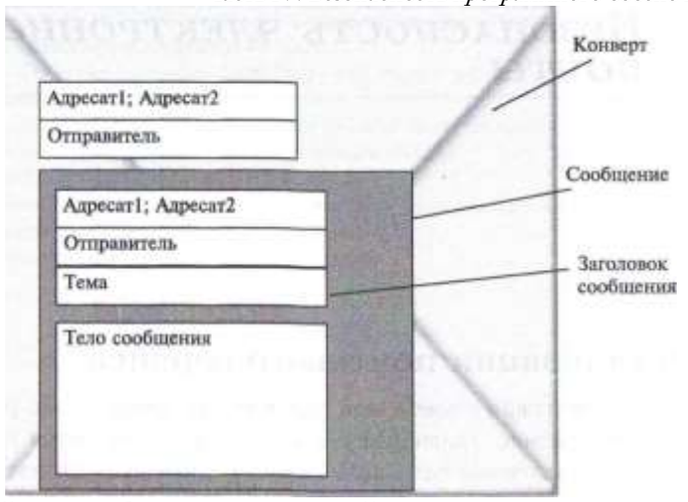


Рис. 21.1. Конверт и сообщение электронной почты Интернет

Упрощенно электронное сообщение может быть представлено в виде двух частей, одна из которых (заголовок) содержит вспомогательную информацию для почтовой службы, другая часть (тело сообщения) — это собственно то «письмо», которое предназначается для прочтения, прослушивания или просмотра адресатом (RFC 822).

Главными элементами заголовка являются адреса отправителя и получателя в виде Polina@domen.com, где Polina — идентификатор пользователя почтовой службы, а domen.com — имя домена, к которому относится этот пользователь. Кроме этого, почтовая служба включает в заголовок дату и тему письма, делает отметки о применении шифрации, о срочности доставки, о необходимости подтверждения факта прочтения этого сообщения адресатом и др. Дополнительная информация заголовка может оповещать почтового клиента получателя об использовании той или иной кодировки.

При транспортировке через Интернет почтовой сообщение помещается в ***конверт (envelope),*** который также имеет несколько служебных полей, например поле отправителя и поля получателей (рис. 21.1). Информация конверта используется только при транспортировке почтового сообщения, а информация заголовка сообщения — почтовым клиентом получателя.

Первоначально тело сообщения представляло собой сплошной текст ASCII, такая же кодировка использовалась и для служебных полей конверта, и заголовка сообщения.

Большая популярность электронной почты привела к ее интернационализации, что заставило принять несколько новых стандартов, разрешающих использовать в теле сообщения не только коды ASCII, но и такие коды, как UTF-8, позволяющие пользователям всех стран использовать свой родной язык при написании электронного письма. Ограничение на использование ASCII осталось только для полей конверта,

и это ограничение приходится преодолевать несколько искусственным способом, преобразуя исходные не-ASCII символы в более длинную последовательность ASCII кодов (например, используя популярный в системах электронной почты алгоритм base64).

Важную роль в расширении возможности электронной почты по передаче мультимедийной информации сыграл стандарт *MIME* (*Multi-purpose Internet Mail Extensions*) **—** « *Многоцелевые расширения почты Интернет».* Этот стандарт описывает структуру сообщения, состоящего из нескольких частей, каждая из которых имеет свой заголовок и тело. Заголовок описывает тип данных, которые содержатся в теле, это могут быть как обычные текстовые данные в формате ASCII, так и данные другого типа, например:

- текст в 8-битном формате (такая возможность стала стандартной совсем недавно, она описана в RFC 6152, принятом в марте 2011 г.);
- текст в не-ASCII коде, преобразованном в коды ASCII (например, с помощью уже упомянутого алгоритма base64);
- гипертекст HTML;
- изображение;
- видеоклип;
- звуковой файл.

Части отделяются друг от друга последовательностью символов, называемой границей *(boundary),* которая не должна встречаться в теле частей сообщения.

В заголовке каждой части сообщения имеется также информация о том, каким образом почтовый клиент должен обрабатывать тело части — отображать ли ее немедленно при открытии сообщения (например, встраивая изображение в текст) или считать это тело приложением *(attachment),* которое пользователь будет обрабатывать сам.

Одна из спецификаций стандарта MIME (каждая спецификация MIME описывает одно или несколько расширений оригинальной спецификации RFC 822), а именно RFC 1847, относится к расширениям безопасности, поэтому ее называют спецификацией *S/MIME (Security MIME).* В S/MIME описаны два новых типа частей MIME:

- цифровая подпись Mutipart/Signed;
- шифрованное тело Multipart/Encrypted.

Эти два типа частей сообщения могут использоваться вместе, для обеспечения аутентичности, целостности и конфиденциальности электронного письма.

Протокол SMTP

В качестве средств передачи сообщения почтовая служба Интернета использует стандартный, разработанный специально для почтовых систем протокол ***SMTP (Simple Mail Transfer Protocol — простой протокол передачи почты).*** Этот протокол был одним из первых протоколов прикладного уровня, стандартизированных IETF, — первая версия этого протокола была описана Джоном Постелом (который был редактором RFC на протяжении многих лет) в RFC 821 в августе 1982 года. Текущая версия SMTP описана в RFC 5321 (октябрь 2008 г.).

Как и большинство других протоколов прикладного уровня, протокол SMTP реализуется несимметричными взаимодействующими частями: SMTP-клиентом и SMTP-сервером. Важно отметить, что этот протокол ***ориентирован на передачу данных по направлению от клиента к серверу,*** следовательно, SMTP-клиент работает на стороне отправителя, а SMTP-сервер — на стороне получателя. SMTP-сервер должен постоянно быть в режиме подключения, ожидая запросов со стороны SMTP-клиента.

Логика протокола SMTP действительно достаточно простая, как это и следует из его названия. После того как, применяя графический интерфейс своего почтового клиента, пользователь щелкает по значку отправки сообщения, SMTP-клиент посылает запрос на установление TCP-соединения на порт 25 SMTP-сервера (это назначенный порт SMTP-сервера). Если сервер готов, то он посылает свои идентифицирующие данные, в частности свое DNS-имя. Затем клиент передает серверу адреса (имена) отправителя и получателя. Если имя получателя соответствует ожидаемому, то после получения адресов сервер дает согласие на установление SMTP-соединения и в рамках этого логического канала происходит передача сообщения. Если после приема тела сообщения сервер отвечает командой ОК, то это означает, что сервер принял на себя ответственность по дальнейшей передаче сообщения получателю. Однако это не означает, что сервер гарантирует успешную доставку, потому что это зависит не только от него, например клиентская машина получателя может быть в течение длительного времени не подсоединена к Интернет. Это только означает, что сервер приложит со своей стороны все усилия, чтобы сообщение было доставлено.

Используя одно TCP-соединение, клиент может передать несколько сообщений, предваряя каждое из них указанием адресов отправителя и получателя.

После завершения передачи сообщения TCP- и SMTP-соединения разрываются. Если в начале сеанса связи SMTP-сервер оказался не готов, то он посылает соответствующее сообщение клиенту и тот снова посылает запрос, пытаясь заново установить соединение. Если сервер не может доставить сообщение, то он передает отчет об ошибке отправителю сообщения и разрывает соединение. После того как передача сообщения благополучно заканчивается, переданное сообщение сохраняется в буфере на сервере.

Нужно отметить, что в протоколе SMTP предусмотрены как положительные, так и отрицательные уведомления о доставке (промежуточной или окончательной) электронного письма. Однако только отрицательные уведомления являются обязательными, поэтому обычно серверы SMTP предпочитают не передавать положительные уведомления в направлении отправителя.

Хотя в любом протоколе предполагается обмен данными между взаимодействующими частями, т. е. данные передаются в обе стороны, различают протоколы, ориентированные на передачу (push protocols), и протоколы, ориентированные на прием данных (pull protocols). В протоколах, ориентированных на передачу, к которым, в частности, относится протокол SMTP, клиент является инициатором передачи данных на сервер, а в протоколах, ориентированных на прием, к которым относятся, например, протоколы НЗ ГР, POP3 и IMAP, клиент является инициатором получения данных от сервера.

Непосредственное взаимодействие клиента и сервера

Теперь, когда мы обсудили основные составляющие почтовой службы, давайте рассмотрим несколько основных схем ее организации. Р1ачнем с простейшего практически не используемого сейчас варианта, когда отправитель непосредственно взаимодействует с получателем. Как показано на рис. 21.2, у каждого пользователя на компьютере установлены почтовые клиент и сервер.

Рис. 21.2. **Схема непосредственного взаимодействия клиента и сервера**

Данила, используя графический интерфейс своего почтового клиента, вызывает функцию создания сообщения, в результате чего на экране появляется стандартная незаполненная форма сообщения, в поля которой Данила вписывает свой адрес, адрес Полины и тему письма, а затем набирает текст письма. При этом он может пользоваться не только встроенным в почтовую программу текстовым редактором, но и привлекать для этой .цели другие программы, например MS Word. Когда письмо готово, Данила вызывает функцию отправки сообщения и встроенный SMTP-клиент посылает запрос на установление связи SMTP-серверу на компьютере Полины. В результате устанавливаются SMTP- и TCP-соединения и сообщение передается через сеть. Почтовый сервер Полины сохраняет письмо в памяти ее компьютера, а почтовый клиент по команде Полины выводит его на экран, при необходимости выполняя преобразование формата. Полина может сохранить, переадресовать или удалить это письмо. Понятно, что в случае, когда Полина решит направить электронное сообщение Даниле, схема работы почтовой службы будет симметричной.

Схема с выделенным почтовым сервером

Рассмотренная только что простейшая схема почтовой связи кажется работоспособной, однако у нее есть серьезный и очевидный дефект. Мы упоминали, что для обмена сообщениями необходимо, чтобы SMTP-сервер постоянно находился в ожидании запроса от SMTP- клиента. Это означает, что для того чтобы письма, направленные Полине, доходили до нее, ее компьютер должен постоянно находиться в режим подключения. Понятно, что такое требование для большинства пользователей неприемлемо.

Естественным решением этой проблемы является размещение SMTP-сервера на специально выделенном для этой цели компьютере-посреднике. Это должен быть достаточно мощный и надежный компьютер, способный круглосуточно передавать почтовые сообщения от многих отправителей ко многим получателям. Обычно почтовые серверы поддерживаются крупными организациями для своих сотрудников или провайдерами для своих клиентов. Для каждого домена имен система DNS создает записи типа MX, в которых хранятся DNS-имена почтовых серверов, обслуживающих пользователей, относящихся к этому домену.

На рис. 21.3 представлена схема с выделенным почтовым сервером. Чтобы не усложнять рисунок, мы показали на нем только те компоненты, которые участвуют в передаче сообщения от Данилы к Полине. Для обратного случая схема должна быть симметрично дополнена.

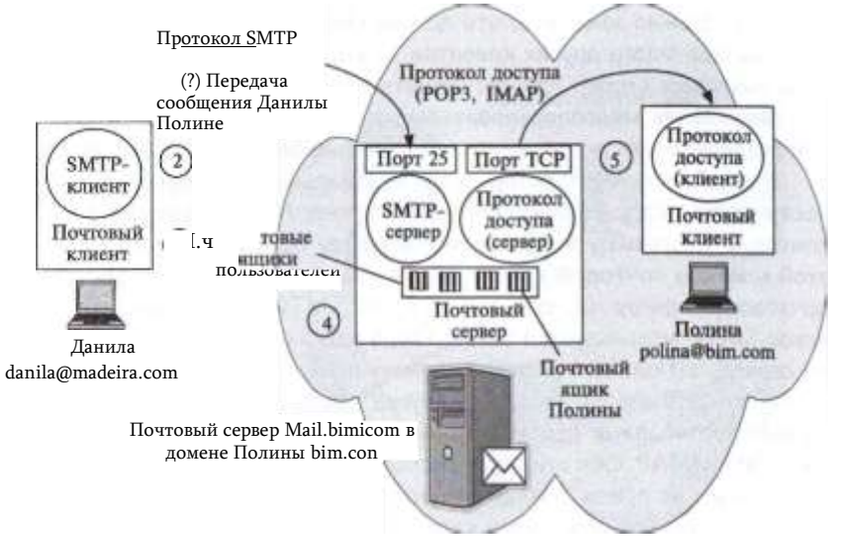


Рис. 21.3. **Схема непосредственного взаимодействия клиента и сервера**

Итак, пусть Данила решает послать письмо Полине, для чего он запускает на своем компьютере установленную на нем программу почтового клиента (например, Microsoft Outlook или Mozilla Thunderbird). Он пишет текст сообщения, указывает необходимую сопроводительную информацию, в частности адрес получателя polina@bim.com, и щелкает мышью на значке отправки сообщения. Поскольку готовое сообщение должно быть направлено совершенно определенному почтовому серверу, клиент обращается к системе DNS, чтобы определить имя почтового сервера, обслуживающему домен Полины bim.com. Получив от DNS в качестве ответа имя mail.bim.com, SMTP-клиент еще раз обращается к DNS, на этот раз, чтобы узнать IP-адрес почтового сервера mail.bim.com.

SMTP-клиент посылает по данному IP-адресу запрос на установление TCP-соединения через порт 25 (SMTP-сервер). С этого момента начинается диалог между клиентом и сервером по протоколу SMTP, с которым мы уже знакомы. Заметим, что здесь, как и у всех протоколов, ориентированных на передачу, направление передачи запроса от клиента на установление SMTP-соединения совпадает с направлением передачи сообщения. Если сервер оказывается готовым, то после установления TCP-соединения сообщение Данилы передается.

Письмо сохраняется в буфере почтового сервера, а затем направляется в индивидуальный буфер, отведенный системой для хранения корреспонденции Полины. Такого рода буферы называют почтовыми

ящиками. Важно заметить, что помимо Полины у почтового сервера имеется еще много других клиентов, и это усложняет его работу. То есть почтовый сервер должен решать самые разнообразные задачи по организации многопользовательского доступа, включая управление разделяемыми ресурсами и обеспечение безопасного доступа.

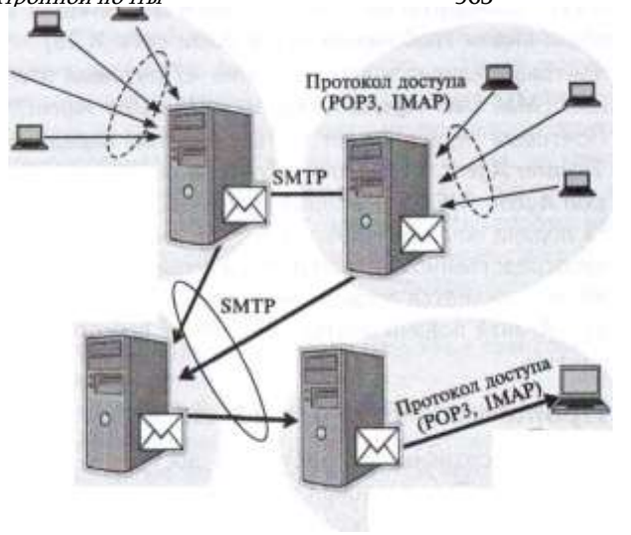
В какой-то момент, который-принципиально не связан с моментом поступления сообщений на почтовый сервер, Полина запускает свою почтовую программу и выполняет команду проверки почты. После этой команды почтовый клиент должен запустить протокол доступа к почтовому серверу. Однако это не будет SMTP. Напомним, что протокол SMTP используется тогда, когда необходимо передать данные на сервер, а Полине, напротив, нужно получить их с сервера. Для этого случая были разработаны другие протоколы, обобщенно называемые протоколами доступа к почтовому серверу, такие, например, как POP3 и IMAP. Оба этих протокола относятся к протоколам, ориентированным на прием данных. Инициатором передачи сообщений от почтового сервера почтовому клиенту по протоколу POP3 или IMAP является клиент. Почтовый сервер ожидает запрос на установление TCP-соединения по протоколу POP3 через порт 110, а по протоколу IMAP — через порт 143, на рисунке эти порты обобщенно изображены как порт TCP. В результате работы любого из них письмо Данилы оказывается в памяти компьютера Полины. Заметим, что на этот раз направление запроса от клиента к серверу не совпадает с направлением передачи данных, показанному стрелкой.

Оба протокола — POP3 и IMAP — поддерживают отправку отправителю квитанций, подтверждающих доставку, а также факт открытия сообщения получателем в случае, когда отправитель запрашивает такую почтовую услугу. Обычно для обеспечения приватности почтовый клиент не отправляет квитанцию автоматически, а спрашивает у получателя разрешение на такое действие.

Схема с двумя почтовыми серверами посре дниками

Прежде чем мы перейдем к сравнению двух протоколов доступа к почте, давайте посмотрим на еще одну схему организации почтовой службы, наиболее приближенную к реальности (рис. 21.4). Здесь передача сообщений между клиентами почты (на рисунке между отправителем Данилой и получателем Полиной) проходит через два промежуточных почтовых сервера, каждый из которых обслуживает домен своего клиента. На каждом из этих серверов установлены также и клиентские части протокола SMTP. При отправке письма почтовый клиент Данилы передает сообщение по протоколу SMTP почтовому серверу домена, к которому относится Данила, — RoyalMail.madeira.com.

Это сообщение буферизуется на данном сервере, а затем по протоколу SMTP передается дальше на почтовый сервер домена Полины — mail.bim.com, откуда описанным уже образом попадает на компьютер



Полины.

Возникает вопрос, зачем нужна такая двухступенчатая передача через два почтовых сервера? Прежде всего для повышения надежности и гибкости процедуры доставки сообщения. Действительно, в схеме с передачей сообщения сразу на сервер получателя почтовый клиент отправителя в случае неисправности почтового сервера должен самостоятельно справляться со сложившейся нештатной ситуацией. Если же посредником в передаче сообщения является другой почтовый сервер, то это позволяет реализовывать разнообразные логические механизмы реакции на отказы на стороне сервера, который к тому же всегда находится в режиме подключения. Например, при невозможности передать письмо почтовому серверу получателя сервер отправляющей стороны может не только рапортовать об этом своему клиенту, но и предпринимать собственные действия — пытаться снова и снова послать письмо, повторяя эти попытки в течение достаточно длительного периода.

Для почтовых клиентов и серверов существуют и другие названия, пришедшие в почту Интернета из почтовых стандартов ITU-T

Рис. 21.4. Схема с выделенными почтовыми серверами в качестве посредников

X.400 (эти стандарты были популярны в до-Интернет времена, когда основным видом глобальных сетей были сети X.25):

- почтовый клиент носит название «***Почтовый агент пользователя***» ***(Mail User Agent, MUA)*** или ***UA (User Agent);***
- Почтовый сервер может быть ***«Агентом передачи почты» (Mail Transfer Agent, MTA)*** или ***«Агентом подачи почты» (Mail Submission Agent, MSA).*** Основной особенностью сервера в роли «Агента подачи почты» является то, что ему передает (подает) почту непосредственно пользователь, а сервер в роли «Агента передачи почты» является посредником, так как принимает сообщение или от «Агента подачи почты» или же от другого «Агента передачи почты».

Протоколы POP3 и IMAP

А теперь сравним два протокола доступа к почте: ***POP3 (Post Office Protocol*** v.3 — протокол почтового отделения версии 3) и ***IMAP (Internet Mail Access Protocol—*** протокол доступа к электронной почте Интернета). Оба протокола решают одну и ту же задачу — обеспечивают пользователей доступом к их корреспонденции, хранящейся на почтовом сервере. В связи с многопользовательским характером работы почтового сервера оба протокола поддерживают аутентификацию пользователей на основе идентификаторов и паролей пользователей. Однако протоколы POP3 и IMAP имеют и принципиальные различия, важнейшее из которых состоит в следующем. Получая доступ к почтовому серверу по протоколу POP3, вы «перекачиваете» адресованные вам сообщения в память своего компьютера, при этом на сервере не остается никакого следа от считанной вами почты. Если же доступ осуществляется по протоколу IMAP, то в память вашего компьютера передаются только копии сообщений, хранящихся на почтовом сервере.

Это различие серьезно влияет на характер работы с электронной почтой. Сейчас очень распространенной является ситуация, когда человек в течение одного и того же периода времени использует несколько различных компьютеров: на постоянном месте работы, дома, в командировке. Теперь давайте представим, что произойдет с корреспонденцией пользователя Полины, если она получает доступ к почте по протоколу POP3. Письма, прочитанные на работе, останутся в памяти ее рабочего компьютера. Придя домой, она уже не сможет прочитать их снова. Опросив почту дома, она получит все сообщения, которые поступили с момента последнего обращения к почтовому серверу, но из памяти сервера они исчезнут, и завтра на работе она, возможно, не обнаружит важные служебные сообщения, которые были загружены на диск ее домашнего ноутбука. Таким образом, получаемая Полиной корреспонденция будет «рассеяна» по всем компьютерам, которыми она пользовалась. Такой подход не позволяет рационально организовать почту: распределять письма по нескольким различным папкам, сортировать их по разным критериям, отслеживать состояние переписки, отмечать письма, на которые

послан ответ, и письма, еще требующие ответа, и т. д. Конечно, если пользователь всегда работает только с одним компьютером, недостатки протокола POP3 не являются столь критичными. Но и в этом случае проявляется еще один «дефект» этого протокола — клиент не может пропустить, не читая, ни одного письма, поступающего от сервера. То есть объемное и возможно совсем ненужное вам сообщение может надолго заблокировать вашу почту.

Протокол IMAP был разработан как ответ на эти проблемы. Предположим, что теперь Полина получает почту по протоколу IMAP. С какого компьютера она бы ни обратилась к почтовому серверу, ей будут переданы только копии запрошенных сообщений. Вся совокупность полученной корреспонденции останется в полной сохранности в памяти почтового сервера (если, конечно, не поступит специальной команды от пользователя об удалении того или иного письма). Такая схема доступа делает возможным для сервера предоставление широкого перечня услуг по рациональному ведению корреспонденции, т. е. именно того, чего лишен пользователь при применении протокола POP3. Важным преимуществом IMAP является также возможность предварительного чтения заголовка письма, после чего пользователь может принять решение о том, есть ли смысл получать с почтового сервера само письмо.

Угрозы и механизмы защиты почты

Угрозы, связанные с электронной почтой, можно разделить на два больших класса:

- угрозы собственно почтовому сервису, например угроза конфиденциальности переписки;
- угрозы программному обеспечению компьютера (ОС, приложения) из-за вредоносного кода, содержащегося в почтовом сообщении, например угроза заражения вирусом, включенным в тело сообщения, программ компьютера.

Угрозы почтовому сервису

Пользователи, обменивающиеся сообщения электронной почты через Интернет, должны принимать во внимание наличие следующих угроз:

- спуфинг имени отправителя, когда злоумышленник выдает себя за другого пользователя;
- спуфинг почтовых серверов, когда MTA или MSA предъявляют при передаче сообщения ложное имя домена;
- модификация сообщения — искажение или отбрасывание сообщения (т. е. нарушение целостности или доступности сервиса);
- утечка информации — чтение сообщения злоумышленником (нарушение конфиденциальности);
- нарушение последовательности сообщений;
- нарушение свойства неотказуемости: отказ отправителя от факта отправки письма, отказ почтового сервера от факта приема письма, отказ получателя от факта получения письма;

- спам — засорение почтовых ящиков пользователей письмами, которые пользователи не просили или же не ожидали получить. Обычно спам состоит из рекламных сообщений;
- фишинг — электронное письмо обычно является первым этапом фишинга (напомним, что целью этой атаки является завладение учетными данными пользователя для последующего использования, например для снятия денег со счета, в электронных платежах и т. п.). Такое электронное письмо может выглядеть очень похожим на «настоящее», т. е. иметь все атрибуты оформления письма некоторого банка или солидной организации, и содержат просьбу обновить свой пароль по приводимой ссылке. Вторую часто фишинга выполняет веб-сайт, на который попадает пользователь, нажав на ссылку;
- нарушение приватности пользователя за счет сбора метаданных почтового сервиса.

Все перечисленные угрозы являются следствием того, что изначально почтовая служба Интернета, основанная на протоколе SMTP, не поддерживала никаких механизмов защиты почтового обмена. Поэтому, например, спуфинг отправителя являлся очень простым делом — почтовый клиент злоумышленника или же его почтовый сервер помещали туда любое имя, требуемое для обмана получателя. Факт такой подмены обнаружить было очень трудно, так как имена пользователей не хранятся в системе DNS и проверить соответствие IP-адреса имени этим путем невозможно, а аутентификация отправителя по протоколу SMTP предусмотрена не была (только получатель аутентифицировался паролем при получении сообщения).

Аналогично обстояло дело с целостностью и конфиденциальностью переписки, так как текст сообщения в пакетах SMTP передавался в открытом виде и его легко было прочитать и модифицировать.

Отсутствие аутентификации отправителя приводила к проблемам неотказуемости — всегда можно отказаться от факта отправки письма, сославшись на спуфинг отправителя, мол, это кто-то другой его написал, а указал меня в качестве отправителя.

Квитанция о прочтении письма тоже не является в таких условиях достоверной, так как ее мог сгенерировать злоумышленник, преследуя какую-то свою цель. Отправителю спама также легко отказаться от авторства рассылки.

К сожалению, применение прошедшего времени в описании такой грустной картины не совсем оправдано — сплошь и рядом электронная почта Интернет используется в своем первозданном виде, несмотря на то, что за долгие годы существования этого сервиса и его огромную популярность (наверно, не будет преувеличением сказать, что почти каждый пользователь Интернета отправляет и получает электронные письма) разработаны различные стандарты безопасности электронной почты. Ситуация с безопасностью почты не очень хороша из-за того, что велика инерция масштабной распределенной системы Интернет-почты —

существует огромное количество почтовых серверов, работающих под управлением старых версий программного обеспечения, не поддерживающего новые стандарты или же под управлением новых версий, в которых новые функции защиты просто не активированы администраторами.

Существует возможность защиты почтового сообщения силами пользователей, и это очень хороший вариант защиты, но он требует от пользователя некоторой дополнительной работы, например получения личного сертификата и установки его в почтовом клиенте. Далеко на все пользователи считают, что их почта требует такой защиты, или просто не знают о ней. В оправдание таких пользователей можно сказать, что они скорее всего относятся к классу «неуловимых Джо», т. е. их никто не ловит, в том смысле, что их перепиской никто не интересуется. Так что их беспечное поведение возможно оправдано. В том же случае, когда вашей перепиской кто-то очень заинтересуется, есть большая вероятность, что никакие средства защиты не помогут, даже будучи правильно примененными. Мы увидим это на примере случая с генералом Петреусом, который, как ему казалось, очень хорошо засекретил свою личную переписку. Однако между этими крайностями есть много оттенков серого, на которые средства защиты и рассчитаны.

Теперь рассмотрим несколько стандартов безопасности почты Интернета, которые могут уменьшить риски, связанные с ее работой.

Аутентификация отправителя

Существует несколько методов аутентификации отправителя:

- ограничение отправителей провайдером услуг;
- аутентификация отправителя провайдером услуг;

- аутентификация отправителя на основе его личного сертификата.

Ограничение отправителей провайдером услуг не является в строгом смысле аутентификацией. Этот способ основан на том, что почтовый сервер провайдера принимает по протоколу SMTP только те письма, которые отправляются клиентами этого провайдера. А принадлежность отправителя к клиентам провайдера проверяется по его IP-адресу, который должен принадлежать к пулу адресов, которым провайдер владеет и которые он дает своим клиентам. Некоторые провайдеры поступают еще более строго — они не разрешают своим клиентам пользоваться чужим почтовыми серверами для отправки писем, блокируя соединения на порт 25 от клиентских компьютеров, если они направлены не к почтовому серверу провайдера. То есть провайдер не только блокирует чужих пользователей, но и не разрешает своим пользоваться почтовыми услугами других провайдеров — тем самым, осуществляется взаимная защита провайдеров от чужих пользователей (а также привязка пользователей к услуге, что преследует чисто коммерческие цели). Этот метод не гарантирует получателю аутентичности отправителя, но защищает провайдера от спама, отправляемого чужими пользователями.

Аутентификация отправителя провайдером услуг. Расширение протокола SMTP, описывающее процедуру аутентификации пользователя при отправке сообщения агентом пользователя (UA) серверу провайдера почтовых услуг (играющего роль сервера MSA) впервые было предложено в 1995 году Джоном Майерсом. Сегодня это предложение стало стандартом Интернета, его текущая версия описана в RFC 4954, ее обычно называют расширением SMTP AUTH.

В соответствии с этим расширением сервер MSA и клиент UA в начале сессии SMTP договариваются о методе аутентификации. В списке возможных методов находятся:

- открытый пароль (обычно через защищенный канала SSL);
- аутентификация на основе слова-вызова;
- протокол DIGEST-MD5;
- протокол CRAM-MD5,

и этот список может расширяться.

Аутентификация пользователя первым сервером почтовой системы (а именно им является сервер MSA) решает многие проблемы — защищает провайдера от спама, позволяет при необходимости решить проблему неотказуемости отправителя.

Однако при дальнейшей передаче информация об аутентичности пользователя теряется, поэтому отправитель должен полагаться на добросовестность провайдера, под чьим административным управлением находится сервер MSA. Даже если этот провайдер достоин доверия, этот факт не исключает атаки «человек посередине», когда кто-то перехватывает сообщение по пути к почтовому серверу получателя и изменяет имя отправителя.

Аутентификация отправителя на основе его личного сертификата. Этот способ аутентификации работает «из конца в конец», так как

сообщение подписывается цифровой подписью отправителя, чей открытый ключ находится в его личном сертификате. Возможность включения цифровой подписи в качестве части сообщения описана в расширении S/MIME, она предусматривает использование различных стандартов цифровой подписи, например ***PKCS-7*** компании RSA или стандарта ***PGP (Pretty Good Privacy).***

Аутентификация на основе цифровой подписи отправителя решает несколько задач:

- получатель может проверить аутентичность отправителя и целостность сообщения;
- отправитель не может отказаться от факта отправки письма;
- подпись квитанции о получении/чтении письма делает невозможным получателю отказаться от факта получения письма. Цифровая подпись в расширении S/MIME занимает две части

сообщения:

- в первой части описывается используемый стандарт цифровой подписи (протокол) и примененная хеш-функция;
- во второй части, которая является приложением, находится сама цифровая подпись, которая охватывает все части сообщения вместе с их заголовками.

В варианте PKCS-7 частью цифровой подписи S/MIME является и цифровой сертификат, выданный одним из сертифицирующих центров, входящих в иерархию PKI. Сертификат удостоверяет принадлежность открытого ключа отправителю, указанному в заголовке почтового сообщения.

Рассмотрим пример электронного сообщения, подписанного по стандарту PKCS-7 почтовым клиентом Microsoft Windows Mail б.О и принятого почтовым клиентом Apple Mail 6.б (сообщение показано в режиме Raw Source программы Apple Mail б.б, который показывает все MIME-элементы сообщения).

```
Return-path:        <natalia@olifer.co.uk>
Envelope-to:        victor@olifer.co.uk
Message-ID: <5ED892093E784C6D9BFD602759D9A7CS@natashaPC>
From:     cnataliaSolifer.co.uk>
To: "victor" <victor@olifer.co.uk>
Subject:        secure email
Date:     Sat, 9 Nov 2013 11:05:18 -0000
MIME-Version:     1.0
Content-Type:     multipart/signed;
   protocol="application/x-pkcs7-signature";
```

```
    micalg=SHAl;
    boundary»" ----- =_NextPart_000_0017_01CEDD3B. 940A3930"
X-Priority:        3
X-MSMail-Priority:          Normal
X-Mailer:          Microsoft Windows Mail 6.0.6002.18197
X-MimeOLE: Produced By Microsoft MimeOLE V6.0.6002.18463 This is a
multi-part message in MIME format.
    =_NextPart_000_0017-01CEDD3B .940A3930
Content-Type:          multipart/alternative;
    boundary»" ----- =-NextPart _00i _0018_01CEDD3B. 940A3930"
    =JJextPart_001_0018_01CEDD3B. 940A3930
Content-Type:          text/plain;
charset=»iso-8859-l»
Content-Transfer-Encoding:              quoted-printable
Hi,=20
It is much better to use a secure email correspondence.=20 Enjoy!

    =_NextPart_000_0017_01CEDD3B. 940A3930
Content-Type: application/x-pkcs7-signature;
name»"smime.p7s"
Content-Transfer-Encoding: base64 Content-
Disposition: attachment;
filename»"smime.p7s"
MIAGCSqGSIb3DQEHAqCAMIACAQExCzAJBgUrDgMCGgUAMIAGCSqGSIb3DQEHAQAAoII
    TJj
CCBDYwggMeoAMCAQICAQEwDQYJKoZIhvcNAQEFBQAwbzELMAkGAlUEBhMCUOUxF
    DASBgNV
BAoTC0FkZFRydXN0IEFCMSYwJAYDVQQLExlBZGRUcnVzdCBFeHRlcm5hbCBUVFAgTmV0
    d2
9yazEiMCAGAlUEAxMZQWRkVHJlc3QgRXhOZXJuYWwgQOEgUm9vdDAeFwOwMDAlM
zAxMDQ4 MzhaFwOyMDAlMzAxMDQ4Mzha
```

Здесь мы видим обе части цифровой подписи:

- первая часть говорит о том, что это сообщение, подписанное цифровой подписью:

```
Content-Type:          multipart/signed;
protocol»"application/x-pkcs7-signature";
micalg=SHAl;
```

- вторая часть, взятая в рамку, является самой цифровой подписью, форматированной алгоритмом base64, который заменил 8- битовые кода подписи символами ASCII.

На клиенте отправителя был установлен личный сертификат, полученный от компании Comodo. Почтовые клиенты автоматически проверяют подлинность сертификата, с помощью которого получена цифровая подпись PKCS-7, поэтому для пользователя этот этап незаметен, в случае положительной проверки он видит только обычное сообщение (помеченное, как правило, особым значком и сообщением «Подписано таким-то»), А вот в случае отрицательной проверки, когда сертификат отправителя по какой-то причине оказался недействительным, пользователю выводится на экран предупреждение и решение о том, принять сообщение или нет, остается за ним.

Второй вариант цифровой подписи в стандарте S/MIME использует технологию PGP.

Система *Pretty Good Privacy (PGP)* заслуживает особого внимания, потому что она была первой системой цифровой подписи и шифрования почтовых сообщений Интернета, использующей технику публичных ключей. Ее разработал в 1991 году Фил Циммерман, эта система постоянно развивалась и существует сегодня в двух версиях — в качестве открытой технологии OpenPGP, стандартизованной IETF (RFC 4880), и в качестве фирменной технологии компании PGP Corporation (сейчас — подразделение Symantec), которая владеет кодами версий PGP, разработанных Циммерманом и его коллегами.

Одним из основных отличий подхода PGP от S/MIME к получению цифровой подписи является то, что принадлежность открытого ключа некоторому отправителю должна быть подтверждена заранее, до получения письма от данного отправителя. Открытые ключи отправителей, с которыми получатель поддерживает защищенную переписку, должны хранится в некотором хранилище (как правило, локальном, но Symantec предлагает своим клиентам хранить их в хорошо защищенном центральном хранилище), доступном почтовому клиенту получателя. При приходе письма от доверенного отправителя клиентская почтовая программа получателя проверяет подлинность цифровой подписи с помощью открытого ключа отправителя, извлекая его из хранилища.

Для поддержки операции проверки принадлежности открытого ключа некоторому пользователю в PGP вводится понятие *«паутины доверия» (Web of Trust).* Эта паутина похожа на публичную структуру PKI, так как использует цифровые сертификаты и подразумевает иерархию подписывающих их сущностей — но этими сущностями являются пользователи, которым вы прямо или косвенно (через иерархию доверительных отношений) доверяете. В принципе, пользователь PGP системы волен сам решать, каким образом проверять принадлежность открытого ключа другому пользователю PGP, эта система достаточно индивидуальна. Программное обеспечение клиента PGP может также импортировать открытый ключ из сертификата X.509, подписанного одним из сертифицирующих центров PKI, после чего ключ также считается подлинным и помещается в хранилище ключей.

Система PGP позволяет использовать различные алгоритмы шифрования, использующие симметричный ключ, вырабатываемый на один сеанс: IDEA (основной алгоритм ранних версий PGP), AES, 3DES и ряд других.

Шифрование содержимого письма

Шифрование содержимого письма может происходить как «из конца в конец», так и на отдельных участках маршрута следования письма, например между агентом пользователя UA и сервером MSA, принимающим письма от пользователей.

Во втором случае шифрование чаще всего осуществляется средствами защищенного канала, создаваемого между двумя непосредственно общающимися сторонами передачи сообщения. Этот канал может быть каналом IPSec или каналом SSL в зависимости от предпочтений администраторов сетей, в которых расположены эти стороны. Однако такой способ шифрования не всегда гарантирует желаемый конечный результат, то есть обеспечение конфиденциальности сообщения на всем пути от отправителя до получателя, так как какой- то другой участок пути может не использовать защищенный канал, а протокола общей координации участников распределенной схемы передачи писем пока не существует.

Шифрование содержимого письма «из конца в конец» предусмотрено спецификацией S/MIME. Она определяет способ шифрования определенной части составного сообщения, причем эта часть шифруется вместе со своим заголовком.

Как и в случае цифровой подписи, для передачи шифрованной части используется две части сообщения — в первой части описывается факт шифрования и его способ, а вторая часть является приложением, в котором находится зашифрованная исходная часть сообщения.

Спецификация S/MIME предусматривает использование различных стандартов шифрования, например на сегодня определено использование стандарта PKCS-7 компании RSA и шифрование стандарта PGP (Pretty Good Privacy).

Шифрование сообщения обеспечивает его конфиденциальность (естественно, с какой-то степенью уверенности, так как при затрате определенных усилий и времени зашифрованное сообщение можно дешифрировать).

Защита метаданных пользователя

Метаданными электронной почты называют некоторые характеристики сообщений, которые, не передавая самого содержимого сообщения, определяют адресатов переписки и некоторые другие обстоятельства этого процесса. Более точно, к метаданным электронной почты относят:

- имя отправителя, его почтовый адрес и его IP-адрес;
- имя получателя, его почтовый адрес и его IP-адрес;
- тип данных и их кодировки;
- уникальный идентификатор сообщения и связанных с ним сообщений;
- дата, время и временная зона отправки и получения сообщения;
- форматы заголовков сообщения;
- тема письма;
- статус сообщения;
- запрос на подтверждения получения и открытия письма.

Как видно из описания, сбор метаданных почтового сервиса может дать детальную картину о деятельности некоторого пользователя, даже если он шифрует свои сообщения.

По сравнению с данными почтовых сообщений метаданные этого сервиса достаточно просто собрать. Во-первых, потому, что метаданные телекоммуникационных сервисов — почты, мобильной связи, веб-сервиса и других

— законодательствами большинства стран или не защищаются совсем, или защищаются в намного меньшей степени, чем данные сообщений сервиса. То есть в то время как раскрытие содержимого переписки в Интернете требует решения суда, сбор метаданных не считается атакой и может проводится беспрепятственно.

Во-вторых, метаданные именно электронной почты легче привязать к определенному пользователю. Метаданные электронной почты хранятся на компьютерах отправителя и получателя (как и сами сообщения), но что опасно для приватности пользователей — также в журналах почтовых серверов, которые передавали эти сообщения. Отличием почтового сервиса является то, что метаданные пользователей этого сервиса гораздо легче найти на серверах провайдеров, чем метаданные пользователей веб-сервиса, потому что пользователи почты «привязаны» к определенным почтовым серверам, например они отправляют почту через сервер либо своего домашнего провайдера, либо корпоративный сервер, либо сервер провайдера гостиницы, вокзала или кафе, где они временно находятся, либо через сервер публичной почты, такой как Gmail. Пользователь получает почту также через вполне определенный сервер, на котором у него имеется учетная запись. Эта ситуация не похожа на веб-сервис, где пользователь может посетить любой сервер Интернета, так что найти следы его посещений за счет проверки серверов практически невозможно, даже если пользователь регистрировался на некоторых из них.

Вынужденная отставка генерала Давида Петреуса из-за раскрытия его любовной связи с автором его биографии Полой Бродвел подтверждает важность почтовых метаданных. Петреуса нельзя считать человеком, неосведомленным в вопросах информационной безопасности, — после многих лет блестящей военной карьеры, на вершине которой он возглавлял штаб вооруженных сил США, а также был командующим объединенной группировкой войск в Афганистане, Пет- реус был назначен директором ЦРУ. Тем не менее, главный шпион Америки понадеялся на то, что анонимный почтовый аккаунт в Gmail будет вполне безопасен для переписки с Полой, если они не будут отправлять с него писем, а только оставлять черновики писем в локальной папке сервера Gmail. Можно, конечно, сказать, что во всем была виновата Пола, которая начала отправлять с этого аккаунта угрожающие письма другу семьи Петреусов Джил Келли. Джил заявила об этих письмах ФБР, и это инициировало расследование. Так как в деле было замешано имя директора ЦРУ, то расследование было проведено тщательно и выйти на след анонимного пользователя почтового аккаунта помогли почтовые метаданные. Несмотря на то что в содержании писем не было никаких «зацепок», позволяющих определить личность автора, агенты ФБР смогли его найти, сопоставив данные логических входов анонима с перемещениями лиц из круга знакомых Петреуса. Выяснилось, что IP-адреса анонима принадлежат нескольким гостиницам, в которых останавливалась Пола точно в те дни, когда аноним входил в свой аккаунт. Этого совпадения оказалось достаточным, чтобы основной подозреваемой стала Пола, ну а дальнейшие доказательства были уже добыты стандартными способами — обысками дома, личного компьютера и допросами.

Ценность метаданных хорошо понимают спецслужбы, недаром одна из программ NSA, о которых рассказал миру Сноуден, называется телефонной и занимается массовым сбором метаданных мобильных пользователей, благо, что

законы, охраняющие приватность в США, запрещают прослушивание телефонных разговоров, но не запрещают собирать метаданные мобильных клиентов.

Вывод из сказанного простой — защиты почтовых метаданных не существует, шифрование не помогает их скрыть. Возможно, в будущем законодательство в области защиты личных данных будет ужесточено и метаданные станут более защищенными в юридическом отношении.

Атаки на компьютер с помощью почты

Спам

Рассылка письма большому числу адресатов без их согласия или даже намерения вступить в переписку получило названием спама (по названию консервов из скетча Монти Пайтон).

Спам является высокодоходным бизнесом, так как считается хорошим рекламным средством и торговые компании платят за рекламу своих товаров и услуг тем лицам и провайдерам, которые рассылают спам, а затраты на рассылку очень низкие. Правда, компаниям- спамерам приходится предпринимать усилия по составлению списков рассылки, адреса жертв стараются найти различными способами, благо, пользователи должны указывать адрес своей почты очень часто — при регистрации в гостинице, при получении доступа к чему-то бесплатному в Интернете, не говоря уже о платных услугах.

Является ли рассылка спама преступлением или нет, определяется законодательством каждой конкретной страны. В начале 2000-х годов во многих странах были приняты акты, определяющие, что является спамом и какие наказания применяются за его рассылку. Однако принятие этих актов только незначительно уменьшило процент спама в общем потоке электронных писем, он по-прежнему очень высок и достигает 80...85 %. Это связано с тем, что во многих таких актах спам определятся достаточно мягко — массовая рассылка не считается спамом, если в письме ясно указана его рекламная цель, отправитель и получатель имеет возможность отписаться от рассылки. Кроме того, доказать на практике тот факт, что пользователь не давал согласие на получение письма, сложно. В России имеется Федеральный закон от 13.03.2006 № 38-ФЗ (ред. от 23.07.2013) «О рекламе», в статье 18, ч. 1 которого говорится:

1. Распространение рекламы по сетям электросвязи, в том числе посредством использования телефонной, факсимильной, подвижной радиотелефонной связи, допускается только при условии предварительного согласия абонента или адресата на получение рекламы. При этом реклама признается распространенной без предварительного согласия абонента или адресата, если рекламораспространитель не докажет, что такое согласие было получено. Рекламораспространитель обязан немедленно прекратить распространение рекламы в адрес лица, обратившегося к нему с таким требованием.*

Однако на практике этот закон применяется редко.

Со спамом борются провайдеры Интернета. В «Терминах и условиях» их договоров с пользователями обычно есть пункт, запрещающий пользователю рассылать спам. В Интернете существуют так называемые «черные списки»

(balcklists) IP-адресов электронной почты и/или доменных имен, с которых рассылался спам и письма с которых рекомендуется блокировать. Наиболее распространенной практикой является ведение черных списков в виде зон системы доменных имен DNS. Такая практика получила название *DNSBL (DNS Black Lists).* Почтовый сервер провайдера может быть сконфигурирован так. что он автоматически опрашивает какой-либо файл зоны DNS, содержащий черный список, и блокирует письмо, если адрес или имя его отправителя имеется в списке.

http://www.consultant. ru/document/cons-doc_LAW_149904/?frame= 1

Одним из наиболее часто используемых черных списков спамеров является список некоммерческой компании Spamhouse. Возможно, вы помните, что веб-сервер этой компании в марте 2013 года был подвергнут мощной DDoS-атаке с суммарной интенсивностью трафика в 75 Гбит/с, — мы рассматривали этот случай в качестве примера атаки на основе отражения трафика от DNS-серверов в главе 9. Атака была местью одной из компании, занимающейся рассылкой спама, за включение ее в черные списки Spamhouse. Spamhouse имеет очень мощную распределенную систему DNS-серверов, которые предоставляют по запросу почтовых серверов или клиентов черные списки. Spamhouse ведет несколько таких списков — для спамеров, для хостов, зараженных вирусами, для хостов, не выполняющих аутентификацию при передаче письма на почтовый сервер (это считается нарушением политики безопасности почтового сервиса). Владельцы адресов, попавших в черный список, могут оспорить решение Spamhouse, это нормальная процедура, так как при современном «незащищенном» состоянии почты Интернета ошибки определения источника спама неизбежны.

Атаки почтовых приложений

Текст почтового сообщения на первый взгляд не может причинить вред вашему компьютеру — ведь это всего-навсего текст. Но мы знаем, что гибкость современной почты Интернет позволяет внедрить в сообщение разнообразную информацию, в том числе и исполняемые коды, а вот они могут нанести значительный вред вашему компьютеру.

Наиболее просто это сделать с помощью помещения вредоносного кода в приложение почтового сообщения. Почтовый клиент при открытии пользователем приложения передает его одной из программ вашего компьютера для обработки. Если приложением является исполняемый файл, например файл с расширением ,exe, то почтовый клиент передаст его на выполнение операционной системе и программа, находящаяся в файле, может содержать вирус или оказаться троянским конем, который установит себя в вашей системе и начнет свою незаметную, но вредительскую работу. Выполняемая программа может действовать и более грубо, просто стерев ваши файлы или файлы операционной системы, если вы зашли в систему как администратор. Расширения выполняемых программ могут быть и другие, возможно, что это Java-программа или скрипт командного процессора.

Понятно, что открывать приложения, которые представляют собой исполняемую программу, очень опасно, поэтому правилом номер один при работе с почтой является запрет такого действия.

Однако исполняемый код может быть также выполнен в виде макроса или скрипта какого-либо документа, например документа MS Word или Excel. Такое приложение вызывает меньше подозрений (опять же — это только текст или таблица), но скрипты документов также могут получить доступ к ресурсам вашего компьютера и причинить ему вред.

Вредоносный код может находится и не в приложении, а в теле самого сообщения, если это сообщение в формате HTML, а многие почтовые клиенты использует этот формат по умолчанию. Тогда в теле сообщения могут находится скрипты JavaScript или же элементы ActiveX. Как мы знаем, скрипты JavaScript не имеют прямого доступа к критическим ресурсам вашего компьютера, а вот элементы ActiveX гораздо более опасны.

Поэтому все почтовые сообщения должны проходить обязательную проверку антивирусной программой на наличие вредоносного кода в приложениях и самом сообщении.

Вопросы к главе 21

1. Каким элементом почтовой системы Интернета обрабатываются данные служебных полей конверта сообщения?

а) почтовым клиентом получателя;

б) почтовым клиентом получателя и почтовыми серверами, осуществляющими транспортировку сообщения;

в) почтовыми серверами, осуществляющими транспортировку сообщения.

2. С какой целью в почтовой службе Интернета используется алгоритм base64?

а) для шифрования тела сообщения;

б) для цифровой подписи тела сообщения;

в) для преобразования 8-битовых кодов в 7-битовые коды ASCII.

3. Иногда вы получаете цифровую фотографию в виде приложения к электронному письму, а иногда она встроена в текст сообщения. От чего зависит режим передачи фотографии?

а) от конфигурации почтовых серверов вдоль пути следования письма;

б) от конфигурации почтового клиента отправителя;

в) от конфигурации почтового клиента получателя.

4. Что из перечисленного является разрешенным типом части тела электронного письма в стандарте MIME?

а) текст в 8-битном формате;

б) текст в не-ASCII коде, преобразованном в код ASCII;

в) текст в коде ASCII;

г) гипертекст HTML;

д) изображение;

е) видеоклип;

ж) звуковой файл.

5. Какие типы части тела электронного сообщения определяет стандарт S/MIME?

а) цифровая подпись;

б) идентификатор пользователя;

в) шифрованное тело;

г) текст в 8-битовом формате.

6. Какие из перечисленных ниже протоколов почтовый клиент может использовать для приема электронного письма:

а) SMTP;

б) POP3;

в) IMAP;

г) SMNP.

7. Является ли следующее утверждение верным: «Почтовый сервер обязан послать отрицательное уведомление почтовому клиенту отправителя >?

а) да; б) нет.

8. Какие из перечисленных ниже протоколов почтовый клиент может использовать для отправки электронного письма:

а) SMTP;

б) POP3;

в) IMAP;

г) SMIMP.

9. Вы работаете со своей электронной почтой, используя несколько компьютеров. Какой протокол вы должны использовать в почтовых клиентах этих компьютерах, чтобы содержимое локальных почтовых ящиков ваших компьютеров было идентичным, независимо от того, на каком компьютере вы прочитали то или иное письмо?

а) IMAP;

б) POP3.

10. На отражение каких угроз направлен механизм аутентификации отправителя электронного письма на основе его цифровой подписи?

а) нарушение неотказуемости: отказ отправителя от факта отправки письма;

б) нарушение целостности письма;

в) спуфинг имени отправителя;

г) утечка информации.

11. Охватывает ли цифровая подпись и шифрование по стандарту S/MIME приложения к электронному письму?

а) да;

б) нет.

12. Каким образом проверяется подлинность публичного ключа отправителя в стандарте PGP?

а) с помощью публичной инфраструктуры PKI при получении подписанного письма;

б) с помощью «паутины доверия» (Web of Trust);

в) с помощью публичной инфраструктуры PKI перед получением писем от данного отправителя.

13. Что из перечисленного относится к метаданным электронной почты:

а) имя отправителя, его почтовый адрес и его 1P-адрес;

б) имя получателя, его почтовый адрес и его 1P-адрес;

в) тип данных и их кодировки;

г) уникальный идентификатор сообщения и связанных с ним сообщений;

д) дата, время и временная зона отправки и получения сообщения;

е) форматы заголовков сообщения;

ж) тема письма;

з) статус сообщения;

и) запрос на подтверждения получения и открытия письма;

к) все из перечисленного.

14. Записи какого-типа составляют список DNSBL:

а) IP-адреса спамеров;

б) почтовые имена спамеров;

в) IP-адреса получателей спама.

22 Системы защиты программного обеспе чения

Файерволы прикладного уровня

Файерволы прикладного уровня понимают протоколы прикладного уровня, такие как HTTP, SMTP, POP3, IMAP, FTP, SSH, SQL и другие. Для того чтобы отслеживать логику взаимодействия клиентов и серверов некоторого приложения, файерволы прикладного уровня является файерволами с запоминанием состояния, т. е. statefull файерволами.

Нужно отличать функции по блокировке приложений, реализуемые файерволами сетевого уровня, от защиты приложений внутренней сети файерволом прикладного уровня. Файервол сетевого уровня понимает структуру заголовков TCP- и UDP-пакетов и за счет этого может запретить или разрешить прохождение пакетов с определенным номером программного порта TCP или UDP, а так как этот номер присвоен серверной части некоторого приложения, то блокируется весь трафик извне к этому приложению.

Файервол прикладного уровня действует не так примитивно. Он контролирует сессию некоторого приложения и разрешает или запрещает некоторые виды взаимодействия между внутренней и внешней частями этого приложения в соответствии с заданными правилами. Например, при контроле веб-сервиса файервол может разрешить использование только определенных команд протокола HTTP, а остальные запретить. В список запрещенных команд могут попасть опасные для веб-сервера команды PUT и DELETE. Аналогично, при контроле почтового сервиса, файервол прикладного уровня может не пропускать вовне письма, не подписанные цифровой подписью отправителя, если в этом состоит политика безопасности предприятия.

Для некоторых приложений файервол прикладного уровня может работать в режиме прокси-сервера, Мы отдельно рассмотрим особенности прокси-серверов в следующем разделе.

Файервол прикладного уровня корпоративного масштаба чаще всёго является интегрированным продуктом с модульной структурой.

Модульность позволяет ему менять набор поддерживаемых функций фильтрации в зависимости от потребностей конкретной корпоративной сети. Обычно файервол прикладного уровня поддерживает фильтрацию сессий некоторого базового набора наиболее популярных приложений, таких как веб, почта, базы данных, удаленный доступ. Для фильтрации сессий более редких приложений необходимо установить на файерволе дополнительный программный модуль, возможно, от стороннего производителя. Модульная структура файервола подразумевает наличие программного интерфейса API, который сторонний разработчик программных модулей использует для вызова стандартных функций ядра файервола из своего модуля.

Корпоративные файерволы прикладного уровня имеют тенденцию поддерживать за счет дополнительных модулей самые различные функции защиты программного обеспечения:

- антивирусный контроль (на лету) загружаемых пользователем файлов и получаемых писем;
- контроль контента, заключающийся, например, в ограничении доступа пользователей к внешним веб-сайтам, страницы которых содержат заданные ключевые слова; такой же контроль может применяться к электронным письмам, отправляемым вовне;
- транзитную аутентификацию пользователей, обращающихся к некоторому приложению на внутреннем сервере. Это функция полезна для тех приложений, которые либо не выполняют аутентификацию пользователей совсем, либо делают это незащищенным способом, как, например, сервер FTP, который принимает пароли пользователей в открытом виде. Файервол перехватывает обращение пользователя к FTP серверу (команду USER) и организует сессию логического входа пользователя, например с сервером аутентификации Kerberos, если именно такой способ аутентификации используется в корпоративной сети;
- централизованное шифрование электронных писем пользователей, что избавляет пользователей от необходимости конфигурировать такую функцию на своих клиентских компьютерах (для этого файервол должен хранить цифровые сертификаты пользователей);
- функции шлюза VPN с удаленными подразделениями предприятия и удаленными пользователями;
- трансляцию внутренних IP адресов пользователей на основе стандарта NAT.

И стандартное выражение «этот список можно продолжить» очень подходит в данном случае, так как стремление администратора сети сделать корпоративный файервол универсальным стражем порядка не знает границ.

Корпоративный файервол, выполняющий роль универсального контрольно-пропускного пункта сети, должен, естественно, быть высокопроизводительным и отказоустойчивым. Высокая производительность файервола достигается различными способами:

- за счет высокой производительности аппаратной платформы (процессор, память, диски);
- за счет поддержки скоростных сетевых интерфейсов (10GE сегодня является стандартной линией доступа для многих предприятий);

- за счет распределенной архитектуры, когда несколько файерволов защищают несколько линий доступа корпоративной сети. *Отказоустойчивость* достигается за счет применения избыточных модулей файервола, работающих в режиме горячего резервирования. Возможен также вариант, когда два файервола обслуживают две различные линии доступа корпоративной сети и работают в режиме горячего резервирования друг друга. Это означает, что они работают в режиме синхронизации, реплицируя состояния сессий пользователей в режиме реального времени, так что каждый файервол в каждый момент времени имеет состояния своих сессий, а также состояния сессий, проходящих через файервол-напарник. При отказе одного из файерволов или же его линии доступа трафик пострадавших пользователей перенаправляется средствами маршрутизации на другую линию доступа. Так как оставшийся работоспособным файервол располагает состояниями сессий отказавшего файервола, то разрыва сессий пострадавших пользователей не происходит и они на самом деле не являются пострадавшими.

Прокси-серверы

В этом разделе мы рассмотрим функциональное назначение, принципы работы и особенности реализации прокси-серверов, которые наряду с пакетными фильтрами являются важнейшими компонентами файерволов.

Функции прокси-сервера

Прокси-сервер (proxy server) — это тип особый тип приложения, которое выполняет функции посредника между клиентскими и серверными частями распределенных сетевых приложений, причем предполагается, что клиенты принадлежат внутренней (защищаемой) сети, а серверы — внешней (потенциально опасной) сети.

Роль транзитного узла позволяет прокси-серверу логически разорвать прямое соединение между клиентом и сервером с целью контролировать процесс обмена сообщениями между ними.

Рис. 22.1. Расположение прокси-сервера на сетевом экране

Рис. 22.2. **Расположение прокси-сервера на узле внутренней сети**

Подобно сетевому экрану, прокси-сервер может эффективно выполнять свои функции только при условии, что контролируемый им трафик не пойдет обходным путем.

Прокси-сервер может быть установлен не только на платформе, где работают все остальные модули файервола (рис. 22.1), но и на любом другом узле внутренней сети или сети демилитаризованной зоны (рис. 22.2). В последнем случае программное обеспечение клиента должно быть сконфигурировано таким образом, чтобы у него не было возможности установить прямое соединение с ресурсным сервером, минуя прокси-сервер.

Когда клиенту необходимо получить ресурс от какого-либо сервера (файл, веб-страницу, почтовое сообщение), он посылает свой запрос прокси-серверу. Прокси-сервер анализирует этот запрос и на основании заданных ему администратором правил решает, каким образом он должен быть обработан (отброшен, передан без изменения ресурсному серверу, модифицирован тем или иным способом перед передачей, немедленно обработан силами самого прокси-сервера).

В качестве правил, которыми руководствуется прокси-сервер, могут выступать условия пакетной фильтрации. Правила могут быть достаточно сложными, например в рабочие часы блокируется доступ к тем или иным узлам и/или приложениям, а доступ к другим узлам разрешается только определенным пользователям, причем для FTP- серверов пользователям разрешается делать лишь загрузку, а выгрузка запрещается. Прокси-серверы могут также фильтровать почтовые сообщения по типу пересылаемого файла (например, запретить получение приложений формата MP3) и по их контенту. К разным пользователям могут применяться разные правила фильтрации, поэтому часто на прокси-серверы возлагается задача аутентификации пол ьзователей.

Если после всесторонней оценки запроса от приложения прокси- сервер констатирует, что запрос удовлетворяет условиям прохождения дальше во внешнюю сеть, то он выполняет *ю поручению приложения, но от своего имени* процедуру соединения ссервером, затребованным данным приложением.

В некоторых случаях прокси-сервер может изменять запрос клиента. Например, если в него встроена функция трансляции сетевых адресов (см. раздел «NAT-файерволы» в главе 10), он может подменять в пакете запроса IP-адреса и/или номера TCP- и UDP-портов отправителя. Таким способом прокси-сервер лишает злоумышленника возможности сканировать внутреннюю сеть для получения информации об адресах узлов и структуре сети. Единственный адрес в таком случае, который может узнать злоумышленник, — это адрес компьютера, на котором выполняется программа прокси-сервера. Поэтому многие атаки, построенные на знании злоумышленником адресов узлов внутренней сети, становятся нереализуемыми.

Прокси-сервер, выступая посредником между клиентом и сервером, взаимодействующими по определенному протоколу, не может не учитывать специфику этого протокола. Так, для каждого из протоколов HTTP, HTTPS, SMTP/POP, FTP, telnet существует особый прокси-сервер, ориентированный на использование соответствующими приложениями: веб-браузером, электронной почтой, FTP-клиентом, клиентом telnet. Каждый из этих посредников принимает и обрабатывает пакеты только того типа приложений, для обслуживания которого он был создан. Обычно несколько разных прокси-серверов объединяют в один программный продукт,

Посмотрим, как учитывает специфику протокола прокси-сервер, ориентированный на веб-службу. Этот тип прокси-сервера может, например, выполнить собственными силами запрос веб-клиента, не отсылая его к соответствующему веб-серверу. Работая транзитным узлом при передаче сообщений между браузерами и веб-серверами Интернета, прокси-сервер не только передает клиентам запрашиваемые веб-страницы, но и сохраняет их в своей кэш-памяти на диске. В соответствии с алгоритмом кэширования на диске прокси-сервера оседают наиболее часто используемые веб-страницы. При получении

запросов к веб-серверам прокси-сервер прежде всего проверяет, есть ли запрошенная страница в его кэше. Если есть, то она немедленно передается клиенту, а если нет, то прокси-сервер обычным образом делает запрос от имени своего доверителя. Прокси-сервер веб-службы может осуществлять административный контроль проходящего через него контента, в частности ограничивать доступ клиента к сайтам, имеющим IP-адреса или DNS-имена из «черных списков». Более того, он может фильтровать сообщения на основе ключевых слов.

Прокси-серверы прикладного уровня и уровня соединений

Прокси-серверы могут выполнять свою посредническую миссию на разных уровнях.

Рассмотрим пример, иллюстрирующий идею посредничества разного уровня. Для покупки акций инвестор (в нашем случае аналог клиентской части приложения) может прибегнуть к посредническим услугам брокера или трейдера. Брокер, точно следуя указаниям инвестора, покупает для него определенное количество акций определенного типа по определенной цене. Трейдер — это посредник более высокого уровня, которому инвестор поручает самостоятельно принимать решения о необходимых покупках, учитывая различные факторы, например состояние рынка.

Различают прокси-серверы прикладного уровня и уровня соединений.

Прокси-сервер прикладного уровня, как это следует из его названия, умеет «вклиниваться» в процедуру взаимодействием клиента и сервера по одному из прикладных протоколов, например тому же HTTP, HTTPS, SMTP/POP, FTP или telnet. Чтобы выступать в роли посредника на прикладном уровне, прокси-сервер должен «понимать» смысл команд, «знать» форматы и последовательность сообщений, которыми обмениваются клиент и сервер соответствующей службы. Это дает возможность прокси-серверу проводить анализ содержимого сообщений, делать заключения о подозрительном характере того или иного сеанса.

Прокси-сервер уровня соединений выполняет свою посредническую миссию на транспортном уровне, контролируя TCP-соединение. Очевидно, что работая на более низком уровне, прокси-сервер обладает гораздо меньшим «интеллектом» и имеет меньше возможностей для выявления и предупреждения атак. Однако он обладает одним очень важным преимуществом перед прокси-сервером прикладного уровня — универсальностью, т. е. он может быть использован любыми приложениями, работающими по протоколу TCP (а в некоторых случаях и UDP).

Примером прокси-сервера данного типа является разработанный достаточно давно, но все еще широко применяемый сервер *SOCKS* (от SOCKetS).

В версии протокола SOCKS V444 клиент обменивается с прокси- сервером SOCKS двумя сообщениями, запросом клиента SOCKS-сер- веру и ответом SOCKS-сервера клиенту.

Запрос клиента SOCKS-серверу:

- поле 1 — номер версии SOCKS, 1 байт (для этой версии — 4);
- поле 2 — код команды, 1 байт (для установки соединения TCP/IP код равен 1);

- поле 3 — номер порта, 2 байта (TCP-порт запрашиваемого пользователем ресурсного сервера, например, для 21 для FTP);
- поле 4 — 1P-адрес, 4 байта (IP-адрес ресурсного сервера);
- поле 5 — идентификатор пользователя (строка переменной длины, завершаемая байтом null).

SOCKS-сервер анализирует все полученные данные и на основании сконфигурированных для него правил определяет предоставить или нет данному пользователю доступ к данному серверу. Результат SOCKS-сервер сообщает клиенту в виде ответа.

Ответ SOCKS-сервера клиенту:

- поле 1 — байт null;
- поле 2 — код ответа, 1 байт (применяются коды для следующих вариантов ответа: запрос разрешен, запрос отклонен или ошибочен, запрос не удался из-за проблем с идентификацией пользователя);
- несколько байтов, игнорируемых клиентом.

Если прокси-сервер сообщил в ответе, что запрос разрешен, то SOCKS-сервер начинает работать промежуточным звеном между клиентом и сервером (например, FTP), контролируя поток квитанций, которыми они обмениваются.

«Проксификация» прилоснсений

Заметим, что не каждое приложение, построенное в архитектуре клиент-сервер, непременно *должно* работать через прокси-сервер, а также не каждое из них *имеет возможность* работать через прокси- сервер.

Список приложений (точнее их клиентских частей), которые должны передавать свои запросы во внешнюю сеть исключительно через прокси-сервер, определяется администратором. А чтобы эти приложения имели возможности для такого режима выполнения, их программы должны быть соответствующим образом написаны.

Точнее, приложения должны быть оснащены средствами, которые распознавали бы запросы к внешним серверам и перед отправкой преобразовывали эти запросы так, чтобы все они попадали на соответствующий прокси-сервер, а не передавались в соответствии со стандартным протоколом прямо на сервер-адресат. Эти средства должны также поддерживать протокол обмена сообщениями приложения- клиента с прокси-сервером. В последние годы в большинстве приложений, ориентированных на работу через Интернет, предусмотрена *встроенная поддержка прокси-сервера.* Такой поддержкой, например, оснащены все веб-браузеры и все клиенты электронной почты, которыми мы сейчас пользуемся.

«Проксификация» приложения, изначально не рассчитанного на работу через прокси-сервер, требует изменения исходного кода с последующей перекомпиляцией — очевидно, что такая работа не представляет сложностей для разработчиков данного приложения, но администратор сети не всегда может ее выполнить, например из-за отсутствия исходного кода или же необходимой квалификации программиста. Задача администратора заключается в приобретении готовых приложений, совместимых с используемым в сети прокси-сервером. Однако даже

44 Более поздняя версия протокола SOCKS V5 расширяет возможности версии SOCKS V4, добавляя поддержку UDP, доменных имен, адресов IPv6 и процедур аутентификации пользователей. Для поддержки UDP SOCKS V5 устанавливает TCP-соединение и передает по нему данные UDP.

приобретение готового «проксифицированного» клиента не делает его готовым к работе — необходимо еще конфигурирование, в Частности нужно сообщить клиенту адрес узла сети, на котором установлен соответствующий прокси-сервер.

Как можно было бы предположить, процедура «проксификации» значительно упрощается для прокси-сервера уровня соединений, в частности SOCKS-сервера. Для «проксификации» приложения в этом случае достаточно внести простейшие исправления в исходный текст, а затем выполнить его перекомпиляцию и связывание с библиотекой процедур SOCKS. Исправления сводятся к замене всех стандартных вызовов сетевых функций версиями этих функций из библиотеки SOCKS, в частности стандартный вызов listen() заменяется вызовом rlistenQ, вызов bind() — вызовом rbindQ, вызов accept() — вызовом raccept().

Имеется еще один подход к «проксификации» — встраивание поддержки прокси-сервера в операционную систему. В этом случае приложения могут оставаться в полном «неведении» о существовании в сети прокси-сервера, за них все необходимые действия выполнит ОС.

Помимо основных функций, многие прокси-серверы способны обнаруживать вирусы еще до того, как они попали во внутреннюю сеть. К другим полезным (для администрации и службы безопасности) вспомогательным функциям прокси-сервера относится сбор статистических данных о доступе пользователей в Интернет: когда и какие сайты посещал тот или иной пользователь, сколько времени продолжалось каждое посещение.

Программные файерволы хоста

Программные файерволы хоста являются частью его программного обеспечения, они являются дополнительным рубежом защиты хоста, образуя наряду с файерволом сети двухступенчатый контроль трафика.

Программный файервол работает в режиме ядра ОС, контролируя сетевые интерфейсы хоста и перехватывая пакеты до передачи их протоколам стека TCP/IP.

С точки зрения функциональности программные файерволы хоста являются, как правило, файерволами сетевого уровня без учета состояния сессии (stateless). Функции файервола с анализом состояния сессии требуют значительных вычислительных ресурсов компьютера, поэтому запуск такого файервола на пользовательском компьютере или на сервере мог бы привести к большому замедлению выполнения основных его функций. По этой причине файерволы с анализом состояния работают на выделенных серверах или специализированных программно-аппаратных платформах.

Как и файерволы сетевого уровня на основе маршрутизаторов, программные файерволы хоста позволяют применять правила, учитывающие номера TCP/UDP-портов. Это означает, что пользователь хоста может разрешать или запрещать доступ по сети к определенным приложениям хоста, пользующимися закрепленными за ними портами.

Рассмотрим, как можно блокировать доступ по сети к приложениям с помощью программного файервола iptables, имеющегося практически во всех версиях Unix/Linux. Этот файервол запускается как Unix-демон и работает на основе правил, записанных в текстовом виде в файле /etc/sysconfig/iptables. Редактировать этот

файл вручную не рекомендуется, потому что синтаксис его правил достаточно сложен. Поэтому произвольное ручное редактирование, которое система не контролирует, может привести к появлению ошибок в конфигурации файервола, а это весьма опасно — можно заблокировать нужные сервисы или же не запретить нежелательные. Рекомендуется создавать правила файервола iptables с помощью команды iptables, которая проверяет синтаксис нового правила перед его записью в файл правил. С помощью команды iptables можно также распечатать на экране список действующих правил, пример их показан на рис. 22.3.

```
(root^ganyiuede sysconfxgl# iptables "l" Chain INPUT (policy ACCEPT)
target     prot opt  source               destination
ACCEPT     udp -•    anywhere             anywhere          udp dpt:domain
ACCEPT     tcp --    anywhere             anywhere          tcp dpt:domain
ACCEPT     udp -*    anywhere             anywhere          udp dpt:bootps
ACCEPT     tcp --    anywhere             anywhere          tcp dpt:bootps
ACCEPT     all -     anywhere             anywhere          State RELATED,ESTABLISHED
ACCEPT     temp --   anywhere             anywhere
ACCEPT     all --    anywhere             anywhere
ACCEPT     tcp ••    anywhere             anywhere          state fi£W tcp dpt:ssh
ACCEPT     tcp -■    anywhere             anywhere          state NEW tcp dpts :VRC-server :5%3
ACCEPT     udp -*    anywhere             anywhere          state NEW udp dpts:vnc-server:5903
ACCEPT     tcp --    anywhere             anywhere          state NEW tcp dpt:oa-systeff,
ACCEPT     tcp --    anywhere             anywhere          state NEW tcp dpt:webcache
ACCEPT     udp       anywhere             anywhere          state NEW udp dptiwebcache
ACCEPT     tcp -     anywhere             anywhere          state NEW tcp dpt:pcsync-https
ACCEPT     tcp -     anywhere             anywhere          state NEW tcp dptiasasrvr
ACCEPT     tcp --    anywhere             anywhere          state NEW tcp dpt:ddi-tcp-l
REJECT     all •-    anywhere             anywhere          reject-with ic(»p -host "prohibited
```

Рис. 22.3. **Правила файервола iptables**

Правила состоят из трех секций:

- INPUT, правила этой секции фильтруют входящий трафик;
- OUTPUT, правила этой секции фильтруют исходящий трафик;
- FORWARD, правила этой секции фильтруют транзитный трафик в том случае, когда хост работает как IP-маршрутизатор, имея два сетевых интерфейса.

На рис. 22.3 показаны только правила секции INPUT. Синтаксис правил похож на синта'ксис правил маршрутизаторов Cisco, который мы рассмотрели в главе 10.

Рассмотрим для примера правило

ACCEPT tcp -- anywhere anywhere state NEW tcp dpts: vnc-server:5903

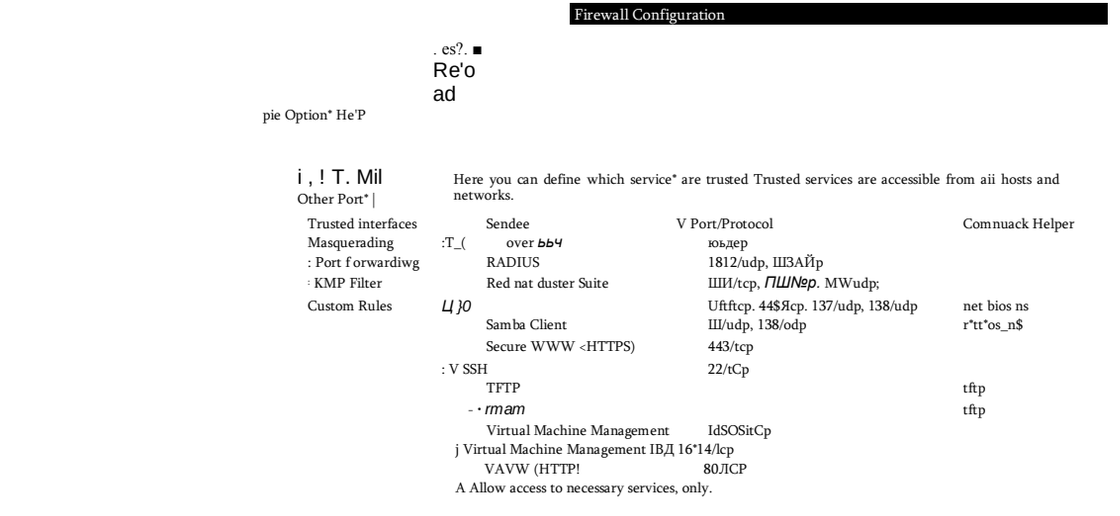


Рис. 22.4. **Графический интерфейс Unix-файервола**

Оно говорит о том, что пакеты, удовлетворяющие условию правила, должны быть приняты (ACCEPT). Этому правилу удовлетворяют пакеты протокола TCP (tcp) с любым адресом источника (anywhere) и любым адресом назначения (anywhere), относящиеся к новой сессии TCP (state NEW, т. е. к пакетам с признаком SYN) и

имеющие в поле порта назначения значения в диапазоне от 5900 (это стандартный порт сервиса vnc-server, поэтому порт задан с помощью своего имени) до 5903.

Список правил секции INPUT состоит еще из нескольких правил, разрешающих прием определенного вида пакетов, а завершает его правило REJECT all anywhere anywhere, которое запрещает все, что не разрешено явно.

Для того чтобы добавить новое правило к уже существующим, нужно выполнить команду iptables с соответствующими параметрами правила.

Для пользователей, которые не любят работать с командной строкой Unix, существует графический интерфейс к файерволу iptables. На рис. 22.4 показан вид окна графической оболочки Firewall ОС CentOS 6.4, отображающего те же правила, которые мы только что видели в режиме командной строки. Для удобства пользователя все стандартные сервисы собраны под пунктом меню Trusted Services, так что пользователю не нужно знать, что, например, сервис SSH использует порт 22, для того, чтобы разрешить его. В то же время для нестандартных сервисов существует возможность, названная в меню Other ports, которая позволяет задавать правила, оперирующие номерами портов TCP или UDP.

Операционные системы Microsoft Windows и Apple OS X включают программный файерволы в комплект поставки ОС. Эти файерволы, как и файервол iptables ОС семейства Unix, представляют собой файерволы без анализа состояния сессий, и также позволяют разрешать или запретить сетевой доступ к отдельным приложениям. В отличие от iptables эти файерволы не поддерживают интерфейс командной строки, ориентируясь на непрофессиональных конечных пользователей, предпочитающих графический интерфейс. По умолчанию, оба эти файервола запрещают любой сетевой доступ извне. Пользователь может добавить к этому правилу по умолчанию исключения — приложения, к которым доступ разрешен. На рис. 22.5 показана панель диалога, с помощью которой пользователь добавляет приложения, используя их привычные имена, а не номера портов.

Файервол Microsoft Windows Vista позволяет также задавать правила фильтрации с использованием имен приложений, но также разрешает пользователю использовать и номера портов TCP/UDP (рис. 22.6).

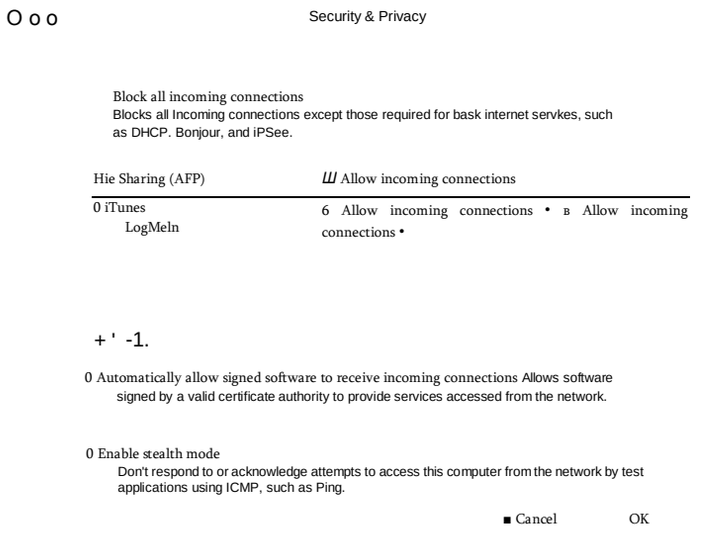


Рис. 22.5. **Добавление приложений к списку разрешенных в файерволе Apple OS X**

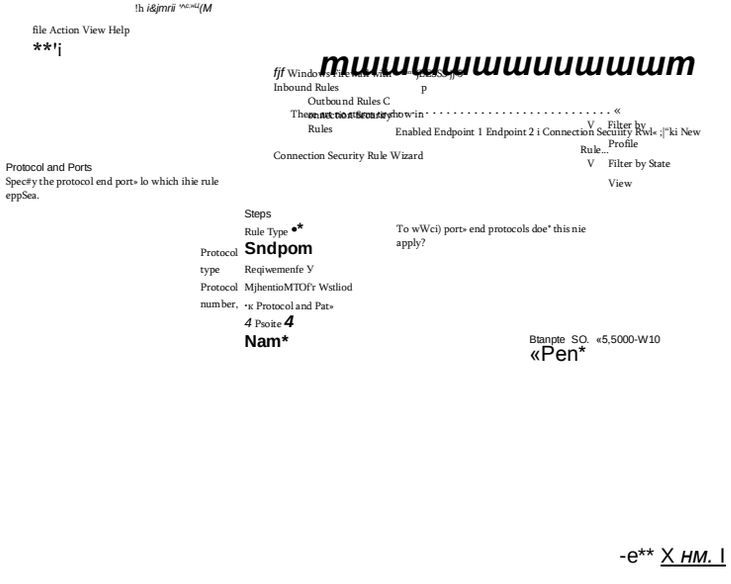


Рис. 22.6. **Задание номера порта в правиле файервола Windows Vista**

Файервол Apple OS X не предоставляет пользователю такой возможности, что отражает общую политику Apple в предоставлении пользователям минимального набора прав для достижения высокой безопасности ОС.

Вопросы к главе 22

1. Какие утверждения являются справедливыми:

а) файервол сетевого уровня не может блокировать трафик определенного приложения:

б) файервол прикладного уровня может блокировать трафик некоторого приложения как целиком, так и только некоторые опасные режимы работы приложения, в то время как файервол сетевого уровня блокирует трафик приложения только целиком:

в) файервол прикладного уровня не может фильтровать трафик на основе информации сетевого уровня.

2. Файервол прикладного уровня может:

а) запоминать состояние сессии приложения:

б) выполнять контроль контента по ключевым словам:

в) шифровать содержимое документов и электронных писем «на лету»;

г) выполнять антивирусный контроль;

д) аутентифицировать пользователей;

е) поддерживать VPN-соединения;

ж) работать в режиме прокси-сервера;

з) остановить атаку изнутри защищаемой сети;

и) выполнять NAT-трансляцию адресов.

3. Отказоустойчивость системы защиты корпоративной сети с помощью файервола может достигаться за счет:

а) горячего резервирования блоков и модулей программно-аппаратной платформы файервола;

б) применения режима прокси-сервера;

в) применения двух файерволов в режим синхронизации.

4. Справедливо ли следующее утверждение: «Прокси-сервер всегда работает в режиме запоминания состояния сессии»?

а) да;

б) нет.

5. Сессии какого протокола контролирует прокси-сервер уровня соединений?

а) протокола HTTP;

б) протокола IP;

в) протокола TCP.

6. Какое из утверждения является ошибочным:

а) любое приложение может работать с прокси-сервером без изменений кода;

б) приложение должно быть написано специальным образом для работы с прокси-сервером;

в) прокси-сервер может модифицировать запросы клиента.

7. Программные файерволы хоста обычно работают в режиме:

а) stateful inspection;

б) stateless filtering.

8. Справедливо ли утверждение: «Программный файервол хоста и файервол прикладного уровня — это два альтернативных названия одного и того же типа файервола»?

а) да;
б) нет.

9. Программный файервол iptables позволяет фильтровать трафик следующего типа:

а) входящий;

б) исходящий;

в) транзитный.

10. Какие свойства правильно характеризуют программный файервол MAC OS X:

а) по умолчанию доступ ко всем приложениям извне разрешен;

б) по умолчанию доступ ко всем приложениям извне запрещен;

в) пользователь может исключить определенные приложения из списка, запретив тем самым к ним доступ извне;

г) пользователь может добавить определенные приложения к списку, разрешив тем самым к ним доступ извне.

Приложение 1. Обзор нормативно-правовых актов РФ в области информационной безопасности

Общегосударственная система мер по обеспечению информационной безопасности России базируется на *Доктрине информационной безопасности Российской Федерации,* утвержденной Президентом Российской Федерации 9 сентября 2000 года.

В соответствии с Доктриной под *информационной безопасностью Российской Федерации* понимается состояние защищенности ее национальных интересов в информационной сфере, определяющихся совокупностью интересов личности, общества и государства. Защита информации включает:

- защиту информации и права на нее, включая право на тайну, право на интеллектуальную собственность, право на доступ к информации;
- защиту информационных систем и прав на них;
- защиту от «вредоносной» информации.

Далее рассматриваются наиболее важные федеральные законы и постановления правительства, в той части, в которой они регулируют правотношения в области информационной безопасности (исключая вопросы защиты интеллектуальной собственности).

Федеральный закон «Об информации, информационных технологиях и о защите информации»

Федеральный закон 149-ФЗ от 8 июля 2006 *«Об информации, информационных технологиях и о защите информации»* является основополагающим. В нем приводятся базовые определения и принципы правового регулирования отношений в сфере информации, информационных технологий и защиты информации, а регулируются права на поиск, получение, передачу, производство и распространение информации, при применении информационных технологий.

Учитывая важность данного закона, приведем перечень названий и некоторые положения его статей для того, чтобы дать представление о содержании закона в целом:

Статья 1. Сфера действия настоящего Федерального закона.

Статья 2. Основные понятия, используемые в настоящем Федеральном законе. Здесь приведены определения терминов «информация», «информационные технологии», «информационная система», «информационно-телекоммуникационная сеть», «электронное сообщение», «электронный документ», сайт в сети «Интернет», «провайдер хостинга», «владелец информации» и другие.

Статья 3. Принципы правового регулирования отношений в сфере информации, информационных технологий и защиты информации. В числе принципов правового регулирования отношений в области защиты информации указаны:

- свобода поиска, получения передачи, производства и распространения информации любым законным способом, ограничиваемым только федеральными законами;
- свободный доступ к информации о деятельности государственных оргэнов;
- достоверность информации и своевременность ее предоставления;
- неприкосновенность частной жизни, недопустимость сбора, хранения, использования и распространения информации о частной жизни лица без его согласия.

Статья 4. Законодательство Российской Федерации об информации, информационных технологиях и о защите информации. Статья 5. Информация как объект правовых отношений. Информация в зависимости от порядка ее предоставления или распространения подразделяется на:

- свободно распространяемую;
- предоставляемую по соглашению лиц, участвующих в соответствующих отношениях;
- предоставляемую или распространяемую по установлению федеральных законов;
- ограниченно-распространяемую или запрещенную к распространению в РФ.

Статья б. Обладатель информации.

Статья 7. Общедоступная информация.

Дополнение к этой статье (от 07.06.2013) устанавливает, что «информация, размещаемая ее обладателями в сети «Интернет» в формате, допускающем автоматизированную обработку без предварительных изменений человеком в целях повторного ее использования, является общедоступной информацией, размещаемой в форме открытых данных». Размещение информации в форме открытых данных должно быть прекращено, если это может привести к распространению сведений, составляющих государственную тайну или к нарушению прав обладателей информации или субъектов персональных данных.

Статья 8. Право на доступ к информации.

Статья 9. Ограничение доступа к информации.

Условия отнесения информации к сведениям, составляющим коммерческую тайну или служебную тайну устанавливаются федеральными законами. Соблюдение конфиденциальности информации, доступ к которой ограничен федеральными законами, является обязательным.

Статья 10. Распространение информации или предоставление информации.

Статья 11. Документирование информации.

Обмен электронными сообщениями, подписанных электронными подписями, при заключении гражданско-правовых договоров приравнивается обмену обычными документами, подписанными собственноручно.

Статья 12. Государственное регулирование в сфере применения информационных технологий.

Статья 13. Информационные системы.

Статья 14. Государственные информационные системы.

Статья 15. Использование информационно-телекоммуникационных сетей.

Статья 15.1. Единый реестр доменных имен, указателей страниц сайтов в сети «Интернет» и сетевых адресов, позволяющих идентифицировать сайты в сети »Интернет», содержащие информацию, распространение которой в Российской Федерации запрещено. (Эта статья введена Федеральным законом от 28.07.2012.)

Реестр создается как часть механизма, предназначенного для оперативного ограничения доступа к сайтам с запрещенным контентом. Реестр ведется и поддерживается федеральным органом исполнительной власти, осуществляющим функции по контролю и надзору в сфере средств массовой информации, массовых коммуникаций, информационных технологий и связи или оператором реестра — привлеченной организацией. Как только решение о внесении доменного имени в реестр принято, оператор реестра сообщает об этом провайдеру хостинга, который в течение суток должен уведомить об этом обслуживаемого им владельца сайта. Если в течение суток с момента получения от провайдера хостинга уведомления о включении доменного имени и (или) указателя страницы сайта в сети «Интернет» в реестр владелец сайта в сети «Интернет» не удаляет интернет-страницу, содержащую информацию, распространение которой в Российской Федерации запрещено, то провайдер хостинга обязан ограничить доступ к такому сайту.

>

Статья 15.2. Порядок ограничения доступа к информации, распространяемой с нарушением исключительных прав на фильмы, в том числе кинофильмы, телефильмы (введена Федеральным законом от 02.07.2013)

Статья 16. Защита информации.

В этой статье приводится определение понятия ***защита информации*:**

«Защита информации представляет собой принятие правовых, организационных и технических мер, направленных на:

1) обеспечение защиты информации от неправомерного доступа, уничтожения, модифицирования, блокирования, копирования, предоставления, распространения, а также от иных неправомерных действий в отношении такой информации;

2) соблюдение конфиденциальности информации ограниченного доступа;

3) реализацию права на доступ к информации.»

Перечисляются обязанности обладателя информации и оператора информационной системы, в число которых входив:

«1) предотвращение несанкционированного доступа к информации и (или) передачи ее лицам, не имеющим права на доступ к информации;

2) своевременное обнаружение фактов несанкционированного доступа к

информации;

3) предупреждение возможности неблагоприятных последствий нарушения порядка доступа к информации;

4) недопущение воздействия на технические средства обработки информации, в результате которого нарушается их функционирование;

5) возможность незамедлительного восстановления информации, модифицированной или уничтоженной вследствие несанкционированного доступа к ней;

6) постоянный контроль за обеспечением уровня защищенности информации.»

В заключении статьи приводится важное положение о том, что «Федеральными законами могут быть установлены *ограничения использования определенных средств защиты* информации и осуществления отдельных видов деятельности в области защиты информации».

Статья 17. Ответственность за правонарушения в сфере информации, информационных технологий и защиты информации.

В июле 2013 года в данный закон было внесено следующее дополнение: «Провайдер хостинга и владелец сайта в сети «Интернет» не несут ответственность перед правообладателем и перед пользователем за ограничение доступа к информации и (или) ограничение ее распространения в соответствии с требованиями настоящего Федерального закона».

Уголовный кодекс РФ

Уголовный кодекс — это юридический документ в ранге федерального закона, имеющий ограничительно-репрессивный характер. Он устанавливает принципы уголовной ответственности, определяет виды преступлений и соответствующие наказания, в том числе и в информационной сфере.

В статье 138 УК РФ определяется уголовная ответственность за незаконный доступ к конфиденциальным персональным данным, за нарушение тайны переписки, телефонных переговоров, почтовых, телеграфных или иных сообщений. Аналогичную роль для банковской и коммерческой тайны играет статья 183 УК РФ.

Преступлениям в сфере компьютерной информации посвящена глава 28 Уголовного кодекса, в ней определяются три вида киберпреступлений и соответствующие наказания:

Статья 272. Неправомерный доступ к компьютерной информации, если это повлекло уничтожение, блокирование, модификацию либо копирование информации, нарушение работы компьютера или компьютерной сети, наказывается штрафом в размере до 200 тысяч рублей, либо исправительными работами на срок до 1 года, либо лишением свободы на срок до 2 лет. За то же преступление, совершенное группой лиц по предварительному сговору или организованной группой либо лицом с использованием своего служебного положения, полагается наказание от штрафа до лишения свободы на срок до 5 лет.

Статья 273. Создание, использование и распространение вредоносных программ для ЭВМ, заведомо приводящих к несанкционированному

уничтожению, блокированию, модификации либо копированию информации, нарушению работы компьютеров, а также использование либо распространение таких программ наказывается лишением свободы на срок до 3 лет со штрафом в размере до 200 тысяч рублей, а в случае тяжких последствий — лишением свободы на срок от 3 до 7 лет.

Статья 274. Нарушение правил эксплуатации компьютерных систем лицом, имеющим к ним доступ, повлекшее уничтожение, блокирование или модификацию охраняемой законом информации, если это деяние причинило существенный вред, наказывается лишением права занимать определенные должности или заниматься определенной деятельностью на срок до 5 лет, либо обязательными работами, либо

ограничением свободы на срок до 2 лет. То же деяние, повлекшее по неосторожности тяжкие последствия, наказывается лишением свободы на срок до 4 лет.

Трудовой, гражданский кодексы и кодекс об административных правонарушениях РФ

Трудовой кодекс РФ устанавливает дисциплинарную ответственность за разглашение охраняемой законом тайны (государственной, коммерческой, служебной и иной), ставшей известной работнику в связи с исполнением им трудовых обязанностей, в том числе разглашение персональных данных другого работника.

Гр*аждански*й *кодекс РФ* в разделе «Права на результаты интеллектуальной деятельности и средства индивидуализации» регулирует один из наиболее спорных вопросов в сфере интеллектуальной собственности — ответственность интернет-провайдеров за размещение в сети контрафактных материалов. Провайдер обязывается оперативно реагировать на претензии правообладателя под угрозой привлечения к ответственности за нарушение исключительного права.

Кодекс об административных правонарушениях РФ определяет административную ответственность граждан за нарушение порядка работы с персональными данными, нарушение условий лицензированной деятельности в области защиты информации, использование несертифицированных информационных систем, баз данных, средств защиты информации, а также за разглашение информации с ограниченным доступом.

Федеральный закон «О национальной платежной системе»

Платежная система — это совокупность организаций, взаимодействующих в целях осуществления перевода денежных средств. В 2012 году был принят Закон РФ 161-ФЗ «О национальной платежной системе», в статье 27 которого устанавливаются требования к обеспечению защиты информации в платежной системе. В этом же году вышло постановление правительства *«Об утверждении положения о защите информации в платежной системе»,* в котором в обязанности участников платежной инфраструктуры вменяется обеспечивать защиту информации «от неправомерных доступа, уничтожения, модифицирования, блокирования, копирования, предоставления и распространения», соблюдать конфиденциальность информации. К числу требуемых мер относится:

«а) создание и организация функционирования структурного подразделения по защите информации (службы информационной безопасности) или назначение должностного лица (работника), ответственного за организацию защиты информации:

б) включение в должностные обязанности работников, участвующих в обработке информации, обязанности по выполнению требований к защите информации;

в) осуществление мероприятий, имеющих целью определение угроз безопасности информации и анализ уязвимости информационных систем;

г) проведение анализа рисков нарушения требований к защите информации и управление такими рисками;

д) разработка и реализация систем защиты информации в информационных системах;

е) применение средств защиты информации (шифровальные (криптографические) средства, средства защиты информации от несанкционированного доступа, средства антивирусной защиты, средства межсетевого экранирования, системы обнаружения вторжений, средства контроля (анализа) защищенности);

ж) выявление инцидентов, связанных с нарушением требований к защите информации, реагирование на них;

з) обеспечение защиты информации при использовании информационно-телекоммуникационных сетей общего пользования;

и) определение порядка доступа к объектам инфраструктуры платежной системы, обрабатывающим информацию;

к) организация и проведение контроля и оценки выполнения требований к защите информации на собственных объектах инфраструктуры не реже 1 раза в 2 года.»

Заметим, что в этом постановлении мы можем видеть, как технические задачи построения систем безопасности, а именно определение уязвимостей и угроз, оценка и управление рисками, трансформируются в правовые нормы.

Законы и нормативно-правовые акты о персональных данных

О важности проблемы защиты персональных данных (ПД) говорит хотя бы тот факт, что эта тематика отражена в 75 международных нормативно-правовых актах и почти в 200 российских федеральных законах и подзаконных актах. Рассмотрим наиболее важные из них.

Федеральный закон 160-ФЗ от 19 декабря 2005 г. *«О ратификации Конвенции Совета Европы о защите физических лиц при автоматизированной обработке персональных данных*» выражает согласие

РФ с Конвенцией, а также фиксирует ряд поправок, в том числе поправку о том, что РФ оставляет за собой право ограничивать субъекта на доступ к персональным данным о себе в целях защиты безопасности государства и общественного порядка.

Через год после ратификации Конвенции был выпущен Федеральный закон 152-ФЗ *«О персональных данных»,* целью которого является обеспечение защиты прав и свобод человека и гражданина при обработке его персональных данных, в том числе защиты прав на неприкосновенность частной жизни, личную и семейную тайну. Закон определяет персональные данные как любую информацию, относящуюся «к определенному или определяемому на основании такой информации физическому лицу (субъекту персональных данных), в том числе его фамилию, имя, отчество, год, месяц, дата и место рождения, адрес, семейное, социальное, имущественное положение, образование, профессия, доходы, другая информация». Закон регламентирует меры по обеспечению безопасности персональных данных при их обработке. Организации, обрабатывающие ПД, обязаны принимать необходимые организационные и технические меры, в том числе использовать шифровальные средства, для защиты персональных данных от неправомерного или случайного доступа к ним, уничтожения, изменения, блокирования, копирования и ПД данных, а также от иных неправомерных действий.

В соответствии с этим законом, в России все частные и государственные компании и организации, а также физические лица, которые хранят, собирают, передают или обрабатывают персональные данные (операторы ПД), должны получать от субъекта *письменное согласие на обработку его персональных данных,* а также направлять ему *уведомление о прекращении обработки* и об уничтожении персональных данных. Обязательное требование к лицу, получившему доступ к персональной информации, не передавать ее третьим лицам без согласия ее обладателя. Объем и характер ПД должен точно соответствовать объявленым целям, недопустимо собирать и обрабатывать избыточные данные, недопустимо также агрегировать различные базы персональных данных.

Последнее утверждение очень важно. Действительно, персональные данные используются в самых различных информационных системах. Различные перечни ПД устанавливаются в более 100 федеральных законах, в том числе в законах «Трудовой кодекс», «Об актах гражданского состояния», «Об индивидуальном (персонифицированном) учете в системе обязательного пенсионного страхования», «О государственной регистрации юридических лиц и индивидуальных предпринимателей» «О транспортной безопасности» и многих др. Персональные данные из отдельных перечней не должны агрегироваться, так как это потенциально во много раз увеличивает уязвимость человека. Еще в древности было известно правило, что открытие врагу мелких сведений, по отдельности несекретных, может быть опасным, поскольку сложившись вместе как частички мозаики, они могут дать целостную картину, которая возможно представляет собой большой секрет.

В целях реализации Федерального закона «О персональных данных» было принято несколько постановлений правительства, в том числе постановление от 21 марта 2012 г. № 211 «Об утверждении перечня мер, направленных на обеспечение выполнения обязанностей, предусмотренных Федеральным законом "О

персональных данных"...». В число таких мер входит разработка должностных инструкций участников процесса, типовых документов (типовое обязательство служащего, непосредственно осуществляющего обработку персональных данных, типовая форма согласия субъекта на обработку его персональных данных и др), порядка доступа в помещения, где ведется обработка данных, и других подобных документов.

Регламентирующие положения постановления правительства № 211 относятся только к *государственным* организациям, работа всех остальных, негосударственных, операторов ПД определяется нормативными актами Федеральной службы по техническому и экспортному контролю (ФСТЭК) и Федеральной службы безопасности (ФСБ).

Очень своевременным и востребованным нормативным документом оказался *приказ ФСТЭК РФ* от 11 февраля 2013 г. № 17 *«Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных».* Этот приказ устанавливает качественно новый подход к защите персональных данных, базирующийся на лучших практиках и позволяющий операторам систем обработки ПД самостоятельно выбирать оптимальный и адекватный набор средств защиты.

В документе приводится базовый набор мер по обеспечению безопасности ПД, в том числе:

- идентификация и аутентификация субъектов доступа и объектов доступа;
- управление доступом субъектов доступа к объектам доступа;
- обеспечение целостности информационной системы и персональных данных;
- обеспечение доступности персональных данных;
- регистрация событий безопасности;
- проверка системного и прикладного ПО на отсутствие недекла- рированных возможностей;

- защита машинных носителей информации для хранения ПД;
- антивирусная защита;
- обнаружение (предотвращение) вторжений;
- защита технических средств информационной системы, систем связи и передачи данных;
- выявление инцидентов, которые могут привести к сбоям или нарушению функционирования информационной системы и к возникновению угроз безопасности ПД, и реагирование на них;

В приложении к этому приказу каждая мера конкретизируется в виде перечня подзадач для разных уровней защиты ПД. Например, для «идентификации и аутентификации субъектов доступа и объектов доступа» на высшем уровне защиты ПД в Приложении к приказу перечислены следующие подзадачи:

- идентификация и аутентификация пользователей, являющихся работниками оператора;
- идентификация и аутентификация устройств, в том числе стационарных, мобильных и портативных;
- управление идентификаторами, в том числе создание, присвоение, уничтожение идентификаторов;
- управление средствами аутентификации, в том числе хранение, выдача, инициализация, блокирование средств аутентификации и принятие мер в случае утраты и (или) компрометации средств аутентификации;
- защита обратной связи при вводе аутентификационной информации;
- идентификация и аутентификация пользователей, не являющихся работниками оператора (внешних пользователей).

Та же мера, но для низшего уровня защиты не включает аутентификацию устройств.

Таким образом, на основе базового набора мер, представленного в данном Приказе и Приложении к нему, специалисты, отвечающие за информационную безопасность некоторого предприятия — оператора ПД, могут строить собственные наборы, отбирая из типового перечня те меры, которые в наибольшей степени соответствуют целям и техническим характеристикам их предприятий. Отобранные меры могут быть в дальнейшем адаптированы, уточнены и дополнены.

Правовые акты об электронной подписи

8 апреля 2011 года вступил в силу Федеральный закон № 63-ФЗ от 06.04.2011 *«Обэлектронной подписи».* Этим законом регулируются отношения в области использования электронных подписей при совершении гражданско-правовых сделок, оказании государственных и муниципальных услуг, исполнении государственных и муниципальных функций, а также при совершении иных юридически значимых действий45. В документе определяется понятие электронной подписи, устанавливаются ее виды, требования к средствам электронной подписи, с помощью которых создаются и проверяются электронная подпись, ключ электронной подписи и ключ

45 www.specremont.su/library/federalnoe-zakonodatelstvo/federalnyij-zakon- 63-fz-ob-elektronnoj-podpisi.html

проверки электронной подписи. Средства, с помощью которых осуществляется создание подписи и её проверка, должны соответствовать установленным законом требованиям и содержать элементы криптографии. Федеральным законом «Об электронной подписи» устанавливаются требования к удостоверяющим центрам, осуществляющим функции по созданию и выдаче сертификатов ключей проверки электронных подписей.

Правовые акты о лицензировании отдельных видов деятельности

Закон 99-ФЗ от 4 мая 2011 г. *«О лицензировании отдельных видов деятельности»:* В законе приводятся определения понятий «лицензия», «лицензирование», «лицензиат» и др. Указывается, что лицензированию подлежат практически все виды деятельности, связанные с разработкой, производством, обслуживанием и распространением шифровальных и других средств защиты конфиденциальной информации, оказанием услуг в области шифрования.

Особенности лицензирования образовательной деятельности, в том числе связанной с информационной безопасностью, затрагиваются в постановлении правительства «Об утверждении Положения о лицензировании образовательной деятельности», которое вступило в силу с 1 сентября 2013 года. В Положении корректирован перечень лицензионных требований, предъявляемых к лицензиату, в частности говорится о необходимости «наличия у соискателя лицензии или лицензиата при реализации образовательного процесса, содержащего сведения, составляющие государственную тайну, средств вычислительной техники и программного обеспечения, удовлетворяющих требованиям законодательства Российской Федерации по режиму секретности и технической защите информации, а также лицензии на осуществление работ с использованием сведений, составляющих государственную тайну.»

ПРИЛОЖЕНИЕ 2. СТЕКИ КОММУНИКАЦИОННЫХ ПРОТОКОЛОВ

Многоуровневый подход

Организация взаимодействия устройств сети является сложной задачей. Для решения сложных задач используется известный универсальный прием — *декомпозиция,* т. е. разбиение одной сложной задачи на несколько более простых задач-модулей.

Еще более эффективной концепцией, развивающей идею декомпозиции, является *многоуровневый подход.* После представления исходной задачи в виде множества модулей эти модули группируют и упорядочивают по уровням, образующим *иерархию* (рис. П.1).

Каждый вышестоящий уровень использует нижестоящий в качестве инструмента для решения своих задач. Так группа модулей, находящихся на верхнем уровне иерархии, может обращаться с запросами на выполнение тех или иных функций только к модулям непосредственно прилегающего нижнего уровня 2, а модули уровня 2 в свою очередь могут обращаться за услугами к модулям уровня 1. С другой стороны, результаты работы каждого из модулей, отнесенных к некоторому уровню, могут быть переданы только модулям соседнего вышележащего уровня. Такая иерархическая декомпозиция задачи предполагает четкое определение функций и интерфейсов не только отдельных модулей, но и каждого уровня.

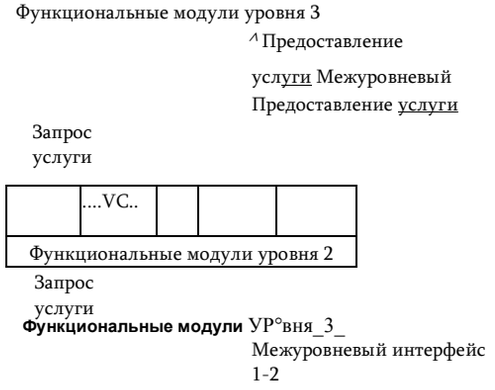


Рис. П.2. Многоуровневый подход — создание иерархии задач

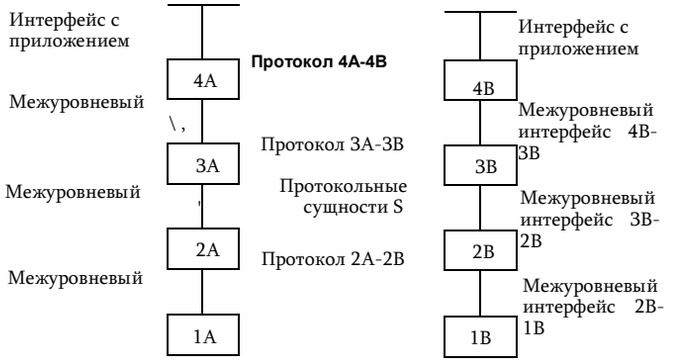


Рис. П.2. Протоколы и интерфейсы многоуровневых средств взаимодействия двух узлов

Межуровневый интерфейс, называемый также *интерфейсом услуг,* определяет набор функций (услуг), которые нижележащий уровень предоставляет вышележащему.

Задача организации взаимодействия компьютеров в сети тоже может быть представлена в виде иерархически организованного множества модулей. Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют, по меньшей мере, две стороны, т. е. в данном случае необходимо организовать согласованную работу *двух иерархий* аппаратных и программных средств, работающих на разных компьютерах.

Оба участника сетевого обмена должны принять множество соглашений на каждом уровне взаимодействия. Например, они должны согласовать уровни и форму электрических сигналов, перечень сообщений и их формат, договориться о методах контроля достоверности и т. п.

Рисунок П.2 поясняет основные идеи и термины многоуровневой организации сетевых средств на примере взаимодействия двух сетевых узлов А и В. С каждой стороны средства сетевого взаимодействия представлены четырьмя уровнями (количество уровней в этом примере условно). Каждый уровень (кроме самого нижнего) поддерживает три интерфейса. Во-первых, это два интерфейса услуг с выше- и нижележащим уровнями «своей» иерархии средств. Во-вторых, это интерфейс со средствами взаимодействия другой стороны, расположенными на том же уровне иерархии. Последний тип интерфейса называют *протоколом.*

Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней, как правило, программными средствами. Модули, поддерживающие протокол, а также межуровневые интерфейсы, называют *протокольными сущностями,* или, для краткости, также *протоколами*.

Протокольные сущности одного уровня двух взаимодействующих сторон обмениваются сообщениями в соответствии с определенным для них протоколом. Сообщения состоят из заголовка и поля данных (иногда оно может отсутствовать). Обмен сообщениями является своеобразным языком общения, с помощью которого каждая из сторон «объясняет» другой стороне, что необходимо сделать на каждом этапе взаимодействия. Работа каждого протокольного модуля состоит в интерпретации заголовков поступающих к нему сообщений и выполнении связанных с этим действий. Заголовки сообщений разных протоколов имеют разную структуру, что соответствует различиям в их функциональности. Протокольные сущности одного уровня не общаются между собой непосредственно, в этом общении всегда участвуют посредники — средства протоколов нижележащих уровней. И только самые нижние уровни двух иерархий взаимодействуют напрямую.

Многоуровневый набор протоколов, достаточный для организации взаимодействия узлов в сети, называется *стеком коммуникационных протоколов*.

Модель OSI

К концу 70-х годов в мире уже существовало большое разнообразие в стеках коммуникационных протоколов, среди которых можно назвать, например, такие популярные стеки, как DECnet (фирменный стек компании Digital), TCP/IP (стек Интернета) и SNA (фирменный стек компании IBM). Однако компьютеры и другие сетевые устройства, поддерживающие разные стеки протоколов, будучи помещенными в одну сеть, «отказывались» работать друг с другом. Одним из путей преодоления такой несовместимости в то время виделся всеобщий переход на единый стек протоколов, который бы аккумулировал в себе все лучшее, что было в других уже существующих стеках. Именно с таких академических позиций несколько международных организаций по стандартизации[46] подошли к разработке нового стека коммуникационных протоколов. Важнейшим результатом их работы стало создание стандартной *модели взаимодействия открытых систем* (Open System Interconnection, OSI).

Модель OSI делит средства взаимодействия на *семь* уровней, за которыми закреплены следующие названия.

Прикладной уровень (application layer) — это в действительности просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким, как файлы, принтеры или гипертекстовые веб-страницы, а также организуют свою совместную работу, например по протоколу электронной почты. Единица данных, которой оперирует прикладной уровень, обычно называется *сообщением*.

Уровень представления (presentation layer), как явствует из его названия, обеспечивает представление передаваемой по сети информации, не меняя при этом ее содержания. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия в кодах символов, например кодов ASCII и EBCDIC. На этом уровне могут выполняться шифрование и дешифрирование данных, благодаря которым секретность обмена данными обеспечивается сразу для всех прикладных служб.

Сеансовый уровень (session layer) обеспечивает управление взаимодействием сторон: фиксирует, какая из сторон является активной в настоящий момент, и предоставляет средства синхронизации сеанса. Функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе.

Транспортный уровень (transport layer) обеспечивает приложениям или верхним уровням стека — прикладному, представления и сеансовому — передачу данных с той степенью надежности, которая им требуется.

Сетевой уровень (network layer) служит для образования единой транспортной системы, объединяющей несколько сетей, называемой составной сетью. Решение этой задачи возлагается на конечные узлы и маршрутизаторы и включает следующие частные задачи: определение маршрута, продвижение данных по этому маршруту, согласование технологий при передаче данных между сетями, построенными на разных технологиях, создание барьеров на пути нежелательного трафика между сетями. Данные, которые необходимо передать через составную сеть, поступают на сетевой уровень от вышележащего транспортного уровня, снабжаются заголовком сетевого уровня, образуя *пакет.* К сетевому уровню относят также *протоколы маршрутизации,* с помощью которых маршрутизаторы собирают информацию о топологии межсетевых соединений, на основании которой осуществляется выбор маршрута продвижения пакетов.

---

46 В частности, в число этих организаций входили International Organization for Standardization (ISO), часто называемая также International Standards Organization, а также International Telecommunications Union (ITU) и некоторые другие.

Канальный уровень (data link layer) обеспечивает прозрачность соединения для сетевого уровня. Для этого он предлагает ему следующие услуги: установление логического соединения между взаимодействующими узлами, согласование в рамках соединения скоростей передатчика и приемника, обеспечение надежной передачи. Для решения этих задач канальный уровень формирует из пакетов собственные протокольные единицы данных — *кадры.* Канальный уровень помещает пакет в поле данных одного или нескольких кадров и заполняет собственной служебной информацией заголовок кадра. Протокол канального уровня обычно работает в пределах сети, являющейся одной из составляющих более крупной составной сети, объединенной протоколами сетевого уровня. Адреса, с которыми работает протокол канального уровня, используются для доставки кадров только в пределах этой сети, а для перемещения пакетов между сетями применяются уже адреса следующего, сетевого, уровня. Протоколы канального уровня реализуются как на конечных узлах (средствами сетевых адаптеров и их драйверов), так и на всех промежуточных сетевых устройствах.

Физический уровень (physical layer) поддерживает интерфейс с канальным уровнем. На стороне отправителя физический уровень получает с канального уровня кадры, которые рассматривает как неструктурированный поток битов, которые он должен передать по физическим каналам связи, таким, как коаксиальный кабель, витая пара, оптоволоконный кабель или цифровой территориальный канал.

Распределение функций между различными элементами сети

Функциональность полного стека протоколов может быть востребована только конечными узлами, а коммуникационные устройства — маршрутизаторы и коммутаторы, решающие задачу транспортировки сообщений между конечными узлами, как правило, ограничиваются поддержкой функциональности нижних трех уровней.

На рис. П.3 показаны основные элементы компьютерной сети: конечные узлы — компьютеры и промежуточные узлы — коммутаторы и маршрутизаторы.

Коммутаторы обычно поддерживают функции двух нижних уровней, физического и канального, что ограничивает их возможности передачей данных в пределах только одной подсети. Однако некоторые коммутаторы, работающие на основе технологии виртуальных каналов, могут поддерживать как два уровня протоколов, так и три.

Маршрутизаторы служат пограничными устройствами, разделяя составную сеть на подсети. Они поддерживают функции всех трех

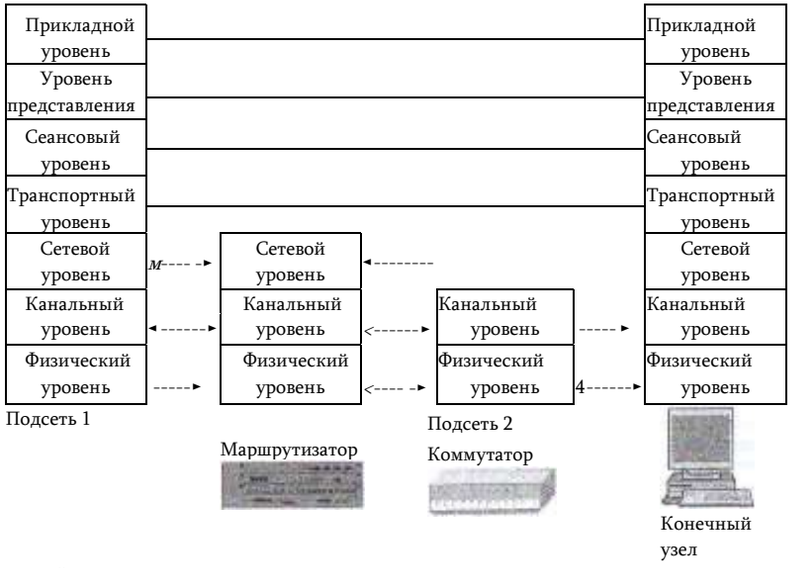


Рис. П.3. **Распределение функций между различными элементами сети** нижних уровней, так как сетевой уровень нужен им для объединения подсетей различных технологий в составную сеть и нахождения маршрута между конечными узами через составную сеть, а функции нижних уровней — для передачи данных в пределах отдельных подсетей.

Компьютеры, на которых работают сетевые приложения, в общем случае поддерживают функции всех уровней, которые реализуются операционной системой или системными приложениями (кроме физического и части функций канального уровня). Протоколы прикладного уровня, пользуясь сервисами протоколов уровня представления и сеансового уровня, предоставляют приложениям набор сетевых услуг в виде сетевого интерфейса API.

Протокол транспортного уровня также работает на конечных узлах. При передаче данных через сеть два модуля транспортного протокола, работающие на узле-отправителе и узле-получателе, взаимодействуют для поддержания транспортного сервиса нужного качества. Коммуникационные устройства сети переносят сообщения транспортного протокола прозрачным образом, не вникая в их содержание.

В реальных сетях некоторые из коммуникационных устройств поддерживают не только протоколы трех нижних уровней, но и протоколы верхних уровней. Так, маршрутизаторы реализуют протоколы маршрутизации, позволяющие автоматически строить таблицы маршрутизации, а коммутаторы часто поддерживают протоколы SNMP и telnet, которые не нужны для выполнения основных функций этих устройств, но позволяют конфигурировать и управлять ими удаленно. Все эти

протоколы являются протоколами прикладного уровня и выполняют некоторые вспомогательные функции транспортной системы.

Стек протоколов TCP/IP

Стек TCP/IP-имеет иерархическую структуру, в которой определены четыре уровня (рис. П.4).

Прикладной уровень	FTP, Telnet, HTTP, SMTP, SNTP, TFTP
Транспортный уровень	TCP, UDP
Сетевой уровень	IP, ICMP, RIP, OSPF
Уровень сетевых интерфейсов	Не регламентируется

Рис. П.4. **Иерархическая структура стека TCP/IP**

Прикладной уровень стека TCP/IP-соответствует трем верхним уровням модели OSI: прикладному, представления и сеансовому. За долгие годы использования стек TCP/IP накопил большое количество протоколов и служб прикладного уровня: протокол передачи файлов (File Transfer Protocol, FTP), протокол эмуляции терминала (telnet), простой протокол передачи электронной почты (Simple Mail Transfer Protocol, SMTP), протокол передачи гипертекста (Hypertext Transfer Protocol, HTTP) и многие другие. Протоколы прикладного уровня развертываются на хостах47.

Транспортный уровень стека TCP/IP может предоставлять вышележащему уровню два типа сервиса:

- гарантированную доставку обеспечивает *TCP* (Transmission Control Protocol);
- доставку «по возможности» обеспечивает протокол *UDP* (User Datagram Protocol).

Программные модули, реализующие протоколы TCP и UDP, подобно модулям протоколов прикладного уровня, устанавливаются на хостах.

Сетевой уровень является стержнем всей архитектуры TCP/IP. Именно этот уровень, функции которого соответствуют сетевому уровню модели OSI, обеспечивает перемещение пакетов в пределах составной сети, образованной объединением нескольких подсетей. Протоколы сетевого уровня поддерживают интерфейс с вышележащим транспортным уровнем, получая ют него запросы на передачу данных по составной сети, а также с нижележащим уровнем сетевых интерфейсов.

Основным протоколом сетевого уровня является *межсетевой протокол IP* (Internet Protocol). В его задачу входит продвижение

47 В Интернете (а значит, и в стеке протоколов TCP/IP) конечный узел традиционно называют *хостом,* **а маршрутизатор —** *шлюзом.*

пакета между сетями — от одного маршрутизатора до другого до тех пор, пока пакет не попадет в сеть назначения, внутри которой доставку в узел назначения обеспечивает локальная технология данной сети. Протокол IP развертывается не только на хостах, но и на всех маршрутизаторах. Протокол IP — это дейтаграммный протокол, работающий без установления соединений и обеспечивающий доставку «по возможности». К сетевому уровню TCP/IP часто относят протоколы маршрутизации *RIP* и *OSPF,* занимающиеся изучением топологии сети, определением маршрутов и составлением и таблиц маршрутизации, на основании которых протокол IP перемещает пакеты в нужном направлении. По этой же причине к сетевому уровню могут быть отнесены еще два протокола: протокол межсетевых управляющих сообщений *ICMP* (Internet Control Message Protocol), предназначенный для передачи маршрутизатором источнику информации об ошибках, возникших при передаче пакета, и некоторые другие протоколы.

Идеологическим отличием архитектуры стека TCP/IP от многоуровневой организации других стеков является интерпретация функций самого нижнего уровня — *уровня сетевых интерфейсов.* Напом- ним, что нижние уровни модели OSI (канальный и физический) реализуют большое количество функций доступа к среде передачи, формированию кадров и согласованию уровней электрических сигналов, кодированию и синхронизации и некоторые другие. Все эти весьма конкретные функции составляют суть таких протоколов обмена данными, как Ethernet, PPP и многих других. У нижнего уровня стека TCP/IP задача существенно проще — он отвечает *только* за организацию взаимодействия с технологиями сетей, входящих в составную сеть. TCP/IP рассматривает *любую* подсеть, входящую в составную сеть, как средство транспортировки пакетов между двумя соседними маршрутизаторами.

Задачу обеспечения интерфейса между технологией TCP/IP и любой другой технологией промежуточной сети упрощенно можно свести:

- к определению способа *инкапсуляции* (упаковки) IP-пакета в единицу передаваемых данных промежуточной сети;
- к определению способа преобразования сетевых адресов в адреса технологии данной промежуточной сети (примером такого протокола является ARP).

Каждый коммуникационный протокол оперирует некоторой единицей передаваемых данных. Названия этих единиц иногда закрепляются стандартом, а чаще просто определяются традицией. В стеке TCP/IP-за многие годы его существования образовалась устоявшаяся терминология в этой области. *Потоком данных* называют дан

ные, поступающие от приложений на вход протоколов транспортного уровня TCP и UDP. Протокол TCP «нарезает» из потока данных *сегменты.* Единицу данных протокола UDP часто называют *дейтаграммой* или *датаграммой.* Дейтаграмма — это общее название для единиц данных, которыми оперируют протоколы без установления соединений. К таким протоколам относится и протокол IP, поэтому его единицу данных иногда также называют дейтаграммой. Однако чаще используется и другой термин — *пакет.* В стеке TCP/IP принято называть *кадрами,* или *фреймами* единицы данных любых технологий, в которые упаковываются IP-пакеты для последующей переноски их через сети составной сети. При этом не имеет значения, какое название используется для этой единицы данных в технологии составляющей сети.

Типы адресов стека TCP/IP

Для идентификации сетевых интерфейсов используются три типа адресов:

Локальные (аппаратные) адреса. В разных сетевых технологиях в общем случае используются собственные системы адресации, которые используются исключительно для обеспечения связи собственных узлов. Роль, которую играют эти адреса в TCP/IP, не зависит от того, какая именно технология используется в подсети, поэтому они имеют общее название — локальные (аппаратные) адреса. Например, если в составную сеть включена подсеть Ethernet, то локальными адресами сетевых интерфейсов этой сети для технологии TCP/1P-будут соответственно MAC-адреса, а если подсеть ATM — то в качестве локальных адресов будут выступать номера виртуальных каналов.

Сетевые адреса (IP-адреса). Чтобы технология TCP/1P-могла решать свою задачу объединения сетей, ей необходима собственная глобальная система адресации, позволяющая универсальным и однозначным способом идентифицировать любой интерфейс составной сети. Очевидным решением является нумерация всех подсетей составной сети, а затем нумерация сетевых интерфейсов в пределах каждой из этих подсетей. Пара, состоящая из номера сети и номера узла, отвечает поставленным условиям и может служить в качестве сетевого адреса, или IP-адреса. Сетевой адрес представляет собой набор чисел, например 192.45.66.17.

Символьные, или *доменные,* или *DNS-имена* более удобно использовать в обычной жизни. Символьные имена в пределах составной сети строятся по иерархическому признаку. Примером доменного имени может служить имя base2.saleszil.ru.

Формат IP-адреса

В заголовке IP-пакета для хранения IP-адреса отводится 4 байта (32 бита). IP-адрес состоит из двух логических частей -*у номера сети* и *номера узла* в сети. Длина каждой из частей может варьироваться от одного адреса к другому. Компьютерная форма представления адреса *не предусматривает специального разграничительного знака* между номером сети и номером узла. Вместе с тем при передаче пакета по сети часто возникает необходимость автоматическими

средствами разделить адрес на эти две части. Например, маршрутизация, как правило, осуществляется на основании номера сети, поэтому каждый маршрутизатор, получая пакет, должен прочитать из соответствующего поля заголовка адрес назначения и выделить из него номер сети. Каким образом маршрутизаторы определяют, какая часть из 32 битов, отведенных под IP-адрес, относится к номеру сети, а какая — к номеру узла? Для этих целей используется два подхода.

Первый способ основан на использовании *маски* — числа, применяемого в паре с IP-адресом. Двоичная запись маски содержит непрерывную последовательность единиц в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Например, если маска, связываемая с некоторым IP-адресом, имеет вид 11111111111100000 000000000000000, то номеру сети соответствуют 10 старших разрядов в двоичном представлении данного IP-адреса.

Второй способ заключается в использовании *классов адресов.* Вводится пять классов адресов: A, B, C, D, E. Три из них (A, B и C) используются для адресации сетей, а два (D и E) имеют специальное назначение. Для каждого класса сетевых адресов определено собственное положение границы между номером сети и номером узла. Признаком, на основании которого IP-адрес относится к тому или иному классу, являются значения нескольких первых битов адреса. Рис. П.5 иллюстрирует структуру IP-адресов разных классов.

Если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такой адрес называется *ограниченным широковещательным* (limited broadcast). Ограниченность в данном случае означает, что пакет не выйдет за границы данной подсети не при каких условиях.

Если в поле адреса назначения в разрядах, соответствующих номеру узла, стоят только единицы, то пакет, имеющий такой адрес, рассылается *всем* узлам сети, номер которой указан в адресе назначения. Например, пакет с адресом 192.190.21.255 будет направлен всем 254 узлам сети 192.190.21.0. Такой тип адреса называется *широковещательным* (broadcast).

Рис. П.б. Классы IP-адресов

Заголовок IP-пакета

На рис. П.б показана структура заголовка IP-пакета.

Поле *номера версии* занимает 4 бита и идентифицирует версию протокола IP. Сейчас наиболее используемой является версия 4 (IPv4), хотя все чаще встречается и новая версия (IPv6).

Значение *длины заголовка* IP-пакета также занимает 4 бита и измеряется в 32-битовых словах. Обычно заголовок имеет длину в 20 байт (пять 32-битовых слов), но при добавлении некоторой служебной информации это значение может быть увеличено за счет дополнительных байтов в поле параметров до 60 байтов.

Поле *типа сервиса* (Type of Service, ToS) или *байт дифференцированного обслуживания* (DS-байт) служит для хранения признаков, которые отражают приоритет пакета и критерии выбора маршрута (задержка, пропускная способность или надежность).

Рис. П.б. Структура заголовка IP-пакета

Поле *общей длины* занимает 2 байта и содержит общую длину пакета с учетом заголовка и поля данных.

Следующие несколько полей: *идентификатор пакета* (2 байта), *флаги MF* и *DF* (каждый по 1 биту и один бит резервный), *смещение фрагмента* (13 бит) и *время жизни* (1 байт) используются при выполнении фрагментации, то есть деления пакета на части.

Поле *протокола верхнего уровня* занимает один байт и содержит идентификатор, указывающий, какому протоколу верхнего уровня принадлежит информация, размещенная в поле данных пакета. Значения идентификаторов для разных протоколов приводятся в документе RFC 1700. Например, 6 означает, что в пакете находится сообщение TCP, 17 — сообщение UDP, 1 — сообщение ICMP.

Контрольная сумма заголовка занимает 2 байта (16 битов) и рассчитывается только по заголовку. Поля IP-адресов источника и приемника имеют одинаковую длину — 32 бита.

Поле *параметров* является необязательным и используется обычно только при отладке сети.

Порты

В то время как задачей уровня межсетевого взаимодействия, к которому относится протокол IP, является передача данных между сетевыми интерфейсами в составной сети, главная задача протоколов транспортного уровня TCP и UDP заключается в передаче данных между *прикладными процессами,* выполняющимися на компьютерах в сети.

Каждый компьютер может выполнять несколько процессов, более того, даже отдельный прикладной процесс может иметь несколько точек входа, выступающих в качестве адреса назначения для пакетов данных. Поэтому доставка данных на сетевой интерфейс компьютера- получателя — это еще не конец пути: данные необходимо переправить конкретному процессу-получателю. Процедура распределения протоколами TCP и UDP поступающих от сетевого уровня пакетов между прикладными процессами называется *демультиплексированием* (рис. П.7).

Существует и обратная задача: данные, генерируемые разными приложениями, работающими на одном конечном узле, должны быть переданы общему для всех них протокольному модулю IP для последующей отправки в сеть. Протоколы TCP и UDP выполняют эту работу, называемую *мультиплексированием.*

Протоколы TCP и UDP ведут для каждого приложения две системные очереди: очередь данных, поступающих к приложению из сети, и очередь данных, отправляемых этим приложением в сеть. Такие

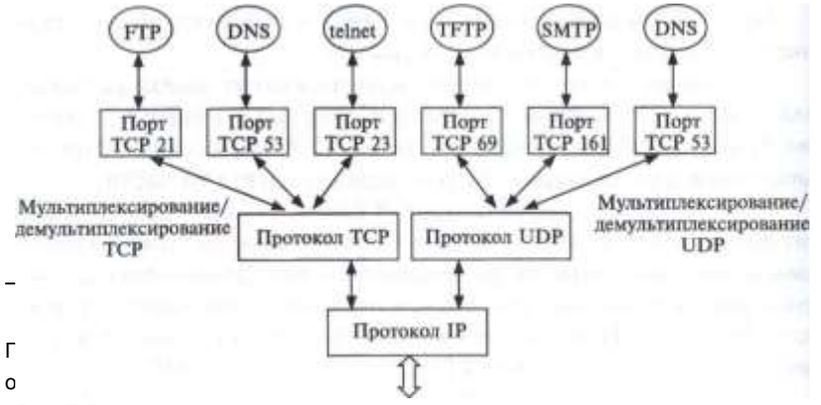


Рис. П.7. **Мультиплексирование и демультиплексирование на транспортном уровне**

Приложение 2. Стеки коммуникационных протоколов 617

системные очереди называются *портами48,* причем входная и выходная очереди одного приложения рассматриваются как один порт. Для идентификации портов им присваивают номера.

Если процессы представляют собой популярные системные службы, такие, как FTP, telnet, HTTP, TFTP, DNS и т. п., то за ними закрепляются *стандартные назначенные* номера, также называемые *хорошо известными* (wellknown) номерами портов. Эти номера закрепляются и публикуются в стандартах Интернета (RFC 1700, RFC 3232). Так, номер

2 байта

Порт источника (source port)

21 закреплен за серверной частью службы удаленного доступа к файлам FTP, а 23 — за серверной частью службы удаленного управления telnet. Назначенные номера являются *уникальными в пределах Интернета* и назначаются приложениям *централизованно* из диапазона от 0 до 1023.

Для тех приложений, которые еще не стали столь распространенными, номера портов назначаются *локально* разработчиками этих приложений или операционной системой в ответ на поступление запроса от приложения. На каждом компьютере операционная система ведет список занятых и свободных номеров портов. При поступлении запроса от приложения, выполняемого на данном компьютере, операционная система выделяет ему первый свободный номер. Такие номера называют *динамическими.* В дальнейшем все сетевые приложения должны адресоваться к данному приложению с указанием назначенного ему динамического номера порта. После того как приложение завершит работу, его номер возвращается в список свободных и может быть назначен другому приложению. Динамические номера

являются *уникальными в пределах каждого компьютера,* но при этом обычной ситуацией является совпадение номеров портов приложений, выполняемых на разных компьютерах. Как правило, клиентские части известных приложений (DNS, WWW, FTP, telnet и др) получают динамические номера портов от ОС.

Все, что было сказано о портах, в равной степени относится к обоим протоколам транспортного уровня (TCP и UDP). Приложения, которые передают данные на уровень IP, используя протокол UDP, получают номера, называемые *UDP-портами.* Аналогично, приложениям, обращающимся к протоколу TCP, выделяются *TCP-порты.* Совпадение номеров TCP- и UDP-портов, как правило, ничего не обозначает. Лишь в некоторых случаях, когда приложение может обращаться по выбору к протоколу TCP или UDP (как, например, DNS- сервер), ему, исходя из удобства запоминания, назначаются совпадающие стандартные номера TCP- и UDP-портов (в данном примере — это *хорошо известный* номер 53).

Заголовки UDP- и TCP-сегментов

Заголовок UDP состоит из четырех 2-байтовых полей: номера UDP-порта отправителя, номера UDP-порта получателя, контрольной суммы и длины дейтаграммы.

Рис. П.8. **Формат заголовка TCP**

Заголовок TCP (рис. П.8) содержит значительно больше полей, чем заголовок UDP, что отражает более развитые возможности первого протокола. На рисунке помещено краткое пояснение содержимого большинства полей.

Коротко поясним значение однобитовых полей, называемых *флагами* или *кодовыми битами* (code bits). Они расположены сразу за резервным полем и содержат служебную информацию о типе данного сегмента. Положительное значение сигнализируется установкой этих битов в единицу: *URG* — срочное сообщение; *ACK* — квитанция на принятый сегмент; *PSH* — запрос на отправку сообщения без ожидания заполнения буфера; *RST*— запрос на восстановление соединения; *SYN* — сообщение, используемое для синхронизации счетчиков переданных данных при установлении соединения; *FIN* — признак достижения передающей стороной последнего байта в потоке передаваемых данных.

Сокеты

Стандартные назначенные номера портов уникально идентифицируют тип приложения (FTP или HTTP или DNS и т.д), однако они не могут использоваться для однозначной идентификации прикладных процессов, связанных с каждым из этих типов приложений. Пусть, например, на одном хосте запущены две копии DNS-сервера — DNSi, DNS_2 (рис. П.9). Каждый из этих DNS-серверов имеет хорошо известный UDP-порт 53. Какому из этих серверов нужно было бы направить запрос клиента, если бы в DNS-запросе в качестве идентификатора сервера был указан только номер порта?

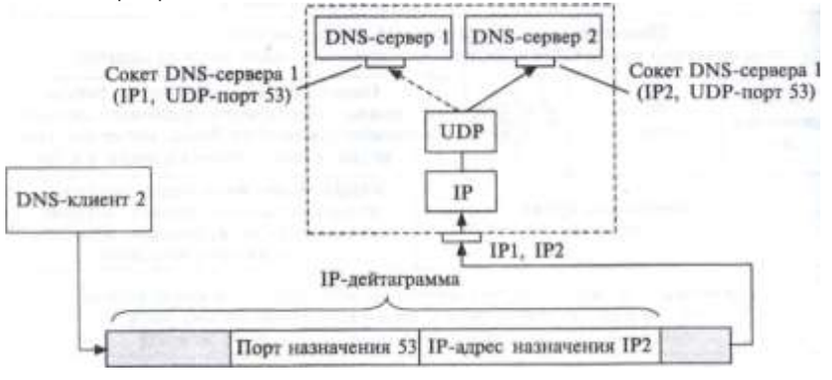
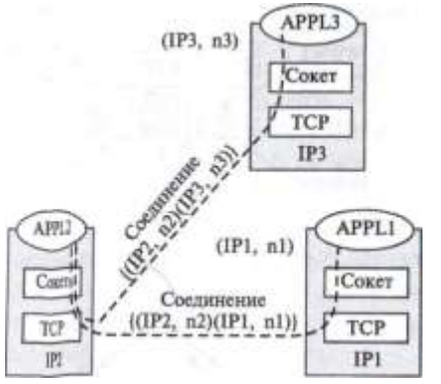


Рис. П.9. Демультеплекирование на основе сокетов

Чтобы снять неоднозначность в идентификации приложений, разные копии связываются с разными IP-адресами. Для этого сетевой интерфейс компьютера, на котором выполняется несколько копий приложения, должен иметь, соответствующее число IP-адресов, на рисунке это $1P_X$ и $1P_2$. Во всех IP-пакетах,



направляемых серверу DNSj, в качестве IP-адреса указывается $1P_ь$ а серверу DNS_2 — адрес $1P_2$. Поэтому показанный на рисунке пакет, в поле данных которого содержится UDP-дейтагрзмма, с указанием номера порта 53, а в поле заголовка указан адрес $1P_2$ буден направлен однозначно определенному адресату — DNS_2.

Прикладной процесс однозначно определяется в пределах сети и в пределах отдельного компьютера парой (IP-адрес, номер порта), называемой *сокетом* (socket).

Логическое TCP-соединение однозначно идентифицируется *парой* сокетов, определенных для этого соединения двумя взаимодействующими процессами. Сокет одновременно может участвовать в нескольких соединениях. Так (рис. П.10), показаны три компьютера с адресами IP1, IP2, IP3. На каждом компьютере выполняется по одному приложению — APPL1, APPL2 и APPL 3, сокеты которых (IP1, n 1), (IP2, п2), (IP3, п3) соответственно, а nl, п2, п3 — номера TCP- портов приложений.

На рисунке показаны два логических соединения, которое установило приложение 2 с приложением 1 и приложением 3. Логические соединения идентифицируются как {(IP2, n2), (IP1, nl)} и {(IP2, п2), (IP3, п3)} соответственно. Мы видим, что в обоих соединениях участвует один и тот же сокет — (IP2, п2).

А теперь рассмотрим на примере, как протокол TCP выполняет демультиплексирование. Пусть некий поставщик услуг оказывает

(IP2, n2)

Рис. П.10. Один сокет **может участвовать в нескольких соединениях**

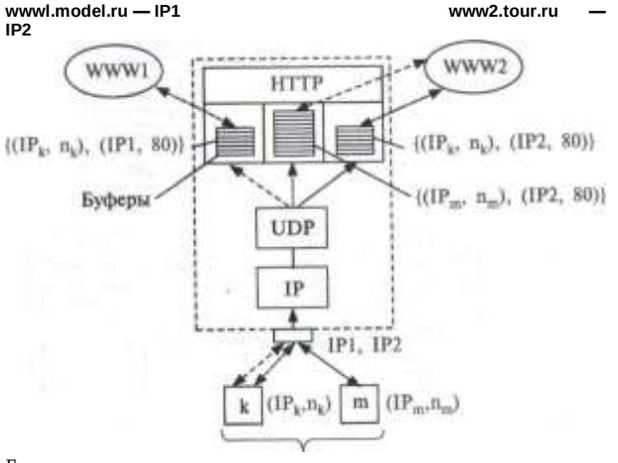


Рис. П.11. Демультиплексирование TCP на основе соединений

услугу по веб-хостингу, т. е. на его компьютере клиенты могут устанавливать свои веб-серверы. Веб-сервер основан на протоколе прикладного уровня HTTP, который передает свои сообщения в сегментах TCP. TCP ожидает запросы от веб-клиентов (браузеров), «прослушивая» хорошо известный порт 80.

На рис. П.11 показан вариант хостинга с двумя веб-серверами — сервером wwwl.model.ru, имеющим IP-адрес IP1, и сервером www2. tour.ru с адресом IP2. К каждому из них может обращаться множество клиентов, причем клиенты могут одновременно работать как с сервером wwwl, так и с сервером www2. Для каждой пары клиент-сервер протоколом TCP создается *отдельное логическое соединение*.

На рисунке показаны два браузера, имеющие соответственно сокеты (IPk, nk) и (IPm, nm). Пользователь браузера к обращается одновременно к серверам WWW1 и WWW2. Наличие отдельных соединений для работы с каждым из этих серверов не только служит обеспечению надежной доставки, но и гарантирует разделение информационных потоков — у пользователя никогда не возникает вопроса, каким сервером ему была послана та или иная страница. Одновременно с пользователем браузера к с сервером WWW2 работает пользователь браузера т. И в этом случае отдельные логические соединения, в рамках которых идет работа обоих пользователей, позволяют изолировать их информационные потоки. На рисунке показаны буферы, количество которых определяется не числом веб-серверов, и не числом клиентов, а числом логических соединений. Сообщения в эти буферы направляются в зависимости от значений сокетов как отправителя, так и получателя.

Протокол ICMP. Утилита ping

Протокол ICMP) является вспомогательным протоколом, использующимся для диагностики и мониторинга сети. На рис. П.12 показана таблица основных типов ICMP-сообщений. Эти сообщения можно разделить на две группы: сообщения об ошибках и сообщения запрос-ответ.

Сообщения, относящиеся к группе сообщений об ошибке, конкретизируются уточняющим кодом. На рисунке показан фрагмент таблицы кодов для сообщения об ошибке «Узел назначения недостижим», имеющей тип 3. Аналогичные таблицы кодов существуют и для других типов сообщений об ошибке.

Сообщения типа запрос и ответ связанны в пары: эхо-запрос — эхо-ответ, запрос маски — ответ маски, запрос времени — ответ времени. Отправитель сообщения-запроса всегда рассчитывает на получение соответствующего сообщения-ответа.

Эхо-запрос и эхо-ответ, в совокупности называемые *эхо-протоколом,* представляют собой очень простое средство мониторинга сети. Компьютер или маршрутизатор посылает по составной сети ICMP-сообщение эхо-запрос, указывая в нем IP-адрес узла, достижимость которого нужно проверить. Узел, получивший эхо-запрос, формирует и отправляет эхо-ответ отправителю запроса. Так как эхо-запрос и эхо-ответ передаются по сети внутри IP-пакетов, то их успешная доставка означает нормальное функционирование всей транспортной системы составной сети.

Рис. П.12. Типы и коды ICMP-сообщений

Таблица типов ICMP-сообщений

Значение поля «Тип»	Тип сообщения
0	Эхо-ответ
3	Узел назначения недостижим
4	Подавление источника
5	Перенаправление маршрута
8	Эхо-запрос
11	Истечение времени диаграммы
12	Проблема с параметром пакета
13	Запрос отметки времени
14	Ответ отметки времени
17	Запрос маски
18	Ответ маски

Код	Причина
0	Сеть не достижима
1	Узел достижим
2	Протокол не достижим
3	Порт не достижим
4	Ошибка фрагментации
5	Ошибка в маршруте источника
6	Сеть назначения неизвестна
7	Узел назначения неизвестен
8	Узел-источник изолирован
9	Административный запрет

? сообщение-запрос ф сообщение-ответ
•/сообщение-ошибка

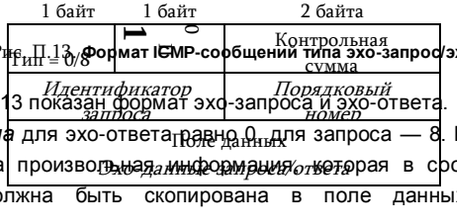


Рис. П.13. **Формат ICMP-сообщений типа эхо-запрос/эхо-ответ**

На рис. П.13 показан формат эхо-запроса и эхо-ответа.

Поле *типа* для эхо-ответа равно 0, для запроса — 8. В поле данных может быть помещена произвольная информация, которая в соответствии с данным протоколом должна быть скопирована в поле данных эхо-ответа. Поля «идентификатор запроса» и «порядковый номер» используются одинаковым образом всеми сообщениями типа запрос-ответ. Посылая запрос, приложение помещает в эти два поля информацию, которая предназначена для последующего встраивания ее в соответствующий ответ. Сообщение-ответ копирует значения этих полей в свои поля этого же назначения. Когда ответ вернется в пункт отправки сообщения-запроса, на основании идентификатора он сможет «найти и опознать» приложение, которое послало этот запрос. А порядковый номер будет использован приложением, чтобы связать полученный ответ с соответствующим запросом (учитывая, что одно приложение может выдать несколько однотипных запросов).

На основе эхо-протокола построена широко используемая утилита ping, предназначенная для тестирования достижимости узлов. Эта утилита обычно посылает серию эхо-запросов к тестируемому узлу и предоставляет пользователю статистику об утерянных эхо-ответах и среднем времени реакции сети на запросы. Утилита ping выводит на экран сообщения следующего вида обо всех поступивших ответах:

```
# ping server23.citmgu.ru
Pinging server23.citmgu.ru [193.107.2.200] with 64 bytes of data:
Reply from 193.107.2.200:       bytes=64 time=256ms TTL= 123
Reply from 193.107.2.200:       bytes=64 time=310ms TTL= 123
Reply from 193.107.2.200:       bytes=64 time=260ms TTL= 123
Reply from 193.107.2.200:       bytes=64 time=146ms TTL= 123
```

Из приведенной распечатки видно, что в ответ на тестирующие запросы, посланные узлу server23.mguru, было получено 4 эхо-ответа. Длина каждого сообщения составляет 64 байта. В следующей колонке помещены значения времени оборота (RTT), т. е. времени от момента отправки запроса до получения ответа на этот запрос. Как видим, сеть работает достаточно нестабильно — время в последней строке отличается от времени во второй более чем в два раза. На экран выведено также оставшееся время жизни поступивших пакетов.

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

ОТВЕТЫ К ГЛАВЕ 1

1. б.

2. а, в.

3. б, в, г, д, е.

4. д, е, б, в, г, а. Предложение с правильно выбранными словами: «Пользователи (субъекты) получают права доступа к ресурсам (объектам) ИС в результате авторизации, однако прежде им необходимо успешно пройти идентификацию и аутентификацию».

5. б, в, ж.

6. б, в, ж, д, е, з.

7. в, б, д, а, г. Правильное предложение "Есть информация о том, что в Интернете уже появились эксплойты, направленные на использование уязвимостей новой версии браузера. Реализация данной угрозы может привести к атаке, которая нанесет ощутимый ущерб нашему предприятию",

8. б, г, д.

9. д, в, г, б, в, ж, з.

10. б, в.

11. б, в, г.

ОТВЕТЫ К ГЛАВЕ 2

1. б, в, г.
2. а, б, в, г, д
3. а, б, в, г, д, е
4. а, б, в
5. а
6. б, а, в, в. Правильное предложение: “Риск тем выше, чем больше ущерб от атаки и чем выше вероятность атаки."
7. а, в, г, д, е
8. б, в, г
9. е. Процесс имеет циклический характер
10. а, б
11. а, б, г, д, е.
12. б.

ОТВЕТЫ К ГЛАВЕ 3

1. а, д.
2. а, б.
3. а, б, в, д.
4. а.
5. б, в, г, д.
6. б.
7. а, г.
8. а, г, д, е.
9. г, е.
10. а, б, в.

ОТВЕТЫ К ГЛАВЕ 4

1. а, б, г, д.
2. а, б, г, д, ж.
3. а.
4. б, в, г.
5. а, б, г, д.
6. в.
7. б, г.
8. а, б.
9. а.
10. а,б.
И. б, д.
12. в.
13. а,б, в,г.
14. а,б, г,е.
15. а,б, в,г.

ОТВЕТЫ К ГЛАВЕ 5

1. **в.**
2. **а, в, д**
3. **а, в, г, д.**
4. **а, в, г, д, е**
5. **б, в.**
6. **б, в.**
7. **а, б.**
8. **а, б, е.**
9. **а, а, б, б. Правильное предложение: «Сертификат может быть представлен в форме, состоящей из 3 частей: во- первых, открытой части; во-вторых, открытой части, зашифрованной закрытым ключом пользователя,**

в-третьих, части, представляющей собой первые две части, зашифрованные закрытым ключом сертифицирующей организации».

10. в.
11. а, б, д.
12. а, в, г.
13. а, б.
14. б, г.
15. а, б.
16. в.

ОТВЕТЫ К ГЛАВЕ 6

1. а, б, в, г.
2. а.
3. а, б, г.
4. б, г.
5. б.
6. б, в.
7. б.
8. а, б, д.
9. б, в.
10. б, в, г.
11. г.
12. б, г.
13. а, б, в.

ОТВЕТЫ К ГЛАВЕ 7

1. а, б, в, г.
2. а, б, в.
3. а, б, г.
4. б, в, г.
5. а, в, г, д.
6. а, б.
7. а, б, в.
8. а.

ОТВЕТЫ К ГЛАВЕ 8

1. в, г, д, е.
2. а, д.
3. а, б, в.
4. г.
5. а.
6. а, б, в, г.
7. а, б, г, д.
8. а, б, в, г.
9. в, г, д.
10. а, г.

ОТВЕТЫ К ГЛАВЕ 9

1. а, в, г, д.
2. б, в.
3. а, в.
4. а.
5. а, в.
6. а, б, в, г.
7. в, г.
8. б.
9. б.
10. б.
11. а, в.
12. б.
13. а.
14. а, б, в.
15. а, б, г.
16. а, в.
17. г.
18. в.
19. а, б, в, г.
20. в.
21. а.
22. а, б, в, г, д.

ОТВЕТЫ К ГЛАВЕ 10

1. в.
2. а, б, в, г, е, ж.
3. а.
4. б.
5. в, ж, д. Строка должны выглядеть следующим образом:

accesslist 101 permit icmp any 10.12.13.0 0.0.0.255.eq 8

6. в.
7. в.
8. а, б.
9. б, в.
10. в.
11. б.
12. а, б, в, г.
13. е.
14. б, в, г, д.
15. а. б, г, д, ж, з, и.
16. б.
17. в.
18. в.

ОТВЕТЫ К ГЛАВЕ И

1. в.
2. а, б, в.
3. а.
4. в.
5. а, в.
6. а, б, в.
7. б, в.
8. б, в.
9. б.
10. а.

и. б.

ОТВЕТЫ К ГЛАВЕ 12

1. а, б, г.
2. а.
3. б.
4. в.
5. а, б, в.
6. б.
7. в.
8. а, б.

ОТВЕТЫ К ГЛАВЕ 13

1. а, б.
2. б, в.
3. б.
4. б.
5. а, б, в.

ОТВЕТЫ К ГЛАВЕ 14

1. в
2. а, б, в, г.
3. в.
4. в.
5. б, в.
6. а, б, в.

ОТВЕТЫ К ГЛАВЕ 15

1. а, в, г.
2. б.
3. а, в.
4. а, б, г.
5. б.
6. а, б, г.
7. а (так как анализ может показать использование в программе привилегированных команд процессора).
8. б.
9. а.

ОТВЕТЫ К ГЛАВЕ 16

1. г.
2. б.
3. б, в, г.
4. а, б, в, г, е.
5. а, б, в, г, д, е.
6. б, г, а, д. Правильное предложение: «В системе Unix команда su используется для временного получения пользователем статуса суперпользователя root, а команда suro для получения права на выполнение некоторых привилегированных операций».
7. в.
8. а, б, г, д.
9. а, в.
10. а, б, г, д, е.
11. б.

ОТВЕТЫ К ГЛАВЕ 17

1. б.
2. в.
3. б.
4. в.
5. г.
6. г.
7. а.
8. б.

ОТВЕТЫ К ГЛАВЕ 18

1. а, в.
2. в.
3. б.
4. б.
5. в.
6. а, б, в.
7. в.
8. а.
9. а.

ОТВЕТЫ К ГЛАВЕ 19

1. а.
2. а.
3. б.
4. б.
5. а, б.
6. а.
7. а.
8. а.
9. а.
10. в.
11. а.
12. б, в.
13. б.
14. а.
15. 6.

ОТВЕТЫ К ГЛАВЕ 20

1. б, в.
2. а.
3. а, б.
4. в.
5. в, д, б.
6. в.
7. в.
8. в.
9. в.
10. б.

и. б.

12. а, б, в.
13. а, б, в.

ОТВЕТЫ К ГЛАВЕ 21

1. **в.**
2. **в.**
3. б.
4. **а, б, в, г, д, е, ж.**
5. **а, в.**
6. **а, б, в.**
7. **а.**
8. **а.**
9. **а.**
10. **а, б, в.**
11. **а.**
12. б.
13. **и.**
14. **а.**

ОТВЕТЫ К ГЛАВЕ 22

1. б.
2. **а, б, в, г, д, е, ж, з.**
3. **а, в.**
4. **а.**
5. **в.**
6. **а.**
7. **б.**
8. б.
9. **а, б, в.**
10. **б, г.**

Предметный указатель

Литература

Рекомендуемая и использованная литература

1. Олифер В.Г., Олифер Н.А. Сетевые операционные системы. 2-е изд. — СПб.: Питер, 2008.

2. Олифер В.Г., Олифер Н.А. Компьютерные сети. 4-е изд. — СПб.: Питер, 2010.

3. Олифер В.Г., Олифер Н.А. Новые технологии и оборудование IP-сетей. — СПб.: БХВ-Санкт-Петербург, 2000.

4. Олифер В.Г., Олифер Н.А. Основы компьютерных сетей. — СПб.: Питер, 2009.

5. Харрис Ш. CISSP. Руководство для подготовки к экзамену. — McGraw-Hill Companies, 2011.

6. Лапонина О.Р. Основы сетевой безопасности: криптографические алгоритмы и протоколы взаимодействия: курс лекций. — М.: Бином. Лаборатория знаний, 2009.

7. Галатенко В. Стандарты информационной безопасности. — М.: Бином. Лаборатория знаний, 2006.

8. Галатенко В. Основы информационной безопасности. — М.: Бином. Лаборатория знаний, 2008.

9. Джонс К., Шеба М., Джонсон Б. Анти-хакер. Средства защиты компьютерных сетей. — М.: Эком, 2003.

10. Чирилло Дж. Защита от хакеров. — СПб.: Питер, 2002.

11. Сингх С., Книга шифров. Тайная история шифров и их расшифровки. — М.: АСТ, 2007.

12. Saltzer 1, Schroeder M. The protection of information in computer systems // Communications of the ACM. 1974. V. 17, № 7.

13. Parker D. Fighting Computer Crime: A New Framework for Protecting Information. - N.Y.: Wiley, 1998.

14. Peltier T. Information Security Risk Analysis, Third Edition. — CRC Press, 2010.

15. Alberts C, Dorofee A. Managing Information Security Risks: The OCTAVE Approach. — Addison-Wesley Professional, 2002.

16. Howard M., LeBlanc D. Writing Secure Code. — Microsoft Press, 2004.

17. Kabiri P. Privacy, Intrusion Detection and Response. — IGI Global, 2011.

18. Solomon M. Security Strategies in Windows Platforms and Applications. — Jones & Bartlett Learning, 2010.

19. Maximum Linux Security. — Sams, 1999.

20. Gibson D. Microsoft Windows Security Essentials. — Sybex, **2011**.

21. Pfleeger C., Lawrence S. Security in Computing. 4-th Edition. — Prentice Hall, 2006.

22. Noonan W., Dubrawsky I. Firewall Fundamentals. — Cisco Press, 2006.

23. Andress J. The Basics of Information Security. — Syngress, 2011.

24. Cole E. Network Security Bible. 2nd Edition. — John Wiley & Sons, 2009.

25. Kahn D. The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet. — Simon and Schuster, 1996.

26. MaoW. Modern Cryptography: Theory and Practice. — Prentice Hall, 2003.

27. Menezes A., van Oorschot P., Vanstone S. Handbook of Applied Cryptography. Discrete Mathematics and Its Applications. — CRC Press, 2010.

28. Pieprzyk J., Hardjono T. Jennifer Seberry, Fundamentals of Computer Security. — Springer, 2010.

29. VaudenayS. Classical Introduction to Cryptography: Applications for Communications Security. — Springer, 2005.

Стандарты и правовые акты

1. Федеральный закон от 8 июля 2006 г. 149-ФЗ «Об информации, информационных технологиях и о защите информации».

2. Федеральный закон РФ 161-ФЗ «О национальной платежной системе», 2012.

3. Федеральный закон от 19 декабря 2005 г. 160-ФЗ «О ратификации Конвенции Совета Европы о защите физических лиц при автоматизированной обработке персональных данных».

4. Федеральный закон 152-ФЗ «О персональных данных», 2006.

5. Постановление правительства РФ от 21 марта 2012 г. № 211 «Об утверждении перечня мер, направленных на обеспечение выполнения обязанностей, предусмотренных Федеральным законом "О персональных данных”».

6. Приказ ФСТЭК РФ от 11 февраля 2013 г. № 17 «Об утверждении состава и содержания организационных и технических мер по

обеспечению безопасности: персональных данных при их обработке в информационных системах персональных данных».

7. Федеральный закон от 8 апреля 2011 г. № 63-ФЗ «Об электронной подписи».

8. Федеральный закон от 4 мая 2011 г. 99-ФЗ «О лицензировании отдельных видов деятельности».

9. Постановление правительства РФ от 1 сентября 2013 г. «Об утверждении Положения о лицензировании образовательной деятельности».

10. Гостехкомиссия России. Руководящий документ. Защита от несанкционированного доступа к информации. Термины и определения. — М., 1992.

11. Гостехкомиссия России. Руководящий документ. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации. — М., 1992.

12. Гостехкомиссия России. Руководящий документ. Временное положение по организации разработки, изготовления и эксплуатации программных и технических средств защиты информации от НСД в автоматизированных системах и средствах вычислительной техники. — М., 1992.

13. Гостехкомиссия России. Руководящий документ. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от НСД к информации. — М., 1992.

14. Гостехкомиссия России. Руководящий документ. Концепция защиты СВТ и АС от НСД к информации. — М., 1992.

15. ГОСТ 13335-1:2006. Информационная технология. Методы и средства обеспечения безопасности. Часть 1. Концепция и модели менеджмента безопасности информационных и телекоммуникационных технологий.

16. Department of Defence, Trusted Computer System Evaluation Criteria, 1985.

17. Common Criteria for Information Technology Security Evaluation. Part 1: Introduction and general model. Version 2.1 CCIMB-99-031, August, 1999.

18. Common Criteria for Information Technology Security Evaluation. Part 2: Security functional requirements. Version 2.1 CCIMB-99-032, August, 1999.

19. Common Criteria for Information Technology Security Evaluation. Part 3: Security assurance requirements. Version 2.1 CCIMB-99-033, August, 1999.

Рекомендуемые и использованные Интернет-ресурсы

1. www.ietf.org/rfc — сайт стандартов Интернета.
2. www.csrc.nist.gov/ — сайт Национального института стандартов и технологий, США.
3. www.sans.org/reading-room/whitepapers — материалы института SANS (System Administration, Networking, and Security Institute).
4. www.www.fas.org/irp/nsa/ — сайт Национального агенства безопасности, США.
5. www.dorlov.blogspot.ru/
6. www.lukatsky.blogspot.ru/
7. www.bankir.ru/dom/forums/143 — информационная безопасность.
8. www.lockdown.co.uk/?pg= combi — о скорости взламывания паролей.
9. www.schneier.com/essay-246.html
10. www.securelist.com/ru/analysis?pubid= 204007610 — о ботнетах.
11. www.merit.edu/mail.archives/nanog/1997-04/msg00444.html
12. www.riskwatch.com - Risk analysis tools&how they work
13. www.cryptography.com/public/pdf/DPATechInfo.pdf — Paul Ko- cher, Joshua Jaffe, Benjamin Jun Introduction to Differential Power Analysis and Related Attacks, 1998.
14. www.prolexic.com/knowledge-center-dos-and-ddos-attack-re- ports. html — The Q4 2013 Global DDoS Attack Report analyzes the growing threat of mobile DDoS attack apps.
15. www.d.root-servers.org/october21.txt и www.icann.org/announcements/announcement-08mar07.htm — отчеты об атаках корневых DNS-серверов.
16. www.blog.cloudflare.com/the-ddos-that-knocked-spamhaus-off- line-and-hoo — отчет об отраженной DDoS атаке на DNS-серверы.
17. Статьи об инцеденте со Сноуденом:
www.thenation.com/blog/174791/why-are-massive-national-securi- ty-breaches-so-ridiculously-easy#
www.thenewstribune.com/2013/06/14/2638057/indefensible-leaks- and-indefensible.html#storylink= cpy
www.usatoday.com/story/news/politics/2013/06/16/snowden-whistleblower-nsa-officials-roundtable/2428809/
www.networkcomputing.com/quickview/thumb-drive-security-snow- den-l-nsa-0/3641?wc= 4?cid= NWC_report_2013-06-18_html&elq= a3c9c8798d3f477fbf474a4a369267d4&wc= 4

Оглавление

Адрес издательства в Интернет www.techbook.ru

Учебное издание

Редактор Ю. Н. Чернышов
Компьютерная верстка Ю. Н. Чернышова
Обложка художника В. Г. Ситникова

Подписано в печать 2G.04.2014. Формат 60><88/10. Уч. изд. л. 40,25.
Тираж 500 экз. (4-й завод 50 экз.)
ООО «Научно-техническое издательство «Горячая линия Телеком»

Олифер Виктор Григорьевич, Олифер Наталья Алексеевна
Безопасность компьютерных сетей

Профессиональные биографии Виктора и Натальи Олифер очень похожи - они получили свое первое высшее образование в МВТУ имени Н. Э. Баумана (специальность «Электронные вычислительные машины»), а второе - в МГУ им. М. В. Ломоносова (специальность «Прикладная математика»). После защиты диссертации каждый из них совмещал преподавание в вузах с научной работой. В 1995 году Наталья и Виктор стали читать лекции по сетевым технологиям в Центре информационных технологий при МГУ. Ими были разработаны несколько авторских курсов, которые и составили в дальнейшем основу для написания нескольких книг, в том числе таких популярных учебников как «Сетевые операционные системы» и «Компьютерные сети. Принципы, технологии, протоколы». Последняя книга была издана на английском, испанском, португальском и китайском языках.

В настоящее время Наталья Олифер работает независимым консультантом в области сетевых технологий, а Виктор Олифер участвует в проекте по развитию сети JANET, объединяющей университеты и исследовательские центры Великобритании, а также международном проекте GEANT, посвященном разработке и поддержке панъевропейской академической сети.

В. Г. Олифер, Н. А. Олифер

Безопасность компьютерных сетей

Систематизированы обширные теоретические и практические сведения в области методов и способов обеспечения безопасности компьютерных сетей. Рассмотрены меры обеспечения безопасности компьютерных систем как органической части общей информационной системы предприятия; методы защиты программного обеспечения компьютеров и обрабатываемой ими информации; сетевые аспекты передачи информации между узлами компьютерной сети (вопросы безопасности сетевых протоколов и сервисов); базовые технологии, используемые для защиты информации в компьютерной сети, такие как шифрование, аутентификация, авторизация, организация защищенного канала и другие, которые в той или иной мере являются основой всех методов обеспечения безопасности компьютерных сетей.

Книга структурирована в виде учебного курса и отличается широким охватом затронутых тем, при этом авторы стремились сохранить достаточную глубину рассмотрения вопросов, позволяющую понять их суть. Все главы книги завершаются набором вопросов для проверки и самопроверки.

Для широкого круга читателей, которые хотят углубить и систематизировать свои знания в области безопасности компьютерных сетей. Будет особенно полезна специалистам в области информационной безопасности, занимающимся практическими вопросами построения комплексных систем защиты информации, слушателям курсов переподготовки и повышения квалификации, студентам и аспирантам, обучающимся по направлению «Информационная безопасность».

Сайт издательства: