С. А. НЕСТЕРОВ

ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

УЧЕБНОЕ ПОСОБИЕ

Издание третье, стереотипное

ЛАНЬ®
САНКТ-ПЕТЕРБУРГ · МОСКВА · КРАСНОДАР
2017

ББК 32.81я73
Н 56

Нестеров С. А.

Н 56 Основы информационной безопасности: Учебное пособие. — 3-е изд., стер. — СПб.: Издательство «Лань», 2017. — 324 с. — (Учебники для вузов. Специальная литература).

ISBN 978-5-8114-2290-6

Системно излагаются теоретические основы информационной безопасности и описываются практические аспекты, связанные с их реализацией. В пособии рассматриваются теоретические основы защиты информации, основы криптографии, защита информации в IP-сетях, анализ и управление рисками в сфере информационной безопасности. Теоретический материал сопровождается лабораторными работами, выделенными в отдельный раздел.

Пособие может использоваться в системах повышения квалификации в рамках образовательной программы дополнительного профессионального образования «Информатика и вычислительная техника». Также может быть полезно широкому кругу специалистов в области информационных технологий.

ББК 32.81я73

Рецензент
А. А. ЕФРЕМОВ — кандидат физико-математических наук, доцент Санкт-Петербургского государственного политехнического университета.

Книга издана при реализации совместного проекта
издательства «Лань» и издательства
Санкт-Петербургского политехнического университета Петра Великого

Обложка
Е. А. ВЛАСОВА

ОГЛАВЛЕНИЕ

СПИСОК ПРИНЯТЫХ СОКРАЩЕНИИ

АС — автоматизированная система (обработки информации);
БД — база данных;
ИБ — информационная безопасность;
ИС — информационная система;
ИТ — информационные технологии;
ЛК — Лаборатория Касперского;
МЭ — межсетевой экран;
НСД — несанкционированный доступ;
ОО — объект оценки;
ОС — операционная система;
ПО — программное обеспечение;
СЗИ — средство защиты информации;
СМИБ — система менеджмента информационной безопасности;
СФБ — стойкость функции безопасности;
ЦС — центр сертификации;
ЭЦП — электронная цифровая подпись;
ACL (Access Control List) — список управления доступом;
AH (Authentication Header) — протокол аутентифицирующего заголовка;
CA (Certification Authority) — центр сертификации или удостоверяющий центр;
CBC (Cipher Block Chaining) — сцепление блоков шифра (режим работы шифра DES);
CFB (Cipher FeedBack) — обратная связь по шифртексту (режим работы шифра DES);
CRL (Certificate Revocation List) — список отозванных сертификатов;
ECB (Electronic Code Book) — электронная кодовая книга (режим работы шифра DES);
ESP (Encapsulating Security Payload) — протокол инкапсулирующей защиты данных;

ICV (Integrity Check Value) — значение контроля целостности;
MAC (Message Authentication Code) — код аутентификации сообщений, имитовставка;
OFB (Output FeedBack) — обратная связь по выходу (режим работы шифра DES);
PKI (Public Key Infrastructure) — инфраструктура открытых ключей;
SA (Security Association) — контекст защиты или ассоциация безопасности;
SPI (Security Parameter Index) — индекс параметров защиты.

ВВЕДЕНИЕ

Современный специалист в области информационных технологий должен обладать знаниями и навыками обеспечения информационной безопасности. Связано это с тем, что в информационных системах предприятий и организаций хранится и обрабатывается критически важная информация, нарушение конфиденциальности, целостности или доступности которой может привести к нежелательным последствиям. Поэтому вопросам обеспечения информационной безопасности должно уделяться внимание на всех этапах разработки и эксплуатации информационных систем.

В данном пособии изложен материал учебной дисциплины «Основы информационной безопасности», в ходе изучения которой слушатели получают базовые знания о теории защиты информации, методах и средствах обеспечения информационной безопасности, а также практические навыки организации защиты информационных систем. Пособие включает в себя пять разделов.

В разделе 1 «Теоретические основы защиты информации» вводятся базовые понятия, связанные с обеспечением информационной безопасности, рассматриваются основные угрозы безопасности и меры противодействия им. Также делается обзор формальных моделей безопасности и современных стандартов в этой области.

Раздел 2 «Основы криптографии» включает описание основных понятий криптографии. Также изучаются наиболее распространенные алгоритмы симметричного и асимметричного шифрования, формирования дайджестов сообщений с помощью хэш-функций, процесс создания инфраструктуры открытых ключей (PKI).

В разделе 3 «Защита информации в IP-сетях» рассматриваются протоколы криптографической защиты данных, передаваемых по телекоммуникационным сетям, использующим стек протоколов TCP/IP, использование межсетевых экранов для защиты сетей.

В разделе 4 рассматриваются современные методики анализа и управления рисками, связанными с информационной безопасностью.

В разделе 5 приведены описания лабораторных работ.

Пособие может использоваться в системах повышения квалификации в рамках образовательной программы дополнительного профессионального образования «Информатика и вычислительная техника». Также оно может быть полезно широкому кругу специалистов в области информационных технологий.

1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННОИ БЕЗОПАСНОСТИ

1.1. БАЗОВЫЕ ПОНЯТИЯ

Начнем изучение дисциплины с определения ряда базовых понятий.

Информация — это сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления. Информация может существовать в различных формах в виде совокупностей некоторых знаков (символов, сигналов и т. д.) на носителях различных типов. Она может представлять ценность для отдельных лиц или организаций.

Защищаемая информация — информация, являющаяся предметом собственности и подлежащая защите в соответствии с требованиями правовых документов или требованиями, устанавливаемыми собственниками информации. Собственниками информации могут быть государство, юридическое лицо, группа физических лиц, отдельное физическое лицо [1].

В последнее время все большие объемы информации, в том числе и критически важной для отдельных людей, организаций или государств, хранятся, обрабатываются и передаются с использованием автоматизированных систем (АС) обработки информации. *Система обработки информации* — совокупность технических средств и программного обеспечения, а также методов обработки информации и действий персонала, необходимых для выполнения автоматизированной обработки информации [2]. *Объект информатизации* — совокупность информационных ресурсов, средств и систем обработки информации, используемых в соответствии с заданной информационной технологией, а также средств их обеспечения, помещений или объектов (зданий, сооружений, технических средств), в которых эти средства и системы установлены, или помещений и объектов, предназначенных для ведения конфиденциальных переговоров.

В зависимости от конкретных условий может решаться задача обеспечения комплексной безопасности объекта информатизации или защиты отдельных ресурсов — информационных, программных и т. д.

Информационные ресурсы (активы) — отдельные документы и отдельные массивы документов, документы и массивы документов, содержащиеся в информационных системах (библиотеках, архивах, фондах, банках данных, информационных системах других видов).

Рассматривая вопросы безопасности АС, можно говорить о наличии некоторых «желательных» состояний системы, через которые и описывается ее «защищенность» или «безопасность». Безопасность является таким же свойством системы, как надежность или производительность, и в последнее время ей уделяется все большее внимание. Чтобы указать на причины выхода системы из безопасного состояния, вводятся понятия «угроза» и «уязвимость».

Угроза (безопасности информации) — совокупность условий и факторов, создающих потенциальную или реально существующую опасность нарушения безопасности информации.

Источник угрозы безопасности информации — субъект (физическое лицо, материальный объект или физическое явление), являющийся непосредственной причиной возникновения угрозы безопасности информации. По типу источника угрозы делят на связанные и несвязанные с деятельностью человека. Примерами могут служить, соответственно, удаление пользователем файла с важной информацией и пожар в здании. Угрозы, связанные с деятельностью человека, разделяют на угрозы случайного и преднамеренного характера. В последнем случае источник угрозы называют нарушителем или злоумышленником.

Уязвимость (информационной системы) — свойство информационной системы, обуславливающее возможность реализации угроз безопасности обрабатываемой в ней информации. Например, угроза потери информации из-за сбоя в сети электропитания реализуется, ес-

ли в АС не применяются источники бесперебойного питания или средства резервного электроснабжения (это является уязвимостью).

Если говорить об информационных ресурсах, то реализация угрозы может привести к таким последствиям, как получение информации людьми, которым она не предназначена, уничтожение или изменение информации, недоступность ресурсов для пользователей. Таким образом, мы подошли к определению трех основных угроз безопасности.

Угроза конфиденциальности (угроза раскрытия) — это угроза, в результате реализации которой конфиденциальная или секретная информация становится доступной лицу, группе лиц или какой-либо организации, которой она не предназначалась. Здесь надо пояснить разницу между секретной и конфиденциальной информацией. В отечественной литературе «секретной» обычно называют информацию, относящуюся к разряду государственной тайны, а «конфиденциальной» — персональные данные, коммерческую тайну и т. п.

Угроза целостности — угроза, в результате реализации которой информация становится измененной или уничтоженной. Необходимо отметить, что и в нормальном режиме работы АС данные могут изменяться и удаляться. Являются ли эти действия легальными или нет, должно определяться политикой безопасности. *Политика безопасности* — совокупность документированных правил, процедур, практических приемов или руководящих принципов в области безопасности информации, которыми руководствуется организация в своей деятельности.

Угроза отказа в обслуживании (угроза доступности) — угроза, реализация которой приведет к отказу в обслуживании клиентов АС, несанкционированному использованию ресурсов злоумышленниками по своему усмотрению.

Ряд авторов [3] дополняют приведенную классификацию, вводя *угрозу раскрытия параметров* АС, включающей в себя подсистему защиты. Угроза считается реализованной, если злоумышленником в

ходе нелегального исследования системы определены все ее уязвимости. Данную угрозу относят к разряду опосредованных: последствия ее реализации не причиняют какой-либо ущерб обрабатываемой информации, но дают возможность для реализации первичных (непосредственных) угроз.

Таким образом, *безопасность информации* — это состояние защищенности информации, при котором обеспечены ее конфиденциальность, доступность и целостность. *Защита информации* может быть определена как деятельность, направленная на предотвращение утечки защищаемой информации, несанкционированных и непреднамеренных воздействий на защищаемую информацию. Выделяются следующие направления защиты информации:

- *правовая защита информации* — защита информации правовыми методами, включающая в себя разработку законодательных и нормативных правовых документов (актов), регулирующих отношения субъектов по защите информации, применение этих документов (актов), а также надзор и контроль за их исполнением;

- *техническая защита информации* — защита информации, заключающаяся в обеспечении некриптографическими методами безопасности информации (данных), подлежащей (подлежащих) защите в соответствии с действующим законодательством, с применением технических, программных и программно-технических средств;

- *криптографическая защита информации* — защита информации с помощью ее криптографического преобразования[1];

- *физическая защита информации* — защита информации путем применения организационных мероприятий и совокупности средств, создающих препятствия для проникновения или доступа неуполномоченных физических лиц к объекту защиты.

Защита информации осуществляется с использованием способов и средств защиты. *Способ защиты информации* — порядок и правила

[1] Вопросы, связанные с криптографической защитой информации, будут более подробно рассмотрены в разделе 2.

применения определенных принципов и средств защиты информации. *Средство защиты информации* — техническое, программное, программно-техническое средство, вещество и (или) материал, предназначенные или используемые для защиты информации. Отдельно выделяют:

- средства контроля эффективности защиты информации;
- средства физической защиты информации;
- криптографические средства защиты информации.

1.2. ОБЩАЯ СХЕМА ПРОЦЕССА ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ

Рассмотрим теперь взаимосвязь основных субъектов и объектов обеспечения безопасности, как это предлагается в международном стандарте ISO/IEC-15408 (в России он принят как ГОСТ Р ИСО/МЭК 15408-2002 [4]).

Безопасность связана с защитой активов от угроз. Разработчики стандарта отмечают, что следует рассматривать все разновидности угроз, но в сфере безопасности наибольшее внимание уделяется тем из них, которые связаны с действиями человека. Рис. 1.1 иллюстрирует взаимосвязь между высокоуровневыми понятиями безопасности.

За сохранность активов отвечают их владельцы, для которых они имеют ценность. Существующие или предполагаемые нарушители также могут придавать значение этим активам и стремиться использовать их вопреки интересам их владельца. Действия нарушителей приводят к появлению угроз. Как уже отмечалось выше, угрозы реализуются через имеющиеся в системе уязвимости.

Владельцы активов анализируют возможные угрозы, чтобы определить, какие из них могут быть реализованы в отношении рассматриваемой системы. В результате анализа определяются риски (т. е. события или ситуации, которые предполагают возможность ущерба) и проводится их анализ.

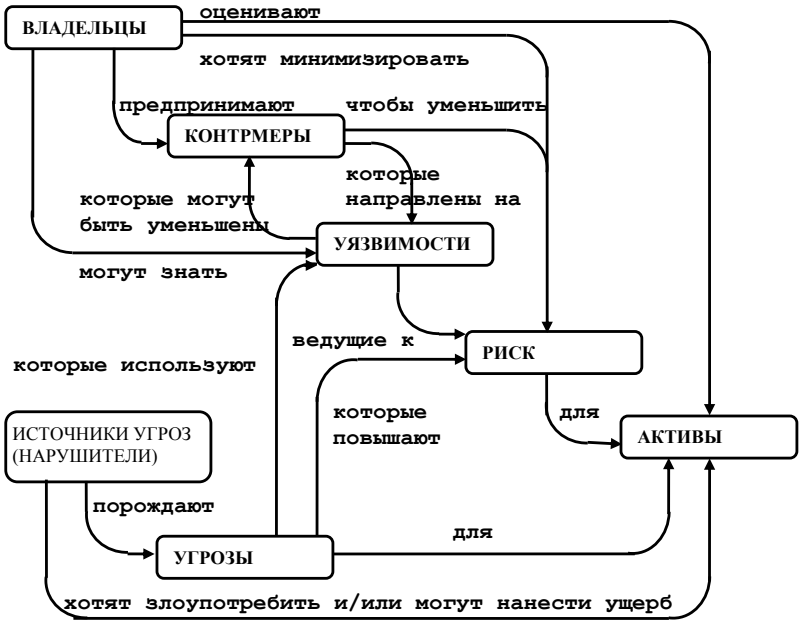


Рис. 1.1. Понятия безопасности и их взаимосвязь

Владельцы актива предпринимают контрмеры для уменьшения уязвимостей и выполнения политики безопасности. Но и после введения этих контрмер могут сохраняться остаточные уязвимости и соответственно — остаточный риск.

1.3. ИДЕНТИФИКАЦИЯ, АУТЕНТИФИКАЦИЯ, УПРАВЛЕНИЕ ДОСТУПОМ. ЗАЩИТА ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА

В этом разделе будут рассмотрены вопросы, связанные с защитой информации от несанкционированного доступа (НСД).

Защита информации от несанкционированного доступа — защита информации, направленная на предотвращение получения защищаемой информации заинтересованными субъектами с нарушением установленных нормативными и правовыми документами (актами)

или обладателями информации прав или правил разграничения доступа к защищаемой информации.

Для защиты от НСД, как правило, используется идентификация, аутентификация и управление доступом. В дополнение к перечисленным, могут применяться и другие методы.

Идентификация — присвоение пользователям идентификаторов (уникальных имен или меток) под которыми система «знает» пользователя. Кроме идентификации пользователей, может проводиться идентификация групп пользователей, ресурсов АС и т. д. Идентификация нужна и для других системных задач, например, для ведения журналов событий. В большинстве случаев идентификация сопровождается аутентификацией. *Аутентификация* — установление подлинности — проверка принадлежности пользователю предъявленного им идентификатора. Например, в начале сеанса работы в АС пользователь вводит имя и пароль. На основании этих данных система проводит идентификацию (по имени пользователя) и аутентификацию (сопоставляя имя пользователя и введенный пароль).

Управление доступом — метод защиты информации путем регулирования использования всех ресурсов системы.

Система идентификации и аутентификации является одним из ключевых элементов инфраструктуры защиты от НСД любой информационной системы. Обычно выделяют 3 группы методов аутентификации.

1. Аутентификация по наличию у пользователя уникального объекта заданного типа. Иногда этот класс методов аутентификации называют по-английски “I have” («у меня есть»). В качестве примера можно привести аутентификацию с помощью смарт-карт или электронных USB-ключей.

2. Аутентификация, основанная на том, что пользователю известна некоторая конфиденциальная информация — “I know” («я знаю»). Например, аутентификация по паролю. Более подробно парольные системы рассматриваются далее в этом разделе.

3. Аутентификация пользователя по его собственным уникальным характеристикам — “I am” («я есть»). Эти методы также называются биометрическими. Биометрические методы аутентификации делят на статические и динамические.

Примеры аутентификации по статическим признакам — это проверка отпечатка пальца, рисунка радужной оболочки глаз, геометрии кисти руки, сравнение с фотографией и т. д. Достоинством этих методов является достаточно высокая точность. Но надо отметить, что подобные методы, как правило, требуют наличия специализированного оборудования (например, специальных сканеров) и имеют ограниченную область применения (например, при аутентификации по отпечатку пальца из-за грязи на руке человек может не пройти аутентификацию, т. е. подобные методы неприменимы на стройках и на многих производствах).

Примеры динамической аутентификации — аутентификация по голосу (при произнесении заранее определенной фразы или произвольного текста), аутентификация по «клавиатурному почерку» (проверяются особенности работы пользователя на клавиатуре, такие как время задержки при нажатии клавиш в различных сочетаниях) и т. д.

Нередко используются комбинированные схемы аутентификации, объединяющие методы разных классов. Например, двухфакторная аутентификация — пользователь предъявляет системе смарт-карту и вводит пин-код для ее активации.

Аутентификация может быть *односторонней*, когда одна сторона аутентифицирует другую (например, сервер проверяет подлинность клиентов), и *двусторонней*, когда стороны проводят взаимную проверку подлинности.

Также аутентификация может быть *непосредственной*, когда в процедуре аутентификации участвуют только две стороны, или *с участием доверенной стороны*. В последнем случае в процессе аутентификации участвуют не только стороны, проверяющие подлинность друг друга, но и другая или другие, вспомогательные. Эту третью

сторону иногда называют сервером аутентификации (англ. «authentication server») или арбитром (англ. «arbitrator»).

Парольные системы аутентификации

Наиболее распространенными на данный момент являются парольные системы аутентификации. Определим ряд понятий, использующихся при описании подобных систем.

Идентификатор пользователя — уникальная информация, позволяющая различить отдельных пользователей парольной системы (провести идентификацию). Это может быть имя учетной записи пользователя в системе или специально генерируемые уникальные числовые идентификаторы.

Пароль пользователя — секретная информация, известная только пользователю (и возможно — системе), которая используется для прохождения аутентификации. В зависимости от реализации системы, пароль может быть одноразовым или многоразовым. При прочих равных условиях системы с одноразовыми паролями являются более надежными. В них исключаются некоторые риски, связанные с перехватом паролей — пароль действителен только на одну сессию и, если легальный пользователь его уже задействовал, нарушитель не сможет такой пароль повторно использовать. Но системы с многоразовыми паролями (в них пароль может быть использован многократно) проще реализовать и дешевле поддерживать, поэтому они более распространены.

Учетная запись пользователя — совокупность идентификатора, пароля и, возможно, дополнительной информации, служащей для описания пользователя. Учетные записи хранятся в базе данных парольной системы.

Парольная система — это программный или программно-аппаратный комплекс, реализующий функции идентификации и аутентификации пользователей компьютерной системы путем проверки паролей. В отдельных случаях подобная система может выполнять дополнительные функции, такие как генерация и распределение

криптографических ключей и т. д. Как правило, парольная система включает в себя интерфейс пользователя, интерфейс администратора, базу учетных записей, модули сопряжения с другими компонентами подсистемы безопасности (подсистемой разграничения доступа, регистрации событий и т. д.).

Рассмотрим некоторые рекомендации по администрированию парольной системы, использующей многоразовые пароли.

1. Задание минимальной длины используемых в системе паролей. Это усложняет атаку путем подбора паролей. Как правило, рекомендуют устанавливать минимальную длину в 6–8 символов.

2. Установка требования использовать в пароле разные группы символов — большие и маленькие буквы, цифры, специальные символы. Это также усложняет подбор.

3. Периодическая проверка администраторами безопасности качества используемых паролей путем имитации атак[1], таких как подбор паролей «по словарю» (т. е. проверка на использование в качестве пароля слов естественного языка и простых комбинаций символов, таких как «1234»).

4. Установление максимального и минимального сроков жизни пароля, использование механизма принудительной смены старых паролей. При внедрении данной меры надо учитывать, что при невысокой квалификации пользователей от администратора потребуются дополнительные усилия по разъяснению пользователям того, что «от них требует система».

5. Ограничение числа неудачных попыток ввода пароля (блокирование учетной записи после заданного числа неудачных попыток войти в систему). Данная мера позволяет защититься от атак путем подбора паролей. Но при необдуманном внедрении также может привести к дополнительным проблемам — легальные пользователи из-за

[1] *Компьютерная атака* — целенаправленное несанкционированное воздействие на информацию, на ресурс автоматизированной информационной системы или получение несанкционированного доступа к ним с применением программных или программно-аппаратных средств.

ошибок ввода паролей по невнимательности могут блокировать свои учетные записи, что потребует от администратора дополнительных усилий.

6. Ведение журнала истории паролей, чтобы пользователи после принудительной смены пароля не могли вновь выбрать себе старый, возможно скомпрометированный пароль.

1.4. МОДЕЛИ БЕЗОПАСНОСТИ

Как уже отмечалось в разделе 1.1, важным этапом процесса обеспечения безопасности АС является разработка политики безопасности. Если отсутствует политика безопасности, невозможно даже четко провести разграничение между санкционированным (легальным) доступом к информации и НСД.

Политика безопасности может быть описана формальным или неформальным образом. Формальное описание политики безопасности производится в рамках модели безопасности. С этой точки зрения, модель безопасности можно определить как абстрактное описание поведения целого класса систем, без рассмотрения конкретных деталей их реализации.

Большинство моделей безопасности оперируют терминами «сущность», «субъект», «объект».

Сущность — любая именованная составляющая защищаемой АС.

Субъект — активная сущность, которая может инициировать запросы ресурсов и использовать их для выполнения каких-либо вычислительных операций. В качестве субъекта может выступать выполняющаяся в системе программа или «пользователь» (не реальный человек, а сущность АС).

Объект — пассивная сущность, используемая для хранения или получения информации. В качестве объекта может рассматриваться, например, файл с данными.

Обычно предполагается, что существует безошибочный способ различения объектов и субъектов.

Доступ — взаимодействие между субъектом и объектом, в результате которого производится перенос информации между ними. Два фундаментальных типа доступа: *чтение* — операция, результатом которой является перенос информации от объекта к субъекту; *запись* — операция, результатом которой является перенос информации от субъекта к объекту.

Также предполагается существование *монитора безопасности объектов*, т. е. такого субъекта, который будет активизироваться при любом обращении к объектам, может различать (на базе определенных правил) легальные и несанкционированные обращения и разрешать только легальный доступ.

В литературе выделяются три основных класса моделей политики безопасности: дискреционные, мандатные и ролевые.

Основу *дискреционной* (избирательной) политики безопасности составляет дискреционное управление доступом, которое характеризуется следующими свойствами [3]:

- все субъекты и объекты должны быть идентифицированы;

- права доступа субъекта к объекту системы определяются на основании некоторого внешнего по отношению к системе правила.

Правила дискреционного управления доступом часто задаются матрицей доступов. В подобной матрице строки соответствуют субъектам системы, столбцы — объектам, элементы матрицы описывают права доступа для соответствующей пары «субъект – объект».

Одной из наиболее известных дискреционных моделей является модель Харрисона–Рузо–Ульмана, часто называемая матричной моделью. Она будет подробно описана ниже.

Этот тип управления доступом наиболее часто используется в операционных системах в связи с относительной простотой реализации. В этом случае правила управления доступом часто описываются через *списки управления доступом* (англ. «Access Control List», сокр. ACL). Список связан с защищаемым объектом и хранит перечень субъектов и их разрешений на данный объект. В качестве примера

можно привести использование ACL для описания прав доступа пользователей и групп к файлу в файловой системе NTFS в операционных системах семейства Windows NT.

Основу *мандатной* политики безопасности составляет мандатное управление доступом, которое подразумевает, что:

- все субъекты и объекты должны быть идентифицированы;
- задан линейно упорядоченный набор меток секретности;
- каждому объекту системы присвоена метка секретности, определяющая ценность содержащейся в нем информации — его *уровень секретности*;
- каждому субъекту системы присвоена метка секретности, определяющая уровень доверия к нему — его *уровень доступа*;
- решение о разрешении доступа субъекта к объекту принимается исходя из типа доступа и сравнения метки субъекта и объекта.

Чаще всего мандатную политику безопасности описывают в терминах модели Белла–ЛаПадула, которая будет рассмотрена ниже в данном разделе.

Управление доступом, основанное на ролях, оперирует в терминах «роль», «пользователь», «операция». Вся информация рассматривается как принадлежащая организации (а не пользователю, ее создавшему). Решения о разрешении или отказе в доступе принимаются на основе информации о той функции (роли), которую пользователь выполняет в организации. Роль можно понимать как множество действий, которые разрешены пользователю для выполнения его должностных обязанностей. Администратор описывает роли и авторизует пользователей на выполнение данной роли. Таким образом, ролевые модели содержат как признаки мандатных, так и признаки избирательных моделей.

1.4.1. Модель Харрисона–Рузо–Ульмана

Модель Харрисона–Рузо–Ульмана (матричная модель) используется для анализа системы защиты, реализующей дискреционную политику безопасности. При этом система представляется конечным

автоматом, функционирующим согласно определенным правилам перехода.

При использовании матричной модели доступа должны быть определены множества субъектов **S**, объектов **O** и прав доступа **R**. В качестве субъектов системы рассматриваются в первую очередь выполняющиеся программы, поэтому предполагается, что $\mathbf{S} \subset \mathbf{O}$. Условия доступа субъекта $s \in \mathbf{S}$ к объекту $o \in \mathbf{O}$ определяются матрицей доступа. Пусть, например, множество прав доступа состоит из прав на чтение (r), запись (w), выполнение (e). Запрет будет соответствовать пустому множеству прав доступа (∅). Тогда матрица доступа может быть такой, как представлено в табл. 1.1.

Таблица 1.1

Пример матрицы доступа

	o_1	o_2	o_3	o_4
s_1	rwe	∅	rw	rw
s_2	e	rwe	r	∅

Здесь мы предполагаем, что объекты o_1, o_2 — это исполняемые файлы, которые после запуска становятся субъектами s_1 и s_2.

Могут определяться и другие наборы прав, например, {чтение, запись, владение}.

При описании систем с большим числом объектов и субъектов размерность матрицы доступа может получиться весьма значительной. Для ее снижения одинаковые по имеющимся правам субъекты и сходные по значимости объекты можно организовать в группы и давать разрешения группе субъектов на группу объектов.

Функционирование системы рассматривается с точки зрения изменений в матрице доступа. Модель определяет 6 примитивных операций: «создать»/«уничтожить» объект и субъект, «внести»/«удалить» право доступа субъекта к объекту. Их описание приведено в табл. 1.2.

Таблица 1.2

Элементарные операции модели Харрисона–Рузо–Ульмана

Операция	Результат операции
«создать» субъект s’, где s’$\notin$**S**	**S’**=**S**$\cup${s’}; **O’**=**O**$\cup${s’}; **M’**[s,o]= **M**[s,o] для всех s$\in$**S,** o$\in$**O**; **M’**[s’,o]= $\varnothing$ для всех o$\in$**O’**, **M’**[s,s’]= $\varnothing$ для всех s$\in$**S’**
«создать» объект o’, где o’$\notin$**O**	**S’**=**S**; **O’**=**O**$\cup${o’}; **M’**[s,o]= **M**[s,o] для всех s$\in$**S,** o$\in$**O**; **M’**[s,o’]= $\varnothing$ для всех s$\in$**S’**
«уничтожить» субъект s’, где s’$\in$**S**	**S’**=**S**\{s’}; **O’**=**O**\{s’}; **M’**[s,o]= **M**[s,o] для всех s$\in$**S’,** o$\in$**O’**;
«уничтожить» объект o’, где o’$\in$**O**	**S’**=**S**; **O’**=**O**\{o’}; **M’**[s,o]= **M**[s,o] для всех s$\in$**S’,** o$\in$**O’**;
«внести» право r’$\in$**R** в **M**[s’,o’], где s’$\in$**S,** o’$\in$**O**	**S’**=**S**; **O’**=**O**; **M’**[s,o]=**M**[s,o] для s$\neq$s’,s$\in$**S’,** o$\neq$o’, o$\in$**O’**; **M’**[s’,o’]=**M**[s’,o’] $\cup${r’}
«удалить» право r’$\in$**R** из **M**[s’,o’], где s’$\in$**S,** o’$\in$**O**	**S’**=**S**; **O’**=**O**; **M’**[s,o]=**M**[s,o] для s$\neq$s’,s$\in$**S’,** o$\neq$o’, o$\in$**O’**; **M’**[s’,o’]=**M**[s’,o’]\ {r’}

Начальное состояние системы описывается множеством прав доступа **R**, множеством субъектов **S**, множеством объектов **O** (**S**$\subseteq$**O**, мощности указанных множеств |**S**| = i, |**O**| = j, i ≤ j), матрицей доступа $\mathbf{M}_{i\times j}$ (элемент матрицы, соответствующий субъекту s и объекту o обозначается **M**[s,o] и является подмножеством множества прав доступа). Конечное состояние (после выполнения операции) — **S’**, **O’**, **M’**, **R** (множество прав доступа не изменяется).

Из примитивных операторов могут составляться команды. Команда состоит из двух частей: условия, при котором она выполняется, и последовательности операторов.

Общий вид команды [3]:

```
command C (x1, …, xk):
if r1∈ M[xs1,xo1] and … and rm∈ M[xsm,xom] then
α1;
…
αn;
end,
```

где r_1, …, $r_m \in$ **R** — права доступа, α_1, …, α_n — последовательность примитивных операторов. При выполнении команды система переходит из состояния **Q** в новое состояние **Q'**. При этом, если хотя бы одно из условий команды не выполнено, **Q = Q'**. Для примера рассмотрим команду создания субъектом s файла f. Множество прав доступа — чтение (read), запись (write), владение (own). Считаем, что для создания файла не требуется выполнения каких-либо дополнительных условий.

```
command «создать файл» (s, f)
    «создать» объект f;
    «внести» право владения own в M[s,f];
    «внести» право на чтение read в M[s,f];
    «внести» право на запись write в M[s,f];
end.
```

Как показали результаты анализа данной модели безопасности, задача построения алгоритма проверки безопасности систем, реализующих дискреционную политику безопасности, не может быть решена в общем случае.

Введем ряд определений.

Будем считать, что возможна *утечка права* **r** ∈ **R** в результате выполнения команды **c**, если при переходе системы в конечное состояние **Q'** выполняется примитивный оператор, вносящий **r** в элемент матрицы доступов **M**, до этого **r** не содержавший.

Начальное состояние $\mathbf{Q_0}$ называется *безопасным по отношению к некоторому праву* **r**, если невозможен переход системы в такое состояние **Q**, в котором может возникнуть утечка права **r.**

Система называется *монооперационной*, если каждая команда содержит только один примитивный оператор.

Для модели Харрисона–Рузо–Ульмана были доказаны следующие утверждения:

1. Существует алгоритм, который проверяет, является ли исходное состояние монооперационной системы безопасным для данного права **r**.

2. Задача проверки безопасности произвольных систем алгоритмически неразрешима.

Таким образом, с одной стороны, общая модель Харрисона–Рузо–Ульмана может выражать большое разнообразие политик дискреционного доступа, но при этом не существует алгоритма проверки их безопасности. С другой стороны, можно предпочесть монооперационную систему, для которой алгоритм проверки безопасности существует, но данный класс систем является слишком узким. Например, монооперационные системы не могут выразить политику, дающую субъектам права на созданные ими объекты, т. к. не существует одной операции, которая и создает объект, и одновременно помечает его как принадлежащий создающему субъекту.

1.4.2. Модель Белла–ЛаПадула

Классической мандатной моделью безопасности является модель Белла–ЛаПадула. В ней для описания системы используются:

S — множество субъектов (например, множество пользователей и программ);

O — множество объектов (например, множество файлов);

L — линейно упорядоченное множество уровней безопасности (например, «общий доступ», «для служебного пользования», «секретно», «совершенно секретно»);

F: $\mathbf{S} \cup \mathbf{O} \rightarrow \mathbf{L}$ — функция, определяющая уровень безопасности субъекта или объекта в данном состоянии;

V — множество состояний — множество упорядоченных пар (F, **M**), где **M** — матрица доступа субъектов к объектам (матрица,

строки которой соответствуют субъектам системы, столбцы — объектам, элемент матрицы M_{so}, далее обозначаемый как M[s, o], описывает права на доступ субъекта s к объекту o).

Система описывается начальным состоянием $v_0 \in \mathbf{V}$, множеством запросов **R** и функцией переходов T: $(\mathbf{V} \times \mathbf{R}) \rightarrow \mathbf{V}$, описывающей переход системы из состояния в состояние под действием запроса.

В модели Белла–ЛаПадула вводится определение двух свойств безопасности системы: безопасность по чтению и безопасность по записи.

Состояние (F, **M**) *безопасно по чтению* тогда и только тогда, когда для $\forall$ s $\in$ **S**, $\forall$ o $\in$ **O** выполняется требование:

$$\text{чтение} \in M[s, o] \Rightarrow F(s) \geq F(o),$$

т. е. субъект s может прочитать информацию из объекта o, только если уровень секретности *o* меньше или равен уровню доступа s. Данное свойство безопасности также называется правилом запрета чтения с верхнего уровня.

Состояние (F, **M**) *безопасно по записи* тогда и только тогда, когда для $\forall$ s $\in$ **S**, $\forall$ o $\in$ **O** выполняется требование:

$$\text{запись} \in M[s, o] \Rightarrow F(o) \geq F(s),$$

т. е. субъект s может записать информацию в объект o, только если уровень секретности *o* выше или равен уровню доступа s. Данное свойство безопасности также называется правилом запрета записи на нижний уровень.

Состояние системы $v \in \mathbf{V}$ *безопасно* тогда и только тогда, когда оно безопасно и по чтению, и по записи.

Система (v_0, **R**, T) *безопасна* тогда и только тогда, когда ее начальное состояние v_0 безопасно и любое состояние, достижимое из v_0 после выполнения конечной последовательности запросов из **R**, также безопасно.

Большим достоинством модели Белла–ЛаПадула является то, что для нее доказана основная теорема безопасности. В общем случае, данная теорема формулируется следующим образом: если начальное

состояние системы безопасно, и все переходы из состояния в состояние не нарушают ограничений, сформулированных политикой безопасности, то любое состояние системы, достижимое за конечное число переходов будет безопасным. В случае модели Белла–ЛаПадула ограничения не позволяют нарушить безопасность по чтению и записи.

Основная теорема безопасности для модели Белла–ЛаПадула

Система (v_0, **R**, T) (т. е. система с начальным состоянием v_0, множеством запросов **R**, функцией переходов T) безопасна тогда и только тогда, когда состояние v_0 безопасно, и функция переходов T такова, что для $\forall\ v \in \mathbf{V}$, достижимого из состояния v_0 после выполнения конечной последовательности запросов из **R** (таких что $T(v, r) = v^*$, где $v = (F, \mathbf{M})$ — исходное состояние, $v^* = (F^*, \mathbf{M}^*)$ — состояние после перехода), для $\forall\ s \in \mathbf{S}$, $\forall\ o \in \mathbf{O}$ выполняются следующие условия:

- если чтение$\in M^*[s,o]$ и чтение$\notin M[s,o]$, то $F^*(s) \geq F^*(o)$;
- если чтение$\in M[s,o]$ и $F^*(s) < F^*(o)$, то чтение$\notin M^*[s, o]$;
- если запись $\in M^*[s, o]$ и запись $\notin M[s, o]$, то $F^*(o) \geq F^*(s)$;
- если запись $\in M[s, o]$ и $F^*(o) < F^*(s)$, то запись $\notin M^*[s, o]$.

Кратко рассмотрим доказательство теоремы.

Необходимость. Если система безопасна, то начальное состояние v_0 безопасно по определению. Пусть существует некоторое состояние v^*, достижимое из v_0 путем выполнения конечного числа запросов из **R** и полученное в результате перехода из безопасного состояния v: $T(v, r) = v^*$. Тогда, если при таком переходе нарушено хотя бы одно из первых двух ограничений, накладываемых теоремой на функцию T, то состояние v^* не будет безопасным по чтению. Если функция T нарушает одно из двух последних условий теоремы, то состояние v^* не будет безопасным по записи. Таким образом, при нарушении условий теоремы система становится небезопасной. Необходимость доказана.

Достаточность. Используем метод доказательства от противного. Пусть система небезопасна. В этом случае либо начальное состояние v_0 небезопасно, что противоречит условиям теоремы, либо должно существовать небезопасное состояние v^*, достижимое из безопасного начального состояния v_0 путем выполнения конечного числа запросов из **R**. В этом случае обязательно будет иметь место переход $T(v, r) = v^*$, при котором состояние v — безопасно, а v^* — нет. Однако четыре условия теоремы делают такой переход невозможным.

Несмотря на достоинства модели Белла–ЛаПадула, при ее строгой реализации в реальных АС возникает ряд проблем.

1. *Завышение уровня секретности*, связанное с одноуровневой природой объектов и правилом безопасности по записи. Если субъект с высоким уровнем доступа хочет записать что-то в объект с низким уровнем секретности, то сначала приходится повысить уровень секретности объекта, а потом осуществлять запись. Таким образом, даже один параграф, добавленный в большой документ субъектом с высоким уровнем доступа, повышает уровень секретности всего этого документа. Если по ходу работы изменения в документ вносят субъекты со все более высоким уровнем доступа, уровень секретности документа также постоянно растет.

2. *Запись вслепую*. Эта проблема возникает, когда субъект производит операцию записи в объект с более высоким уровнем безопасности, чем его собственный. В этом случае после завершения операции записи субъект не сможет проверить правильность выполнения записи при помощи контрольного чтения, так как ему это запрещено в соответствии с правилом безопасности по чтению.

3. *Проблема удаленного чтения-записи*. В распределенных системах при удаленном чтении файла создаются два потока: от субъекта к объекту (запросы на чтение, подтверждения, прочая служебная информация) и от объекта к субъекту (сами запрашиваемые данные). При этом, например, если $F(s) > F(o)$, то первый поток будет противоречить свойству безопасности по записи. На практике для решения

этой проблемы надо разделять служебные потоки (запросы, подтверждения) и собственно передачу информации.

4. *Доверенные субъекты*. Модель Белла–ЛаПадула не учитывает, что в реальной системе, как правило, существуют субъекты, действующие в интересах администратора, а также системные процессы, например, драйверы. Жесткое соблюдение правил запрета чтения с верхнего уровня и запрета записи на нижний уровень в ряде случаев делает невозможной работу подобных процессов. Соответственно, их также приходится выделять.

1.4.3. Ролевая модель безопасности

Ролевая модель безопасности появилась как результат развития дискреционной модели. Однако она обладает новыми по отношению к исходной модели свойствами: управление доступом в ней осуществляется как на основе определения прав доступа для ролей, так и путем сопоставления ролей пользователям и установки правил, регламентирующих использование ролей во время сеансов.

В ролевой модели понятие «субъект» замещается понятиями «пользователь» и «роль» [5]. Пользователь — человек, работающий с системой и выполняющий определенные служебные обязанности. Роль — это активно действующая в системе абстрактная сущность, с которой связан набор полномочий, необходимых для выполнения определенной деятельности. Подобное разделение хорошо отражает особенности деятельности различных организаций, что привело к распространению ролевых политик безопасности. При этом как один пользователь может быть авторизован администратором на выполнение одной или нескольких ролей, так и одна роль может быть сопоставлена одному или нескольким пользователям.

При использовании ролевой политики управление доступом осуществляется в две стадии:

- для каждой роли указывается набор полномочий (разрешений на доступ к различным объектам системы);

- каждому пользователю сопоставляется список доступных ему ролей.

При определении ролевой политики безопасности используются следующие множества:

U — множество пользователей;

R — множество ролей;

P — множество полномочий (разрешений) на доступ к объектам системы;

S — множество сеансов работы пользователя с системой.

Как уже отмечалось выше, ролям сопоставляются полномочия, а пользователям — роли. Это задается путем определения следующих множеств:

$PA \subseteq P \times R$ — определяет множество полномочий, установленных ролям (для наглядности условно может быть представлено в виде матрицы доступа);

$UA \subseteq U \times R$ — устанавливает соответствие между пользователями и доступными им ролями.

Рассмотрим процесс определения прав доступа для пользователя, открывшего сеанс работы с системой (в рамках одного сеанса работает только один пользователь). Правила управления доступом задаются с помощью следующих функций:

$user: S \to U$ — для каждого сеанса $s \in S$ эта функция определяет пользователя, который осуществляет этот сеанс работы с системой: $user(s) = u \mid u \in U$;

roles — для каждого сеанса $s \in S$ данная функция определяет подмножество ролей, которые могут быть одновременно доступны пользователю в ходе этого сеанса: $roles(s) = \{r_i \mid (user(s), r_i) \in UA\}$;

$permissions: S \to P$ — для каждого сеанса эта функция задает набор доступных в нем полномочий, который определяется путем объединения полномочий всех ролей, задействованных в этом сеансе: $permissions(s) = \bigcup_{r \in roles(s)} \{p_i \mid (p_i, r) \in PA\}$.

В качестве критерия безопасности ролевой модели используется следующее правило: система считается безопасной, если любой пользователь системы, работающий в сеансе $s \in S$, может осуществлять действия, требующие полномочия $p \in P$, только в том случае, если $p \in permissions(s)$.

Существует несколько разновидностей ролевых моделей управления доступом, различающихся видом функций $user$, $roles$ и $permissions$, а также ограничениями, накладываемыми на множества PA и UA.

В частности, может определяться иерархическая организация ролей, при которой роли организуются в иерархии, и каждая роль наследует полномочия всех подчиненных ей ролей.

Могут быть определены взаимоисключающие роли (т. е. такие роли, которые не могут быть одновременно назначены одному пользователю). Также может вводиться ограничение на одновременное использование ролей в рамках одной сессии, количественные ограничения при назначении ролей и полномочий, может производиться группировка ролей и полномочий.

1.5. ПРОЦЕСС ПОСТРОЕНИЯ И ОЦЕНКИ СИСТЕМЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ. СТАНДАРТ ISO/IEC 15408

Одним из наиболее распространенных современных стандартов в области информационной безопасности является международный стандарт ISO/IEC 15408. Он был разработан на основе стандарта «Общие критерии безопасности информационных технологий» вер. 2.1. В 2002 году этот стандарт был принят в России как ГОСТ Р ИСО/МЭК 15408-2002 «Информационная технология. Методы обеспечения безопасности. Критерии оценки безопасности информационных технологий» [4], часто называемый в литературе «Общие критерии».

Стандарт разработан таким образом, чтобы удовлетворить потребности трех групп специалистов: разработчиков, экспертов по сер-

тификации и пользователей объекта оценки. Под объектом оценки (ОО) в стандарте понимаются «подлежащие оценке продукт информационных технологий (ИТ) или система с руководствами администратора и пользователя». К таким объектам относятся, например, операционные системы, прикладные программы, информационные системы и т. д. Ранее, в разделе 1.2 пособия рассматривалась определяемая стандартом взаимосвязь высокоуровневых понятий в области информационной безопасности (рис. 1.1).

«Общие критерии» предусматривают наличие двух типов требований безопасности — функциональных и доверия. Функциональные требования относятся к сервисам безопасности, таким как управление доступом, аудит и т. д. Требования доверия к безопасности относятся к технологии разработки, тестированию, анализу уязвимостей, поставке, сопровождению, эксплуатационной документации и т. д.

Описание обоих типов требований выполнено в едином стиле: они организованы в иерархию «класс – семейство – компонент – элемент». Термин «класс» используется для наиболее общей группировки требований безопасности, а элемент — самый нижний, неделимый уровень требований безопасности. В стандарте выделены 11 классов функциональных требований:

- аудит безопасности;
- связь (передача данных);
- криптографическая поддержка (криптографическая защита);
- защита данных пользователя;
- идентификация и аутентификация;
- управление безопасностью;
- приватность (конфиденциальность);
- защита функций безопасности объекта;
- использование ресурсов;
- доступ к объекту оценки;
- доверенный маршрут/канал.

Рис. 1.2. Структура профиля защиты

Основные структуры, определяемые «Общими критериями» — это профиль защиты и задание по безопасности. Профиль защиты — это независимая от реализации совокупность требований безопасно-

сти для некоторой категории ОО, отвечающая специфическим запросам потребителя. Профиль состоит из компонентов или пакетов функциональных требований и одного из уровней гарантированности. Структура профиля защиты представлена на рис. 1.2.

Профиль определяет «модель» системы безопасности или отдельного ее модуля. Количество профилей потенциально не ограничено, они разрабатываются для разных областей применения (например, профиль «Специализированные средства защиты от несанкционированного доступа к конфиденциальной информации»).

Профиль защиты служит основой для создания задания по безопасности, которое можно рассматривать как технический проект для разработки ОО. Задание по безопасности может включать требования одного или нескольких профилей защиты. Оно описывает также уровень функциональных возможностей средств и механизмов защиты, реализованных в ОО, и приводит обоснование степени их адекватности.

По результатам проводимых оценок, создаются каталоги сертифицированных профилей защиты и продуктов (операционных систем, средств защиты информации и т. д.), которые затем используются при оценке других объектов.

2. ОСНОВЫ КРИПТОГРАФИИ

2.1. ОСНОВНЫЕ ПОНЯТИЯ. КЛАССИФИКАЦИЯ ШИФРОВ

Исторически *криптография* (в переводе с греческого — «тайнопись») зародилась как способ скрытой передачи сообщений без сокрытия самого факта их передачи [6]. Для этой цели сообщение, написанное с использованием какого-либо общепринятого языка, преобразовывалось под управлением дополнительной информации, называемой *ключом*. Результат преобразования, называемый *криптограммой*, содержит исходную информацию в полном объеме, однако последовательность знаков в нем внешне представляется случайной и не позволяет восстановить исходную информацию без знания ключа.

Процедура преобразования называется *шифрованием*, обратного преобразования — *расшифровыванием*.

Сейчас *криптографией* принято называть науку о математических методах обеспечения конфиденциальности и аутентичности (целостности и подлинности) информации. Задачей исследования методов преодоления криптографической защиты занимается *криптоанализ*. Для обозначения совокупности криптографии и криптоанализа используется термин «*криптология*».

Несмотря на то, что шифры применялись еще до нашей эры, как научное направление современная криптография относительно молода. Одной из важнейших работ в данной области является статья Клода Шеннона (Claude Shannon) «Теория связи в секретных системах», опубликованная в открытой печати в 1949 году. На рис. 2.1 изображена предложенная Шенноном схема секретной системы [7].

На стороне отправителя имеются два источника информации — источник сообщений и источник ключей. Источник ключей выбирает из множества всех возможных ключей один ключ K, который будет использоваться в этот раз. Ключ передается отправителю и получателю сообщения таким образом, что его невозможно перехватить.

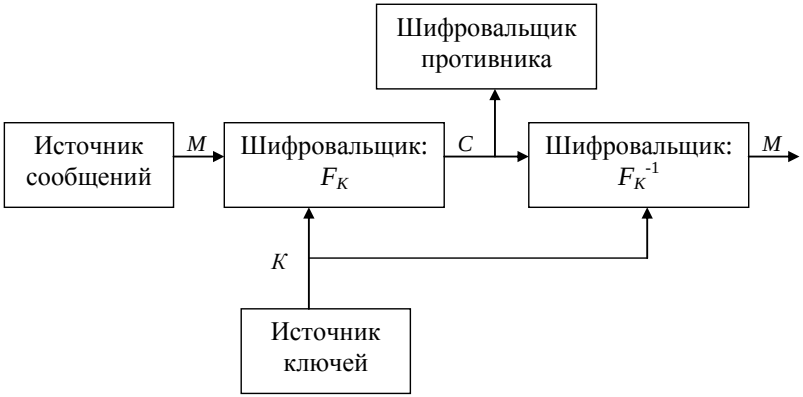


Рис. 2.1. Схема секретной системы

Отображение F_K, примененное шифровальщиком к сообщению M, дает криптограмму C:

$$C = F_K M . \qquad (2.1)$$

В связи с тем, что получатель должен иметь возможность восстановить сообщение M из криптограммы C при известном ключе K, отображение F_K должно иметь единственное обратное отображение F_K^{-1}, такое что:

$$M = F_K^{-1} C . \qquad (2.2)$$

Секретная система (или в современной терминологии — *шифр*) определяется как семейство однозначно обратимых отображений множества возможных сообщений во множество криптограмм. Выбор ключа K определяет, какой именно элемент F_K будет использоваться. Предполагается, что противнику известна используемая система, т. е. семейство отображений $\{F_i \mid i=1..N\}$ и вероятности выбора различных ключей. Однако он не знает, какой именно ключ выбран, и остальные возможные ключи столь же важны для него, как и истинный.

Процесс расшифровывания сообщения для легального получателя информации состоит в применении криптографического отображения, обратного по отношению к отображению, использованному при шифровании.

Процесс расшифровки для противника представляет собой попытку определить сообщение (или конкретный ключ), имея в распоряжении только криптограмму и априорные вероятности различных ключей и сообщений.

Существуют шифры, для которых любой объем перехваченной информации недостаточен для того, чтобы найти шифрующее отображение. Шифры такого типа называются *безусловно стойкими*. Иными словами, безусловно стойкими являются такие шифры, для которых криптоаналитик (даже если он обладает бесконечными вычислительными ресурсами) не может улучшить оценку исходного со-

общения M на основе знания криптограммы C по сравнению с оценкой при неизвестной криптограмме.

Шифры другого типа характеризуются тем, что при определенном объеме перехваченных данных определить ключ (или расшифровать сообщение без знания ключа) становится теоретически возможно. Минимальный объем криптограммы, для которого существует единственное решение криптоаналитической задачи, называется *интервалом единственности*. Однако для криптоаналитика, обладающего ограниченными вычислительными ресурсами, вероятность найти это решение за время, в течение которого информация представляет ценность, чрезвычайно мала. Шифры такого типа называются *условно стойкими*. Их стойкость основана на высокой вычислительной сложности «взлома» шифра. Большинство применяемых сейчас шифров относятся к этому типу.

Доказано, что безусловно стойкие шифры существуют. Но для их построения необходимо использовать равновероятный случайный ключ, имеющий длину, равную длине сообщения. При соблюдении этого условия сама процедура преобразования может быть достаточно простой.

Рассмотрим следующий пример. Пусть нужно передать сообщение M, представленное в двоичной кодировке. Вероятность того, что очередной символ сообщения будет 1, равна q, 0 — $(1 - q)$. Криптограмма получается путем побитного сложения по модулю 2 (т. е. сложения без переноса старшего разряда) сообщения с бесконечным, случайным, равномерно распределенным ключом K:

$$C = M \oplus K. \qquad (2.3)$$

Подобное преобразование также называют *гаммированием*, а ключ K — *ключевой гаммой*. Найдем вероятность того, что очередной символ криптограммы будет равен 1. Это произойдет, если в исходном сообщении соответствующий символ равен 0, а в ключе — 1 или в сообщении — 1, в ключе — 0. Эти пары событий взаимоисключающие, так что следует применить формулу сложения вероятностей:

$$p(C=1)=(1-q)\times 0{,}5+q\times 0{,}5=0{,}5. \qquad (2.4)$$

Таким образом, вероятность появления в криптограмме единицы не зависит от статистических свойств исходного сообщения. И анализируя криптограмму, нарушитель не сможет получить дополнительной информации об исходном сообщении. Надо отметить, что подобными свойствами обладает только случайный бесконечный равномерно распределенный ключ. Если вероятность появления в ключе единицы отлична от 0,5, то q в формуле (2.4) не удастся исключить из результата.

Рассмотрим, какими же свойствами должен обладать хороший шифр. Во-первых, шифрование и расшифровывание должно осуществляться достаточно быстро в тех условиях, в которых применяется шифр (с использованием ЭВМ, при шифровании вручную и т. п.). Во-вторых, шифр должен надежно защищать сообщение, т. е. быть стойким к раскрытию.

Криптостойкость — стойкость шифра к раскрытию методами криптоанализа. Она определяется вычислительной сложностью алгоритмов, применяемых для атаки на шифр. Вычислительная сложность измеряется временной и емкостной сложностями [8].

Для определения сложности алгоритма с конкретной задачей связывается число, называемое *размером задачи*, которое характеризует количество входных данных. Например, для задачи умножения чисел размером может быть длина наибольшего из сомножителей.

Временная сложность (или просто сложность) — это время, затрачиваемое алгоритмом для решения задачи, рассматриваемое как функция от размера задачи. Нередко сложность измеряют количеством некоторых элементарных операций. *Емкостная сложность* — объем памяти, необходимой для хранения полученных в ходе работы данных, как функция от размера задачи.

Очень важное требование к стойкому шифру было сформулировано в XIX веке голландским криптографом Огюстом Керкгоффсом (Auguste Kerckhoffs). В соответствии с ним, при оценке надежности

шифрования необходимо предполагать, что противник знает все об используемой системе шифрования, кроме применяемых ключей. Данное правило отражает важный принцип организации защиты информации: защищенность системы не должна зависеть от секретности долговременных элементов (т. е. таких элементов, которые невозможно было бы быстро изменить в случае утечки секретной информации).

Существует несколько обобщенных постановок задачи криптоанализа. Все они формулируются в предположении, что криптоаналитику известны применяемый алгоритм шифрования и полученные криптограммы. Могут рассматриваться:

- атака при наличии только известной криптограммы;

- атака при наличии известного фрагмента открытого текста. В этом случае, криптоаналитик имеет доступ к криптограммам, а также к соответствующим некоторым из них исходным сообщениям. Задача — определить использующийся при шифровании ключ или расшифровать все остальные сообщения. Разновидность данного класса атак — атака с возможностью выбора открытого текста (когда криптоаналитик может навязать текст для шифрования и получить соответствующую ему криптограмму);

- атаки, использующие особенности реализации аппаратных шифраторов. В частности, может анализироваться тепловое и электромагнитное излучение от устройств, распространение ошибок после однократного воздействия на аппаратуру (по цепи электропитания или иным образом) и т. д.;

- атака методом полного перебора множества возможных ключей. Данная атака также называется «атака методом грубой силы» (от англ. «brute force»).

Виды шифров

Рассмотрим классификации шифров по разным признакам. По типу преобразований шифры можно разделить на следующие группы:

- шифры замены (подстановки);

- шифры перестановки;
- шифры гаммирования;
- шифры на основе аналитических преобразований.

При этом надо учитывать, что некоторые современные шифры совместно используют преобразования различных типов.

Шифры замены (подстановки): преобразование заключается в том, что символы шифруемого текста заменяются символами того или иного алфавита (алфавита криптограммы) в соответствии с заранее обусловленной схемой замены.

Подстановки разделяются на *одноалфавитные* и *многоалфавитные*. В первом случае определенному символу алфавита исходного сообщения всегда ставится в соответствие один и тот же символ алфавита криптограммы. Один из наиболее известных шифров данного класса — шифр Цезаря. В нем каждая буква алфавита заменялась на следующую через одну после нее. В случае русского алфавита, «а» меняется на «в», «б» на «г» и т. д. Алфавит «замыкался», поэтому «я» надо было заменять на «б». В качестве ключа в данном случае выступает число, на которое надо «сдвигать» символ алфавита, в нашем примере — 2. К достоинству таких шифров относится простота преобразования. Но они легко взламываются путем сравнения частоты появления различных символов в естественном языке и криптограмме.

При использовании многоалфавитных подстановок учитываются дополнительные параметры (например, положение преобразуемого символа в тексте), и в зависимости от них символ исходного алфавита может заменяться на один из нескольких символов алфавита шифртекста. Например, нечетные символы сообщения заменяются по одному правилу, четные — по другому.

Шифры перестановок: шифрование заключается в том, что символы исходного текста переставляются по определенному правилу в пределах блока этого текста. При достаточной длине блока и

сложном, неповторяющемся порядке перестановки можно достичь приемлемой стойкости шифра.

Шифрование *гаммированием* заключается в том, что символы шифруемого текста складываются с символами некоторой случайной последовательности, называемой гаммой шифра или ключевой гаммой. Стойкость шифрования определяется длиной (периодом) неповторяющейся части гаммы шифра, а также сложностью предугадывания следующих элементов гаммы по предыдущим.

Шифрование *аналитическими преобразованиями* подразумевает использование аналитического правила (формулы) по которому преобразуется текст.

По типу использования ключей шифры делятся на:

- *симметричные*, использующие для шифрования и расшифровывания информации один и тот же ключ;

- *асимметричные,* использующие для шифрования и расшифровывания два различных ключа. Данный тип шифров будет подробно рассматриваться в разделе 2.4.

По размеру преобразуемого блока шифры делятся на блочные и потоковые.

Блочные шифры осуществляют преобразование информации блоками фиксированной длины. Если длина шифруемого сообщения некратна размеру блока, то его добавляют до нужной длины последовательностью специального вида. Например, это может быть последовательность 100…0. После расшифровки последний блок просматривают справа налево и отбрасывают «хвост» до первой единицы включительно. Чтобы подобное дополнение было применимо во всех случаях, если сообщение кратно длине блока, в его конец надо добавить целый блок указанного вида.

Потоковые шифры предназначены для преобразования сообщения поэлементно (элементом может быть бит, символ и т. п.). Примером такого вида шифров являются шифры гаммирования.

2.2. СИММЕТРИЧНЫЕ ШИФРЫ

2.2.1. Схема Фейстеля

Современные блочные шифры часто строятся на базе многократного повторения некоторого набора операций преобразования, называемых *раундом шифрования*.

В каждом раунде используется некоторая часть ключа, называемая *раундовым ключом*. Порядок генерации и использования раундовых ключей называется *расписанием использования ключа шифрования*.

В общем виде подобное итеративное преобразование может быть описано следующей формулой:

$$B_i = E(B_{i-1}, K_i), \tag{2.5}$$

где E — раундовая функция шифрования, B_i — выходной блок, B_{i-1} — входной блок для i-го раунда, K_i — ключ, используемый на -м раунде; $i = 1 \ldots N$, где N — число раундов. Преобразование должно быть обратимо. Пусть D — раундовая функция дешифрования. Тогда раунд обратного преобразования описывает формула:

$$B_i = D(B_{i-1}, K_{N-i+1}). \tag{2.6}$$

Здесь надо обратить внимание на расписание использования ключей в случае прямого и обратного преобразования. Так, в первом раунде при шифровании будет использоваться первый раундовый ключ, а в первом раунде при расшифровывании — последний.

Для разработки итерационных блочных шифров широко используется схема, предложенная в начале 1970-х годов Хорстом Фейстелем (Horst Feistel). Данная схема, также называемая сетью Фейстеля, приведена на рис. 2.2. Ее достоинство заключается в том, что она позволяет использовать любые (в том числе необратимые) функции *F* для реализации обратимых шифрующих преобразований.

Входной блок сообщения разбивается на два равных по длине полублока: левый *L* и правый *R*.

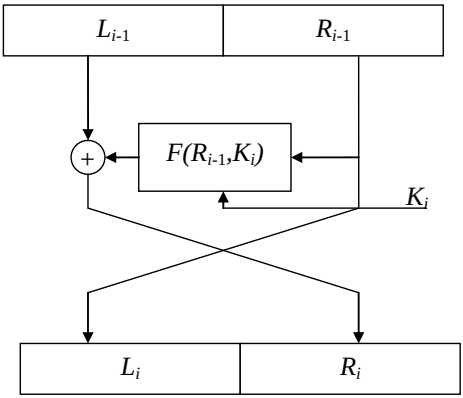


Рис. 2.2. Прямое преобразование
по схеме Фейстеля

Прямое преобразование осуществляется в соответствии со следующими соотношениями:

$$L_i = R_{i-1},$$
$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i), \qquad (2.7)$$

где $\oplus$ — операция побитного сложения по модулю 2 (в табл. 2.1 приведено ее определение). Тогда исходный блок данных — это $L_0 \mid R_0$, а $L_N \mid R_N$ — это выходной блок данных.

Таблица 2.1

Операция сложения по модулю 2

X	Y	X⊕Y
0	0	0
0	1	1
1	0	1
1	1	0

Как видно из табл. 2.1, для всех возможных значений операндов выполняется соотношение: (X Y) Y = X. Именно это свойство и позволяет при преобразовании использовать в числе прочих необра-

тимые функции F и, несмотря на это, иметь возможность провести обратное преобразование. Обратное преобразование по схеме Фейстеля представлено на рис. 2.3 и описывается формулами:

$$R_i = L_{i-1},$$
$$L_i = R_{i-1} \oplus F(L_{i-1}, K_{N-i+1}). \qquad (2.8)$$

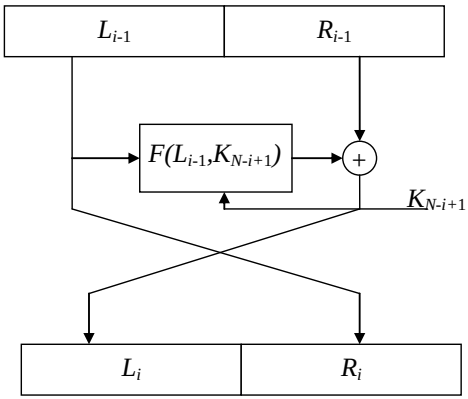


Рис. 2.3. Обратное преобразование
по схеме Фейстеля

2.2.2. Шифр DES

Алгоритм шифрования DES (от англ. «Data Encryption Standard») был опубликован в 1977 году и предназначался для защиты важной, но несекретной информации в государственных и коммерческих организациях США. Реализованные в нем идеи были во многом позаимствованы в более ранней разработке корпорации IBM — шифре «Люцифер» (а как раз в IBM работал Хорст Фейстель, автор рассмотренной выше схемы). Но для своего времени «Люцифер» был слишком сложным, и его реализации отличались низким быстродействием.

Шифр DES является блочным — преобразования в нем проводятся блоками по 64 бита. Ключ также 64-битный, но значащими яв-

ляются только 56 бит — каждый 8-й разряд использовался для контроля четности (шифр разрабатывался тогда, когда аппаратура была не слишком надежной и подобные проверки были необходимы). Обобщенная схема алгоритма представлена на рис. 2.4.

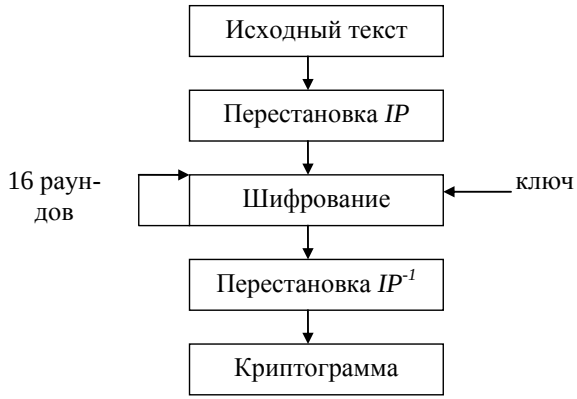


Рис. 2.4. Обобщенная схема шифра DES

Конечная перестановка — IP^{-1} — является обратной по отношению к начальной — *IP*. Задающие их таблицы жестко определены стандартом (таблицы можно узнать из официального описания стандарта, также они приводятся, например в [9]).

Раунды шифрования стоятся в соответствии с рассмотренной ранее схемой Фейстеля (рис. 2.2). Шифрование производится в 16 раундов. Схема раундовой функции *F* представлена на рис. 2.5.

Сначала 32-битный правый полублок преобразуемого текста расширяется до 48 бит с помощью функции расширения *E*. Эта функция производит дублирование и перестановку некоторых элементов блока. В табл. 2.2 показано, как эта функция работает. После расширения в первых позициях полученного 48-битного блока будут стоять 32-й, 1-й и т. д. биты входного блока.

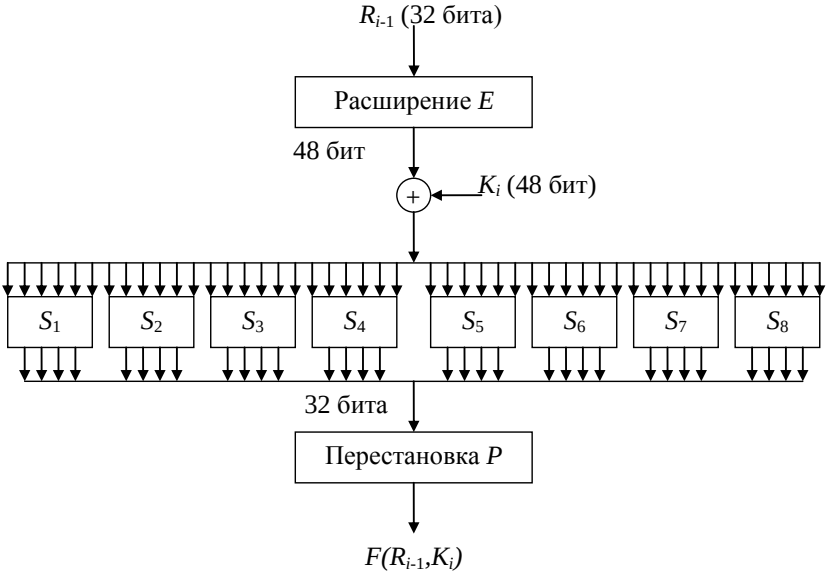


Рис. 2.5. Схема раундовой функции шифра DES

Таблица 2.2

Функция *E* — расширение блока

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

После расширения полученный 48-разрядный блок складывается побитно по модулю 2 с раундовым ключом K_i, схема генерации которого будет рассмотрена ниже.

Далее производится подстановка по таблицам $S_1, \ldots, S_8$, в результате которой каждому 6-разрядному входному значению ставится

в соответствие 4-разрядное выходное. Таким образом, получив на входе 48 бит, на выходе снова имеем 32. S_i представляет собой таблицу с 16-ю столбцами и 4-мя строками, содержащую 4-битные элементы. Таблицы подстановки, также как и перестановки, четко определены стандартом.

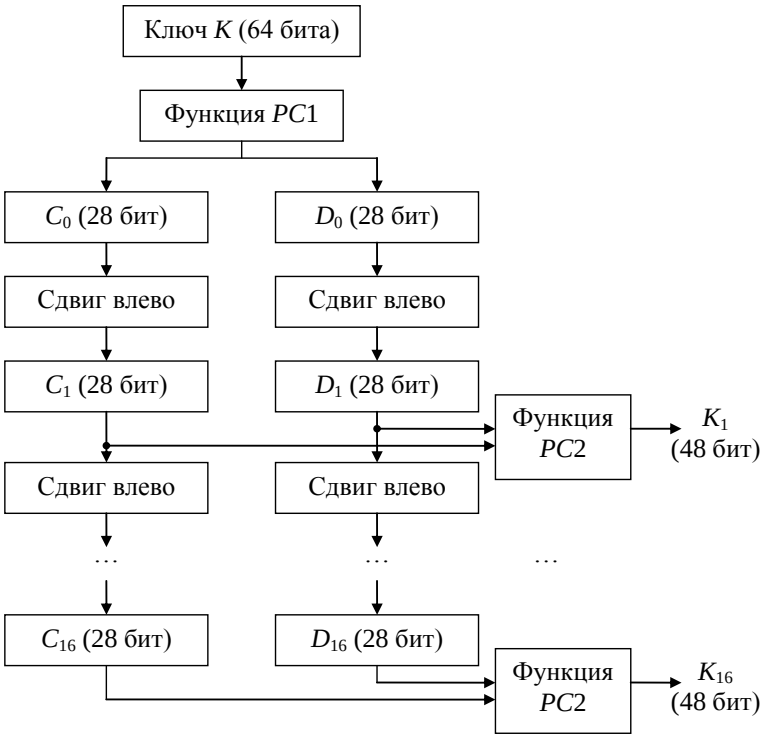


Рис. 2.6. Вычисление раундовых ключей

Пусть на вход подстановки подается 6-разрядный блок $B=b_1b_2...b_6$. Тогда совокупность старшего и младшего разрядов b_1b_6 будет указывать номер строки, а четырехбитное значение $b_2b_3b_4b_5$ — номер столбца. Из ячейки на пересечении найденных строки и столбца будет браться выходное значение подстановки. Полученный со-

единением выходных значений подстановок S_1, …, S_8 32-битный блок подвергается перестановке P, порядок которой также строго определен в стандарте.

Рассмотрим теперь порядок формирования раундовых ключей на основе секретного. В общем виде алгоритм представлен на рис. 2.6.

Как было отмечено ранее, секретный ключ шифра DES имеет длину 64 бита, но каждый 8-й предназначается для контроля четности, поэтому эффективная длина ключа — 56 бит. Функция *PC*1 (сокращение «PC» от англ. «Permuted Choice» — выбор с перестановкой) осуществляет перестановку элементов исходного блока, отбрасывая 8-й, 16-й и т. д. биты. После перестановки полученный блок делится на полублоки C_0 и D_0 длиной 28 бит каждый.

В зависимости от номера шага полублоки C_i и D_i независимо друг от друга преобразуются путем циклического сдвига влево на одну или две позиции (сдвиг на одну позицию производится на 1, 2, 9 и 16-м шагах, в остальных случаях — сдвиг на две позиции). Функция *PC2* преобразует блок $C_i|D_i$, переставляя элементы и отбирая 48 бит, которые и формируют раундовый ключ шифрования K_i (i = 1–16).

Для того чтобы иметь возможность использовать шифр DES для решения различных криптографических задач, определены 4 режима его работы:

- электронная кодовая книга (англ. «Electronic Code Book» — ECB);
- сцепление блоков шифра (англ. «Cipher Block Chaining» — CBC);
- обратная связь по шифртексту (англ. «Cipher FeedBack» — CFB);
- обратная связь по выходу (англ. «Output FeedBack» — OFB).

При использовании *режима ECB* защищаемое сообщение разбивают на 64-битные блоки M_i. Каждый такой блок шифруют независимо от других, с использованием одного и того же ключа шифрова-

ния (рис. 2.7). При расшифровывании криптограммы C_i также преобразуют независимо.

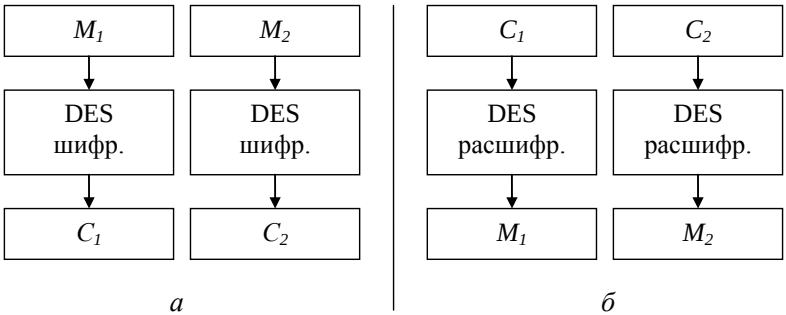


Рис. 2.7. Шифрование (*а*) и расшифровывание (*б*) в режиме ECB

Достоинством данного режима является простота его реализации. Главный недостаток режима ECB заключается в том, что если в исходном сообщении есть повторяющиеся блоки, то и значения соответствующих блоков криптограммы будет совпадать. А это даст криптоаналитику противника дополнительную информацию о содержании сообщения. Поэтому режим ECB рекомендуют использовать для защиты небольших объемов данных (например, криптографических ключей), где вероятность появления совпадающих блоков сообщения невелика.

Указанного выше недостатка лишен *режим CBC*. Исходное сообщение, как и в предыдущем случае, разбивается на блоки M_i по 64 бита. Первый блок складывается побитно по модулю 2 с 64-битным блоком, называемым инициализирующим вектором *IV*, который известен обеим сторонам взаимодействия, периодически ими меняется и держится в секрете от других. Блок исходного сообщения M_2 модифицируется с использованием блока криптограммы C_1 и т. д. Аналогичные действия производятся при расшифровывании. Схема преобразования представлена на рис. 2.8.

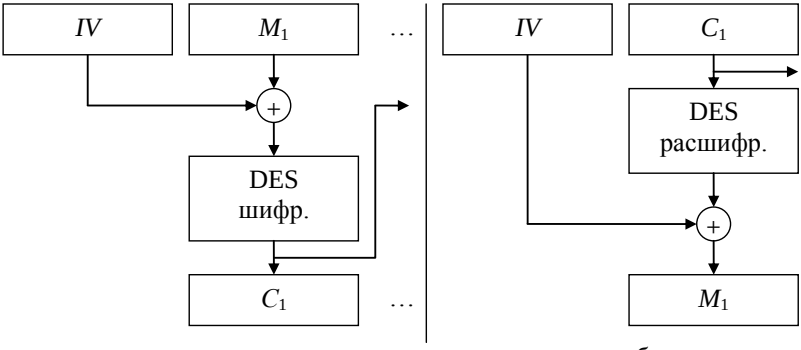


Рис. 2.8. Шифрование (*а*) и расшифровывание (*б*) в режиме CBC

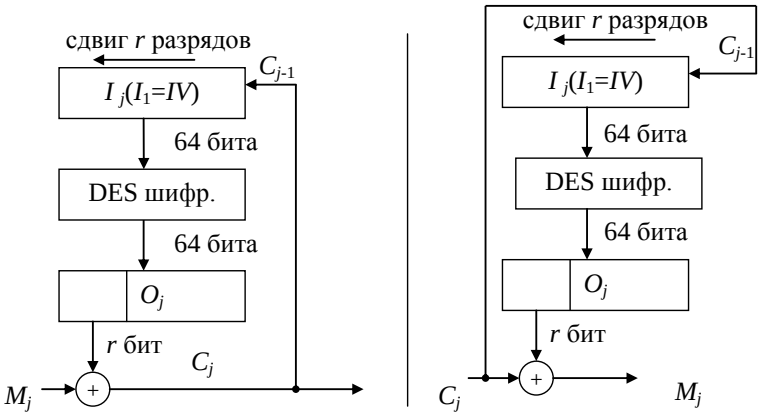
Режим CBC используется для шифрования больших сообщений. Как легко заметить, последний блок криптограммы зависит от инициализирующего вектора, каждого бита открытого текста и значения секретного ключа. Поэтому его можно использовать для контроля целостности и аутентификации сообщений, задавая ему фиксированное значение и проверяя его после расшифровки.

Режим CFB используется в тех случаях, когда длина преобразуемого блока отличается от 64 бит. Пусть необходимо зашифровать сообщение, считываемое последовательно блоками по r бит, где $1 \le r \le 64$. Для построения преобразования используется сдвиговый регистр I, куда на 1-м шаге преобразования помещается инициализирующий вектор IV. Схема преобразования представлена на рис. 2.9.

Преобразуемое сообщение M разбито на блоки по r бит, обозначенные на рис. 2.9 как M_j. Блок криптограммы C_j будет равен M_j, сложенному побитно по модулю 2 с r старшими битами зашифрованного на j-м шаге блока. Шифруемое значение I_j получается путем сдвига предыдущего блока I_{j-1} влево на r позиций и записи блока криптограммы C_{j-1} в младшие позиции.

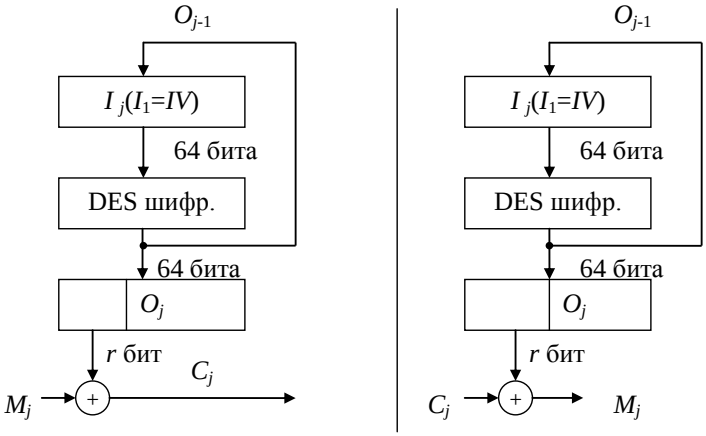
При расшифровывании сдвиговый регистр также инициализируется значением IV. Для того чтобы получить ту же последователь-

ность вспомогательных значений O_j, что и при шифровании, здесь также используется DES шифрование (а не расшифровывание, как например, при обратном преобразовании в режиме CBC).



а *б*

Рис. 2.9. Шифрование (*а*) и расшифровывание (*б*) в режиме CFB



а *б*

Рис. 2.10. Шифрование (*а*) и расшифровывание (*б*) в режиме OFB

Режим OFB также позволяет шифровать блоки, меньшие по длине, чем 64 бита. Его схема представлена на рис. 2.10. Аналогично режиму CFB, сдвиговый регистр сначала содержит значение инициализирующего вектора. Есть два варианта модификации его значения. На рисунке представлен вариант с полной заменой на j-м шаге содержимого сдвигового регистра вспомогательным значением O_{j-1}. Второй вариант построения схемы предполагает, как и в случае CFB, сдвиг I_{j-1} влево на r разрядов и запись в младшие разряды сдвигового регистра старших r разрядов O_{j-1}.

При использовании режима OFB важно для каждого сеанса шифрования данных использовать новое начальное состояние сдвигового регистра (его можно передавать, в том числе, и открытым текстом). Связано это с тем, что в режимах OFB и CFB генерируется псевдослучайная последовательность чисел, которая накладывается на блоки открытого текста. В случае использования режима OFB, если дважды используются один и тот же инициализирующий вектор и ключ шифрования, то и генерируемые последовательности будут совпадать.

Некоторое время шифр DES считался достаточно безопасным. Но по мере развития вычислительной техники короткий 56-битный ключ привел к тому, что атака путем полного перебора ключевого множества стала относительно легко реализуемой. Чтобы увеличить стойкость алгоритма и в то же время сохранить существующие наработки (в виде программных и аппаратных реализаций алгоритма), было использовано многократное шифрование.

Сначала было предложено использовать повторное шифрование на разных ключах. Обозначим шифрование на ключе k как E_k, а расшифровывание как D_k. Тогда предлагаемая схема прямого преобразования описывалась как $C=E_{k2}(E_{k1}(M))$, обратного — $M=D_{k1}(D_{k2}(C))$. Однако впоследствии было доказано, что из-за того, что два раза используется одно и то же преобразование, против подобной схемы может быть успешно применена атака «встреча посередине». Ее суть за-

ключается в том, что если хранить в памяти большой объем предвычислений (расчет криптограммы для всех возможных значений ключа), то можно взломать приведенную выше схему двукратного шифрования за 2^{n+1} попыток (вместо 2^{2n} как было бы при удвоении длины ключа).

Более надежной оказалась схема, включающая шифрование, расшифровывание и повторное шифрование на различных ключах. Данный шифр получил название Triple DES. Варианты схемы его построения приведены на рис. 2.11.

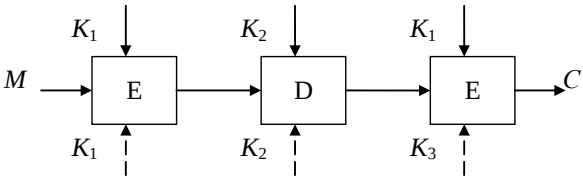


Рис. 2.11. Шифр Triple DES

Использование в шифре Triple DES различных ключей и преобразований (шифрование и расшифровывание) позволяет противостоять атаке «встреча посередине». В первом случае, выбор ключей производится так, как показано в верхней части рис. 2.11: сначала производится шифрование на ключе K_1, далее — расшифровывание на ключе K_2, после — шифрование на ключе K_1. Суммарная длина ключа — 112 бит. Более надежной считается схема с использованием в третьем преобразовании еще одного ключа — K_3 (изображена в нижней части рис. 2.11). Тогда суммарный ключ будет длиной 168 бит.

За счет более высокой надежности в настоящее время шифр Triple DES используется чаще, чем шифр DES.

2.2.3. Шифр ГОСТ 28147-89

Данный алгоритм симметричного шифрования был разработан в СССР и в качестве стандарта утвержден в 1989 году. Он считается достаточно стойким и широко используется в России теми предприяти-

ями и организациями, которым, в силу особенностей сферы их деятельности, необходимо применять сертифицированные средства криптографической защиты данных (это государственные и военные структуры, организации банковской сферы и т. д.).

Этот шифр преобразует сообщение 64-битными блоками, преобразование осуществляется в соответствии со схемой Фейстеля в 32 раунда, размер ключа — 256 бит. Алгоритм предусматривает 4 режима работы:

- шифрование данных в режиме простой замены (аналог режима ECB для шифра DES);

- шифрование данных в режиме гаммирования (аналог режима OFB для шифра DES);

- шифрование данных в режиме гаммирования с обратной связью;

- выработка имитовставки.

Ниже будет рассмотрен режим простой замены, являющийся основой для построения других режимов. Схема раунда шифрования приведена на рис. 2.12.

Преобразование производится в следующем порядке. Правый полублок R_{i-1} складывается по модулю 2^{32} (сложение 32-разрядных двоичных значений без переноса старшего разряда) с раундовым ключом K_j. Далее выполняется подстановка, задаваемая таблицами S_0, …, S_7, и полученное значение преобразуется с помощью циклического сдвига влево на 11 позиций. После этого выполняется побитное сложение по модулю 2 с левым полублоком L_{i-1} и перестановка полублоков.

Расписание использования раундовых ключей формируется следующим образом. 256-битный секретный ключ K представляется в виде сцепления 8-ми 32-битных подключей $K=K_7|K_6|K_5|K_4|K_3|K_2|K_1|K_0$. На первом раунде берется 0-й подключ, на втором — 1-й и т. д., на 9-м раунде опять используется 0-й подключ, на 24-м — 7-й, а вот на 25-м раунде преобразования вновь используется 7-й и далее ключи

используются в обратном порядке. Иными словами, номер раундового ключа j зависит от номера раунда i следующим образом:

$$j = (i-1) \bmod 8 \quad \text{для } 1 \le i \le 24,$$
$$j = (32-i) \bmod 8 \quad \text{для } 25 \le i \le 32. \qquad (2.9)$$

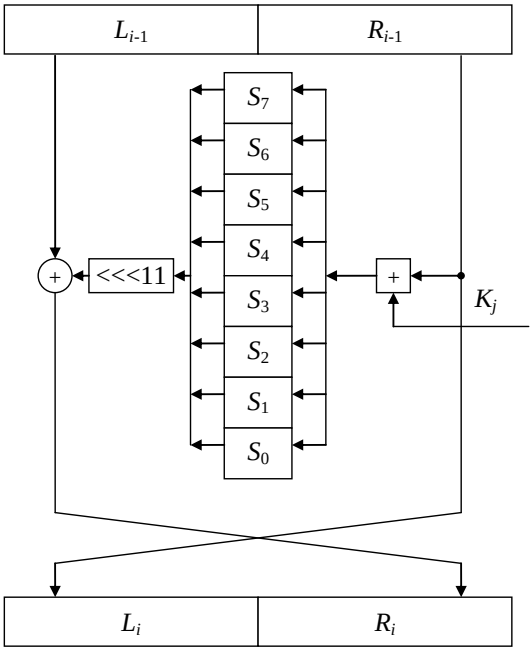


Рис. 2.12. Схема раунда алгоритма ГОСТ 28147-89

Подстановка проводится после разбиения входного значения на 4-битные подблоки. После того как правый полублок R был сложен с раундовым подключем, он разбивается на 8 частей $R=R_7|R_6|R_5|R_4|R_3|R_2|R_1|R_0$. Таблица подстановок S_i представляет собой вектор с 16-ю 4-битовыми элементами. Из нее берется элемент, номер которого совпадает со значением R_i, пришедшим на вход подстановки. Надо отметить, что значения таблиц подстановки не определены

стандартом, как это сделано, например, в шифре DES. В то же время стойкость алгоритма существенно зависит от их правильного выбора.

Считается, что конкретные значения этих таблиц должны храниться в секрете и это своеобразные ключевые элементы, которые являются общими для всей организации или подразделения и редко изменяются.

По сравнению с шифром DES у ГОСТ 28147-89 есть следующие достоинства:

- существенно более длинный ключ (256 бит против 56 у шифра DES), атака на который путем полного перебора ключевого множества на данный момент представляется невыполнимой;

- простое расписание использования ключа, что упрощает реализацию алгоритма и повышает скорость вычислений.

2.2.4. Шифр Blowfish

Шифр Blowfish был разработан известным американским криптографом Брюсом Шнейером (Bruce Schneier) в 1993 году. Алгоритм ориентирован на программную реализацию на 32-разрядных микропроцессорах. Его отличают высокая скорость и криптостойкость. Также в качестве отличительной особенности можно назвать возможность использовать ключ переменной длины. Шифр блочный, размер входного блока равен 64 битам. Преобразование блока выполняется в 16 раундов (есть версия с 111-ю раундами). Ключ переменной длины, максимально 448 бит.

До начала шифрования или расшифровывания данных производится расширение ключа. В результате, на базе секретного ключа получают расширенный, который представляет собой массив из 18 раундовых ключей $K_1, \ldots, K_{18}$ (размерность K_i — 32 бита) и матрицу подстановок Q с 4-мя строками, 256-ю столбцами и 32-битными элементами:

$$Q = \begin{vmatrix} Q_0^{(1)} & \dots & Q_{255}^{(1)} \\ Q_0^{(2)} & \dots & Q_{255}^{(2)} \\ Q_0^{(3)} & \dots & Q_{255}^{(3)} \\ Q_0^{(4)} & \dots & Q_{255}^{(4)} \end{vmatrix}. \qquad (2.10)$$

Данная матрица используется для задания нелинейной функции шифрующего преобразования *F(X)*, где *X* — 32-битный аргумент. *X* представляется в виде сцепления 4-х 8-битных слов $X = X_3 \mid X_2 \mid X_1 \mid X_0$, а сама функция задается формулой (+ здесь обозначает сложение по модулю 2^{32}, $\oplus$ — сложение по модулю 2):

$$F(X) = ((Q_{X_3}^{(1)} + Q_{X_2}^{(2)}) \oplus Q_{X_1}^{(3)}) + Q_{X_0}^{(4)}. \qquad (2.11)$$

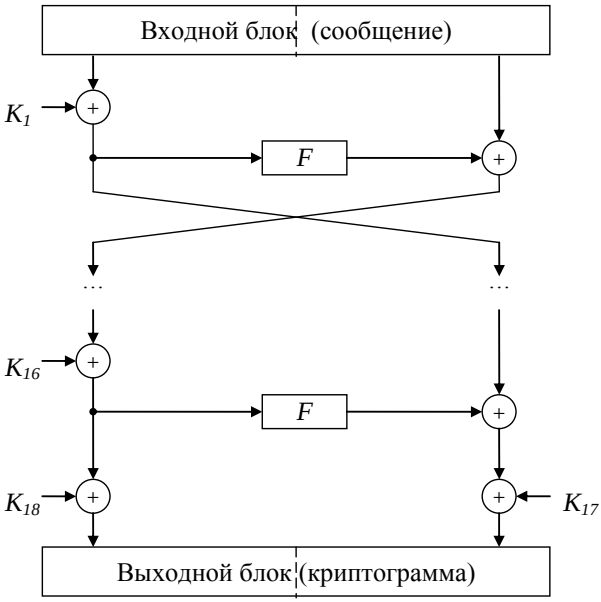


Рис. 2.13. Шифр Blowfish

Схема шифрующего преобразования приведена на рис. 2.13. Расширение секретного ключа *KS* производится следующим образом.

1) Начальный массив раундовых ключей K_i и элементов Q инициализируется фиксированными значениями. Например, в шестнадцатеричном представлении K_1 = 243F6A88 и т. д.

2) K_1 суммируется по модулю 2 с первыми 32-мя битами секретного ключа KS, K_2 — со следующими 32-мя битами и т. д. Когда ключ KS заканчивается, его начинают использовать сначала. Суммируются только K_i (без Q).

3) 64-битный блок нулей **0** = (0…0) зашифровывают с помощью Blowfish на ключах, полученных на шагах 1 и 2: $C_0 = Blowfish(\mathbf{0})$.

4) Раундовые подключи K_1 и K_2 заменяют полученным на шаге 3 значением C_0.

5) C_0 шифруют на модифицированных ключах $C_1 = Blowfish(C_0)$.

6) Раундовые ключи K_3 и K_4 заменяют значением C_1.

7) Процесс продолжается, пока не будут получены сначала все пары раундовых ключей (9 пар), а затем все пары элементов матрицы Q (512 пар).

Таким образом, расширение ключа требует шифрования 521 блока данных. Эта процедура дополнительно осложняет атаку путем перебора ключевого множества, т. к. нарушитель будет вынужден проводить процедуру расширения для каждого возможного ключа.

2.3. УПРАВЛЕНИЕ КРИПТОГРАФИЧЕСКИМИ КЛЮЧАМИ ДЛЯ СИММЕТРИЧНЫХ ШИФРОВ

Под *ключевой информацией* понимают совокупность всех ключей, действующих в системе. Если не обеспечено достаточно надежное и безопасное управление ключевой информацией, то эффект от применения криптографической защиты данных может быть сведен к нулю: завладев ключами нарушитель сможет получить доступ и к защищаемой информации. Процесс управления ключами включает в себя реализацию трех основных функций:

- генерация ключей;
- хранение ключей;
- распределение ключей.

Генерация ключей должна производиться таким образом, чтобы предугадать значение ключа (даже зная, как он будет генерироваться) было практически невозможно. В идеальном случае, вероятность выбора конкретного ключа из множества допустимых равна 1/N, где N — мощность ключевого множества (число его элементов).

Для получения ключей используют аппаратные и программные средства генерации случайных значений. Для систем с высокими требованиями к уровню безопасности более предпочтительными считаются аппаратные датчики, основанные на случайных физических процессах. В то же время, из-за дешевизны и возможности неограниченного тиражирования наиболее распространенными являются программные реализации. Но надо учитывать, что получаемая в этом случае последовательность будет псевдослучайной — если программный генератор повторно запустить с такими же начальными значениями, он выдаст ту же последовательность.

В программных генераторах ключей нередко используют алгоритмы шифрования и ключи, специально резервируемые для задач генерации. В качестве начальных значений могут браться, например, значения таймера вычислительной системы.

Рекомендуется регулярно проводить замену ключей, используемых в системе. В некоторых случаях вместо замены допустимо использовать процедуру модификации. *Модификация ключа* — генерация нового ключа из предыдущего значения с помощью односторонней функции (т. е. такой функции, для которой обратное преобразование вычислить практически невозможно, более подробно см. раздел 2.5). Но в этом случае надо учитывать, что новый ключ безопасен в той же мере, что и прежний, т. к. противник может повторить всю цепочку модификаций.

При организации хранения ключей симметричного шифрования необходимо обеспечить такие условия работы, чтобы секретные ключи никогда были записаны в явном виде на носителе, к которому может получить доступ нарушитель. Например, это требование можно

выполнить, создавая иерархии ключей. Трехуровневая иерархия подразумевает деление ключей на:

- главный ключ;
- ключ шифрования ключей;
- ключ шифрования данных (сеансовый ключ).

Сеансовые ключи — нижний уровень иерархии — используются для шифрования данных и аутентификации сообщений. Для защиты этих ключей при передаче или хранении используются *ключи шифрования ключей*, которые никогда не должны использоваться как сеансовые. На верхнем уровне иерархии располагается *главный ключ* (или мастер-ключ). Его применяют для защиты ключей второго уровня. Для защиты главного ключа в системах, использующих только симметричные шифры, приходится применять не криптографические средства, а, например, средства физической защиты данных (ключ записывается на съемный носитель, который после окончания работы изымается из системы и хранится в сейфе, и т. п.). В относительно небольших информационных системах может использоваться двухуровневая иерархия ключей (главный и сеансовые ключи).

При *распределении ключей* необходимо выполнить следующие требования:

- обеспечить оперативность и точность распределения ключей;
- обеспечить секретность распределения ключей.

Распределение ключей может производиться:

- с использованием одного или нескольких центров распределения ключей (централизованное распределение);
- прямым обменом сеансовыми ключами между пользователями сети (децентрализованное распределение ключей).

Децентрализованное распределение ключей симметричного шифрования требует наличия у каждого пользователя большого количества ключей (для связи с каждым из абонентов системы), которые необходимо сначала безопасно распределить, а потом обеспечивать их секретность в процессе хранения.

Централизованное распределение ключей симметричного шифрования подразумевает, что у каждого пользователя есть только один основной ключ для взаимодействия с центром распределения ключей. Для обмена данными с другим абонентом, пользователь обращается к серверу ключей, который назначает этому пользователю и соответствующему абоненту сеансовый симметричный ключ. Одной из самых известных систем централизованного распределения ключей является Kerberos.

Протокол Kerberos был разработан в Массачусетском технологическом институте в середине 1980-х годов и сейчас является фактическим стандартом системы централизованной аутентификации и распределения ключей симметричного шифрования. Поддерживается операционными системами семейства Unix, Windows (начиная с Windows'2000), есть реализации для Mac OS.

Протокол Kerberos обеспечивает распределение ключей симметричного шифрования и проверку подлинности пользователей, работающих в незащищенной сети. Реализация Kerberos — это программная система, построенная по архитектуре «клиент-сервер». Клиентская часть устанавливается на все компьютеры защищаемой сети, кроме тех, на которые устанавливаются компоненты сервера Kerberos. В роли клиентов Kerberos могут, в частности, выступать и сетевые серверы (файловые серверы, серверы печати и т. д.).

Серверная часть Kerberos называется центром распределения ключей (англ. «Key Distribution Center», сокр. KDC) и состоит из двух компонент:

- сервер аутентификации (англ. «Authentication Server», сокр. AS);
- сервер выдачи разрешений (англ. «Ticket Granting Server», сокр. TGS).

Каждому субъекту сети сервер Kerberos назначает разделяемый с ним ключ симметричного шифрования и поддерживает базу данных

субъектов и их секретных ключей. Схема функционирования протокола Kerberos представлена на рис. 2.14.

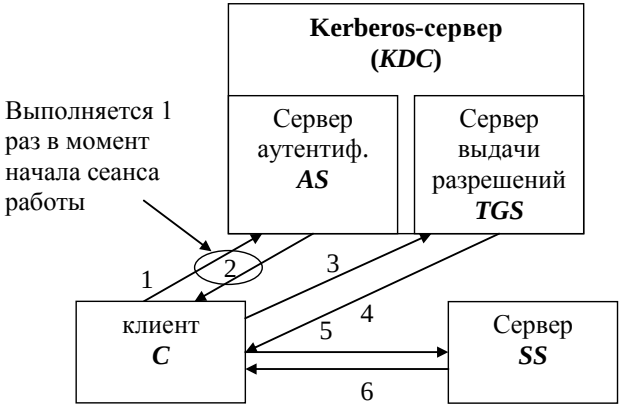


Рис. 2.14. Протокол Kerberos

Пусть клиент *C* собирается начать взаимодействие с сервером *SS* (от англ. «Service Server» — сервер, предоставляющий сетевые сервисы). В несколько упрощенном виде протокол предполагает следующие шаги [10,11].

1). C->AS: {c}.

Клиент *C* посылает серверу аутентификации *AS* свой идентификатор *c* (идентификатор передается открытым текстом).

2). AS->C: {{TGT}K_{AS_TGS}*,* K_{C_TGS}*}*K_C*,*

где K_C — основной ключ *C*;

K_{C_TGS} — ключ, выдаваемый *C* для доступа к серверу выдачи разрешений *TGS*;

{TGT} —— билет на доступ к серверу выдачи разрешений (англ. «Ticket Granting Ticket»); *{TGT}={c, tgs,* t_1*,* p_1*,* K_{C_TGS}*}*, где *tgs* — идентификатор сервера выдачи разрешений, t_1 — отметка времени, p_1 — период действия билета. Запись *{…}*K_X здесь и далее означает, что содержимое фигурных скобок зашифровано на ключе K_X.

На этом шаге сервер аутентификации *AS*, проверив, что клиент *C* имеется в его базе, возвращает ему билет для доступа к серверу выдачи разрешений и ключ для взаимодействия с сервером выдачи разрешений. Вся посылка зашифрована на ключе клиента *C*. Таким образом, даже если на первом шаге взаимодействия идентификатор *c* послал не клиент *C*, а нарушитель *X*, то полученную от *AS* посылку *X* расшифровать не сможет.

Получить доступ к содержимому билета *TGT* не может не только нарушитель, но и клиент *C*, так как билет зашифрован на ключе, который распределили между собой сервер аутентификации и сервер выдачи разрешений.

3). C->TGS: {TGT}K_{AS_TGS}*, {Aut*$_1$*}* K_{C_TGS}*, {ID},*

где *{Aut*$_1$*}* — аутентификационный блок — $Aut_1 = \{c,t_2\}$, t_2 — метка времени; *ID* — идентификатор запрашиваемого сервиса (в частности, это может быть идентификатор сервера *SS*).

Клиент *C* на этот раз обращается к серверу выдачи разрешений *TGS*. Он пересылает полученный от *AS* билет, зашифрованный на ключе K_{AS_TGS}, и аутентификационный блок, содержащий идентификатор *c* и метку времени, показывающую, когда была сформирована посылка.

Сервер выдачи разрешений расшифровывает билет *TGT* и получает из него информацию о том, кому был выдан билет, когда и на какой срок, ключ шифрования, сгенерированный сервером *AS* для взаимодействия между клиентом *C* и сервером *TGS*. С помощью этого ключа расшифровывается аутентификационный блок. Если метка в блоке совпадает с меткой в билете, это доказывает, что посылку сгенерировал на самом деле *C* (ведь только он знал ключ K_{C_TGS} и мог правильно зашифровать свой идентификатор). Далее делается проверка времени действия билета и времени оправления посылки *3*. Если проверка проходит, и действующая в системе политика позволяет клиенту *C* обращаться к клиенту *SS*, тогда выполняется шаг *4*.

4). TGS->C: {{TGS}K_{TGS_SS}*,*K_{C_SS}*}*K_{C_TGS}*,*

где K_{C_SS} — ключ для взаимодействия *C* и *SS*, *{TGS}* — от англ. «Ticket Granting Service» — билет для доступа к *SS* (обратите внимание, что такой же аббревиатурой в описании протокола обозначается и сервер выдачи разрешений). $\{TGS\} = \{c, ss, t_3, p_2, K_{C\_SS}\}$.

Сейчас сервер выдачи разрешений *TGS* посылает клиенту *C* ключ шифрования и билет, необходимые для доступа к серверу *SS*. Структура билета такая же, как на шаге *2*: идентификатор того, кому выдали билет; идентификатор того, для кого выдали билет; отметка времени; период действия; ключ шифрования.

5). C->SS: $\{TGS\}K_{TGS\_SS}, \{Aut_2\} K_{C\_SS}$,

где $Aut_2 = \{c, t_4\}$.

Клиент *C* посылает билет, полученный от сервера выдачи разрешений, и свой аутентификационный блок серверу *SS*, с которым хочет установить сеанс защищенного взаимодействия. Предполагается, что *SS* уже зарегистрировался в системе и распределил с сервером *TGS* ключ шифрования K_{TGS_SS}. Имея этот ключ, он может расшифровать билет, получить ключ шифрования K_{C_SS} и проверить подлинность отправителя сообщения.

6). SS->C: $\{t_4+1\}K_{C\_SS}$

Смысл последнего шага заключается в том, что теперь уже *SS* должен доказать *C* свою подлинность. Он может сделать это, показав, что правильно расшифровал предыдущее сообщение. Вот поэтому *SS* берет отметку времени из аутентификационного блока *C*, изменяет ее заранее определенным образом (увеличивает на 1), шифрует на ключе K_{C_SS} и возвращает *C*.

Если все шаги выполнены правильно, и все проверки прошли успешно, то стороны взаимодействия *C* и *SS*, во-первых, удостоверились в подлинности друг друга, а во-вторых, получили ключ шифрования для защиты сеанса связи — ключ K_{C_SS}.

Нужно отметить, что в процессе сеанса работы клиент проходит шаги *1* и *2* только один раз. Когда нужно получить билет на доступ к другому серверу (назовем его *SS*1), клиент *C* обращается к серверу

выдачи разрешений *TGS* с уже имеющимся у него билетом, т. е. протокол выполняется начиная с шага *3*.

При использовании протокола Kerberos компьютерная сеть логически делится на области действия серверов Kerberos. *Kerberos-область* — это участок сети, пользователи и серверы которого зарегистрированы в базе данных одного сервера Kerberos (или в одной базе, разделяемой несколькими серверами). Одна область может охватывать сегмент локальной сети, всю локальную сеть или объединять несколько связанных локальных сетей. Схема взаимодействия между Kerberos-областями представлена на рис. 2.15.

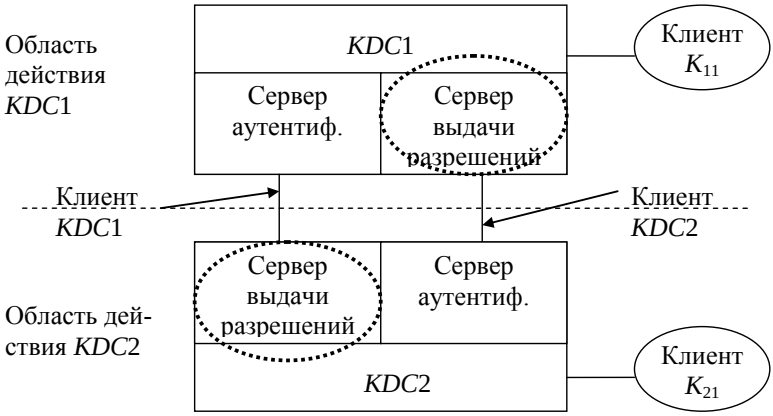


Рис. 2.15. Взаимодействие между Kerberos-областями

Для взаимодействия между областями должна быть осуществлена взаимная регистрация серверов Kerberos, в процессе которой сервер выдачи разрешений одной области регистрируется в качестве клиента в другой области (т. е. заносится в базу сервера аутентификации и разделяет с ним ключ).

После установки взаимных соглашений, клиент из области 1 (пусть это будет K_{11}) может установить сеанс взаимодействия с клиентом из области 2 (например, K_{21}). Для этого K_{11} должен получить у

своего сервера выдачи разрешений билет на доступ к Kerberos-серверу, с клиентом которого он хочет установить взаимодействие (на рис. 2.15 это сервер *KDC*2). Полученный билет содержит отметку о том, в какой области зарегистрирован владелец билета. Билет шифруется на ключе, разделенном между серверами *KDC*1 и *KDC*2. При успешной расшифровке билета удаленный Kerberos-сервер может быть уверен, что билет выдан клиенту Kerberos-области, с которой установлены доверительные отношения. Далее протокол работает, как обычно.

Кроме рассмотренных, Kerberos предоставляет еще ряд дополнительных возможностей. Например, указанный в структуре билета параметр *p* (период времени) задается парой значений «время начала действия» — «время окончания действия», что позволяет получать билеты отложенного действия.

Имеется тип билета «с правом передачи», что позволяет, например, серверу выполнять действия от имени обратившегося к нему клиента.

2.4. АСИММЕТРИЧНЫЕ ШИФРЫ

2.4.1. Основные понятия

Несмотря на достижения в области симметричной криптографии, к середине 1970-х годов стала остро осознаваться проблема неприменимости данных методов для решения целого ряда задач.

Во-первых, при использовании симметричных шифров необходимо отдельно решать часто нетривиальную задачу распределения ключей. Несмотря на использование иерархий ключей и центров распределения, в какой-то начальный момент ключ (или мастер-ключ) должен быть передан по безопасному каналу. Но такого канала может просто не быть, или он может быть достаточно дорогостоящим.

Во-вторых, при использовании методов симметричного шифрования подразумевается взаимное доверие сторон, участвующих во

взаимодействии. Если это не так, совместное использование одного и того же секретного ключа может быть нежелательно.

Третья проблема связана с необходимостью проведения аутентификации информации и защиты от угроз, связанных с отказом отправителя (получателя) от факта отправки (получения) сообщений.

Перечисленные проблемы являются весьма существенными, и работа над их решением привела к появлению асимметричной криптографии, также называемой криптографией с открытым ключом.

Рассмотрим ряд определений.

Односторонней (однонаправленной) функцией называется такая функция $F: X \to Y$, для которой выполняются следующие условия:

1) для всякого $x \in X$ легко вычислить значение функции $y = F(x)$, где $y \in Y$;

2) для произвольного $y \in Y$ невозможно (чрезвычайно сложно) найти значение $x \in X$, такое что $x = F^{-1}(y)$ (т. е. найти значение функции обратной F).

Односторонней функцией с секретом k, называется такая функция $F_k: X \to Y$, для которой выполняются следующие условия:

1) для всякого $x \in X$ легко вычислить значение функции $y = F_k(x)$, где $y \in Y$, даже в том случае, если значение k неизвестно;

2) не существует легкого (эффективного) алгоритма вычисления обратной функции $F_k^{-1}(y)$ без знания секрета k;

3) при известном k вычисление $F_k^{-1}(y)$ для $y \in Y$ не представляет существенной сложности.

В частности, односторонняя функция с секретом может быть использована для шифрования информации. Пусть M — исходное сообщение. Получатель выбирает одностороннюю функцию с секретом, и тогда любой, кто знает эту функцию, может зашифровать сообщение для данного получателя, вычислив значение криптограммы $C = F_k(M)$. Расшифровать данную криптограмму может только законный получатель, которому известен секрет k.

Первой публикацией в области криптографии с открытым ключом принято считать статью Уитфилда Диффи (Whitfield Diffie) и Мартина Хеллмана (Martin Hellman) «Новые направления в криптографии», вышедшую в свет в 1976 году.

В отличие от симметричных, в асимметричных алгоритмах ключи используются парами — открытый ключ (англ. «public key») и секретный или закрытый (англ. «private key»). Схема шифрования будет выглядеть следующим образом.

Получатель *B* генерирует пару ключей — открытый K_{B_pub} и секретный K_{B_pr}. Процедура генерация ключа должна быть такой, чтобы выполнялись следующие условия:

1) ключевую пару можно было бы легко сгенерировать;

2) сообщение, зашифрованное на открытом ключе, может быть расшифровано только с использованием секретного ключа;

3) зная только открытый ключ, невозможно рассчитать значение секретного.

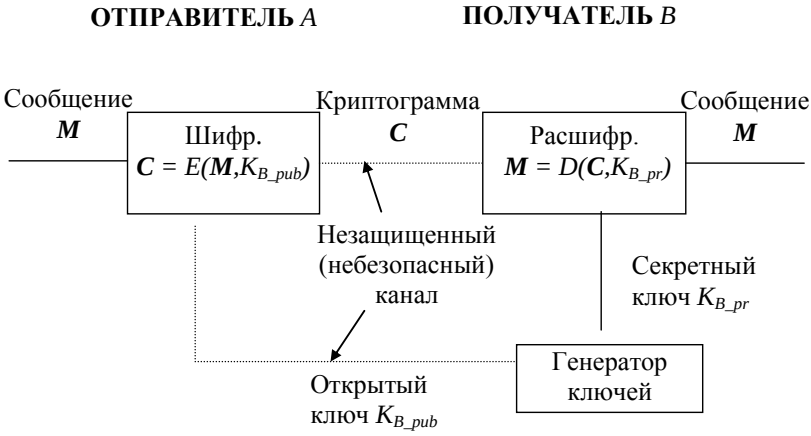


Рис. 2.16. Асимметричное шифрование

После генерации ключей, абонент *B* передает открытый ключ отправителю *A*, а секретный ключ надежно защищает и хранит у себя

(рис. 2.16). Пересылка открытого ключа может осуществляться по незащищенному каналу связи. Отправитель *A*, зная сообщение *M* и открытый ключ, может рассчитать криптограмму $C = E(M, K_{B\_pub})$ и передать ее получателю *B*. Получатель, зная секретный ключ, может расшифровать криптограмму $M = D(C, K_{B\_pr})$.

Нарушитель даже в том случае, если он смог перехватить криптограмму и открытый ключ, не может расшифровать криптограмму.

Если использовать определение односторонней функции с секретом, то алгоритм шифрования и открытый ключ задают прямое преобразование F_k, алгоритм расшифровывания задает обратное преобразование, а секретный ключ получателя играет роль «секрета» *k*.

Рассмотрим теперь вопрос аутентификации сообщений.

Электронная цифровая подпись (ЭЦП)[1] — это реквизит электронного документа, предназначенный для защиты данного электронного документа от подделки, полученный в результате криптографического преобразования информации с использованием закрытого ключа электронной цифровой подписи и позволяющий идентифицировать владельца сертификата ключа подписи, а также установить отсутствие искажения информации в электронном документе, а также обеспечивать неотказуемость подписавшегося.

Функции ЭЦП аналогичны обычной рукописной подписи:

- удостоверить, что подписанный текст исходит от лица, поставившего подпись;

- не дать лицу, подписавшему документ, возможности отказаться от обязательств, связанных с подписанным текстом;

- гарантировать целостность подписанного текста.

Важное отличие ЭЦП заключается в том, что электронный документ вместе с подписью может быть скопирован неограниченное число раз, при этом копия будет неотличима от оригинала.

Обобщенная схема системы ЭЦП представлена на рис. 2.17.

[1] Определение в соответствии федеральным законом Российской Федерации «Об электронной цифровой подписи».

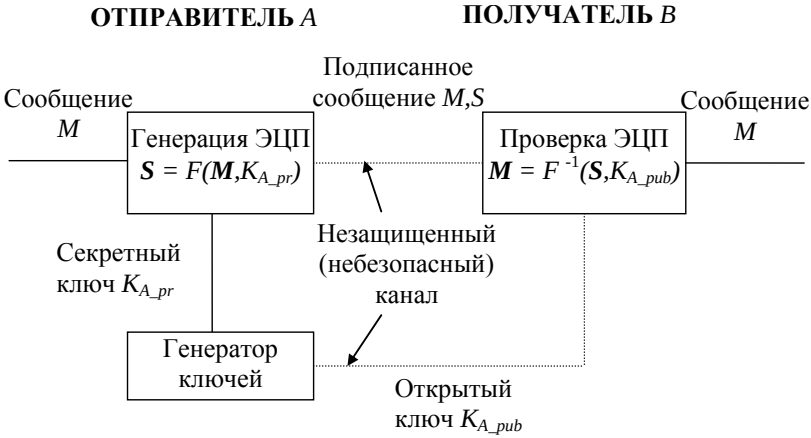


Рис. 2.17. Электронная цифровая подпись

В ходе преобразований здесь используется пара ключей отправителя сообщения. Тот факт, что при вычислении ЭЦП применяется секретный ключ отправителя, позволяет доказать происхождение и подлинность сообщения. Получатель, имея открытый ключ отправителя, проверяет ЭЦП, и если подпись корректна, то он может считать, что сообщение подлинное.

2.4.2. Распределение ключей по схеме Диффи–Хеллмана

Как уже отмечалось выше, основы асимметричной криптографии были заложены американскими исследователями У. Диффи и М. Хеллманом. Ими был предложен алгоритм, позволяющий двум абонентам, обмениваясь сообщениями по небезопасному каналу связи, распределить между собой секретный ключ шифрования.

Для того, чтобы лучше разобраться с особенностями данного алгоритма, рассмотрим несколько определений из теории чисел. Два целых числа n и n' называются *сравнимыми по модулю* m, если при делении на m они дают одинаковые остатки: $n \equiv n' \pmod m$, m — модуль сравнения. Таким образом, можно разбить множество целых чи-

сел $\boldsymbol{Z}$ на классы чисел, сравнимых между собой по модулю m и *называемых классами вычетов* по модулю m. Каждый класс вычетов имеет вид:

$$\{r\}_m = \{r + mk \mid k \in Z\}. \qquad (2.12)$$

Множество всех классов вычетов по модулю m обозначается как $\boldsymbol{Z_m}$ или $\boldsymbol{Z/mZ}$. Каждым двум классам $\{k\}_m$ и $\{l\}_m$ независимо от выбора их представителей можно сопоставить класс, являющийся их суммой и произведением, таким образом, однозначно определяются операции сложения и умножения. Множество $\{\boldsymbol{Z_m}, +, \times\}$ является коммутативным кольцом с единицей, а если число m — простое, то конечным полем.

Мультипликативная группа $\{\boldsymbol{Z_m}, \times\}$ при $m = 1, 2, 4, p^k, 2p^k$ (где $k \in \boldsymbol{N}$, p — нечетное простое число) является циклической [12], т. е. существует образующий элемент $a \in \boldsymbol{Z_m}$, такой что степени a в определенном порядке дают все значения от 0 до m–1. Элемент a также называется *первообразным корнем* по модулю m.

В алгоритме Диффи–Хеллмана в качестве односторонней функции используется возведение в степень по модулю простого числа:

$$y = a^x \bmod p. \qquad (2.13)$$

Здесь p — большое простое число (сейчас считается безопасным использовать модуль порядка 2^{1024} или более), a — первообразный корень по модулю p. Задача нахождения обратного значения, т. е. вычисления x по известному y, называется задачей дискретного логарифмирования и является вычислительно сложной. Иными словами, при достаточно больших значениях модуля, показателя и основания степени функцию (2.13) можно считать необратимой.

Пусть p — простое число, $p > 2$, и известно разложение p–1 на простые множители: $p - 1 = \prod_{j=1}^{k} q_j^{\alpha_j}$. Число a является первообразным корнем по модулю p тогда и только тогда, когда выполняются следующие условия [12]:

$$НОД(a, p) = 1;\ a^{(p-1)/q_j} \neq 1 (\mathrm{mod}\ p),\ j = 1,...,k, \qquad (2.14)$$

где *НОД(x,y)* — наибольший общий делитель чисел *x* и *y*.

Рассмотрим теперь алгоритм Диффи–Хеллмана по шагам. Пусть абоненты сети Алиса и Боб предварительно согласовали значения *a* и *p* из (2.13). При этом требуется, чтобы разложение числа *(p–1)* содержало большой простой множитель, например, *(p–1)*=2*t*, где *t* — простое.

1). Алиса выбирает секретный ключ X_A и вычисляет соответствующий ему открытый ключ $Y_A = a^{X_A} \bmod p$.

2). Боб, в свою очередь, определяет X_B и рассчитывает $Y_B = a^{X_B} \bmod p$.

3). Абоненты обмениваются открытыми ключами Y_A и Y_B.

4). Абоненты вычисляют общий секретный ключ. Алиса пользуется следующим соотношением: $K_{AB} = (Y_B)^{X_A} \bmod p$. Боб использует формулу: $K_{BA} = (Y_A)^{X_B} \bmod p$. Покажем, что $K_{AB} = K_{BA}$, воспользовавшись свойством ассоциативности операции умножения в конечном поле:

$$K_{AB} = (Y_B)^{X_A} \bmod p = (a^{X_B})^{X_A} \bmod p = (a^{X_A})^{X_B} \bmod p = K_{BA}. \qquad (2.15)$$

Таким образом, стороны смогли распределить общий секретный ключ K_{AB}. Нарушитель, который может перехватить передаваемые открытые ключи Y_A и Y_B, должен попытаться по ним вычислить общий секретный ключ без знания секретных ключей абонентов. На данный момент не найдено существенно лучшего пути решения данной задачи, чем дискретное логарифмирование, что и обеспечивает криптографическую стойкость алгоритма.

2.4.3. Криптографическая система RSA

Алгоритм RSA был предложен в 1977 году и стал первым полноценным алгоритмом асимметричного шифрования и электронной цифровой подписи. Алгоритм назван по первым буквам фамилий авторов — Рональд Райвест (Ronald Rivest), Ади Шамир (Adi Shamir) и

Леонард Адлеман (Leonard Adleman). Стойкость алгоритма основывается на вычислительной сложности задачи факторизации (разложения на множители) больших чисел и задачи дискретного логарифмирования.

Криптосистема основана на теореме Эйлера, которая гласит, что для любых взаимно простых чисел M и n ($M < n$) выполняется соотношение:

$$M^{\varphi(n)} \equiv 1 (\mathrm{mod}\, n), \qquad (2.16)$$

где $\varphi(n)$ — функция Эйлера. Эта функция равна количеству натуральных чисел, меньших n, которые взаимно просты с . По определению, $\varphi(1) = 1$. Также доказано, что для любых двух натуральных взаимно простых чисел a и b выполняется равенство $\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b)$.

Алгоритм строится следующим образом. Пусть M — блок сообщения, $0 \leq M < n$. Он шифруется в соответствии с формулой:

$$C = M^e (\mathrm{mod}\, n). \qquad (2.17)$$

В этом случае e — открытый ключ получателя. Тогда соответствующий ему секретный ключ d должен быть таким, что

$$M = C^d (\mathrm{mod}\, n) = (M^e)^d (\mathrm{mod}\, n) = M^{ed} (\mathrm{mod}\, n). \qquad (2.18)$$

Как уже отмечалось, теорема Эйлера утверждает, что $M^{\varphi(n)} = 1 (\mathrm{mod}\, n)$ или, что то же самое, $M^{k\varphi(n)+1} = M (\mathrm{mod}\, n)$, где k — натуральный множитель. Сопоставив данное выражение с выражением (2.18) получаем, что e и d должны быть связаны соотношением:

$$e \cdot d = k \cdot \varphi(n) + 1 \Leftrightarrow ed \equiv 1 (\mathrm{mod}\, \varphi(n)). \qquad (2.19)$$

Теперь предположим, что $n = p \cdot q$, где p и q — простые числа, причем $p \neq q$. Нетрудно показать, что для простого числа $p \neq 1$ функция Эйлера будет равна $\varphi(p) = p - 1$. Тогда, учитывая, что p и q — простые и не равны друг другу (а значит, они и взаимно простые), будет справедливо равенство:

$$\varphi(p \cdot q) = \varphi(p) \cdot \varphi(q) = (p-1) \cdot (q-1). \qquad (2.20)$$

Рассмотрим теперь алгоритм RSA «по шагам». Первым этапом является *генерация ключей*.

1) Выбираются два больших простых числа p и q, $p \neq q$.

2) Вычисляется модуль n: $n = p \cdot q$. Обратите внимание, что для криптосистемы RSA модуль n является частью открытого ключа и должен меняться при смене ключевой пары.

3) Случайным образом выбирается число d, такое, что $1 < d < (p-1) \cdot (q-1)$ и *НОД(d, (p–1)(q–1))*=1.

4) Вычисляется значение e такое, что:

$1 < e < (p-1) \cdot (q-1)$

$e \cdot d \equiv 1 \bmod((p-1)(q-1))$

Доказано, что для случая, когда *НОД(d, (p–1)(q–1))*=1, такое e существует и единственно.

В результате выполнения данных вычислений имеем открытый ключ, представленный парой чисел (n, e) и секретный ключ d.

Шифрование производится следующим образом.

Отправителю известен открытый ключ получателя — (n,e). Пусть M — секретное сообщение, которое надо зашифровать $M < n$. Криптограмма вычисляется по формуле:

$$C = M^e \bmod n. \qquad (2.21)$$

Криптограмма C передается получателю. Владелец секретного ключа d может расшифровать сообщение по формуле (2.22).

$$M = C^d \bmod n. \qquad (2.22)$$

Рассмотрим теперь схему построения э*лектронной цифровой подписи.* Сообщение M подписывается с использованием секретного ключа d (но для генерации подписи используется уже ключевая пара отправителя):

$$S = M^d \bmod n. \qquad (2.23)$$

Отправитель передает получателю подписанное сообщение, т. е. пару значений *(M,S)*. Проверка ЭЦП по открытому ключу (n,e) производится так:

$$M' = S^e \bmod n. \qquad (2.24)$$

Совпадение значений M и M' служит доказательством того, что сообщение получено от владельца соответствующего секретного ключа и не было изменено в процессе передачи.

Как видно, сами преобразования относительно просты. Основную сложность при реализации алгоритма RSA представляет этап генерации ключей. В частности, простые числа p и q выбираются такими, что:

- одно из них должно быть меньше другого на несколько порядков;

- $(p–1)$ и $(q–1)$ должны иметь как можно меньший НОД;

- хотя бы одно из чисел $(p–1)$, $(q–1)$ должно иметь в разложении большой простой множитель (например, $(p–1) = 2t$, где t — простое).

Точное определение, является ли большое число простым или нет, во многих случаях является вычислительно сложной задачей. Поэтому, как правило, используются «псевдопростые» тесты, которые позволяют с достаточно большой вероятностью отбросить числа не являющиеся простыми. Один из таких тестов основан на малой теореме Ферма, которая гласит, что если p — простое число и $1 \le x < p$, то $x^{p-1} = 1 \bmod p$. Проверив «кандидата» в простые числа с несколькими x, выбираемыми в соответствии со специальными требованиями, можно с большой вероятностью выяснить, является ли он простым.

Нахождение наибольшего общего делителя и определение значения e на шаге *4* алгоритма генерации ключей производится в соответствии с алгоритмом Евклида и обобщенным алгоритмом Евклида [7,9,12].

2.4.4. Криптографическая система Эль–Гамаля

В 1984 году американским исследователем египетского происхождения Тахером Эль–Гамалем (Taher Elgamal) были опубликованы алгоритмы шифрования с открытым ключом и ЭЦП, получившие его имя. Криптографическая система Эль–Гамаля использует ту же мате-

матическую основу, что и рассмотренная ранее схема распределения ключей Диффи–Хеллмана: в качестве односторонней функции в этой криптосистеме используется возведение в степень по модулю большого простого числа.

Алгоритм *шифрования* строится следующим образом.

Выбирается большое простое число p такое, что разложение числа $(p–1)$ содержит большой простой множитель, а также число a такое, что $1 < a < (p–1)$ и a — первообразный корень по модулю p.

Получатель сообщения генерирует ключевую пару: случайным образом выбирает секретный ключ x ($0 < x < p$) и вычисляет открытый ключ $y = a^x \bmod p$.

Для шифрования сообщения M ($0 \le M < p$), отправитель должен выполнить следующие действия.

1. Выбрать случайное число k такое, что $1 < k < p–1$, *НОД*$(k,p–1)=1$.

2. Вычислить вспомогательное значение $r = y^k \bmod p$.

3. Рассчитать значение криптограммы, состоящее из двух частей: $c_1 = a^k \bmod p$, $c_2 = rM \bmod p$.

Надо отметить, что в [10] приводится вариация данного алгоритма, где вторая часть криптограммы рассчитывается как $c_2^{'} = r \oplus M$, где $\oplus$ — побитное сложение по модулю 2.

Рассмотрим процесс расшифровки. Для того чтобы по c_2 определить M, получателю потребуется рассчитать значение r. С учетом того, что ему известен секретный ключ x и значение c_1, это становится возможным: $c_1^x \equiv (a^k)^x \equiv r \bmod p$. Тогда $M \equiv c_2 r^{-1} \equiv c_2 (c_1^{\,x})^{-1} \bmod p$. Или для варианта со сложением по модулю 2: $M \equiv c_2^{'} \oplus c_1^{\,x} \bmod p$.

При использовании шифра Эль–Гамаля требуется, чтобы выбираемое в процессе шифрования число k каждый раз менялось. В противном случае, если сообщения M и M' предназначены одному получателю, и нарушитель смог узнать одно из них, то ему не составит

труда рассчитать и второе. Пусть, например, нарушитель знает сообщение M, соответствующие ему криптограммы c_1 и c_2, и криптограммы c_1' и c_2'. Из-за того, что k и ключи не менялись, будут совпадать r и r', c_1 и c_1'. M' нарушитель может рассчитать как:

$$M \equiv c_2 r^{-1} \bmod p; \qquad M' \equiv c_2' r^{-1} \equiv c_2' M c_2^{-1} \bmod p. \qquad (2.25).$$

Рассмотрим теперь предложенный Эль–Гамалем *алгоритм электронной цифровой подписи*. Надо отметить, что он получил широкое распространение и на его базе был разработан американский стандарт ЭЦП DSA (англ. «Digital Signature Algorithm»).

Как и при шифровании, стороны согласуют параметры a и p. После этого отправитель выбирает секретный ключ x и рассчитывает открытый ключ $y = a^x \bmod p$.

Подписываемое сообщение M должно удовлетворять условию $0 \le M < p$. Подписью абонента служит пара чисел r и s ($0 \le r < p$, $0 \le s < p$), которые удовлетворяют соотношению:

$$a^M \equiv y^r r^s \bmod p. \qquad (2.26).$$

Получатель, знающий сообщение и открытый ключ, может проверить выполнение этого равенства. Но только владелец секретного ключа x может правильно рассчитать значения r и s. Для этого он выполняет следующие действия.

1. Выбирает случайное число k: $1 < k < p–1$, *НОД(k,p–1)*=1.

2. Вычисляет $r = a^k \bmod p$.

3. Вычисляет s из уравнения $M \equiv xr + ks \bmod(p-1)$. Это уравнение получено, основываясь на доказанном в теории числе утверждении: если p — простое, a — целое, $1 < a < p$ (в этом случае, a и p будут и взаимно просты), то $a^x \equiv a^y \bmod p$ равносильно $x \equiv y \bmod(p-1)$ для любых целых неотрицательных x, y. Таким образом, $s \equiv (M - xr)k^{-1} \bmod(p-1)$. При выполнении условия *НОД(k,p–1)*=1, s существует и единственно.

Отправитель передает сообщение с подписью (M,r,s) получателю, который, пользуясь соотношением (2.26), может проверить ЭЦП.

При применении алгоритма ЭЦП Эль–Гамаля, также как и в случае шифрования, недопустимо использовать одно и то же значение k для подписи двух разных сообщений.

2.4.5. Совместное использование симметричных и асимметричных шифров

Основным достоинством криптографических алгоритмов с открытым ключом является возможность решения таких задач, как распределение ключа по небезопасному каналу, аутентификации сообщения и отправителя и т. д. В то же время асимметричные шифры работают существенно более медленно, чем симметричные. Это связано с необходимостью производить операции над сверхбольшими числами. Поэтому симметричные и асимметричные алгоритмы часто используют вместе — для распределения ключей и ЭЦП используют криптографию с открытым ключом, данные шифруют с помощью симметричных алгоритмов.

При анализе системы, в которой совместно используются несколько алгоритмов, принято оценивать сложность ее взлома по сложности взлома самого слабого звена. В литературе [9] приводится примерное соответствие длин ключей для алгоритма симметричного шифрования (атака производится путем перебора ключевого множества) и алгоритма RSA, обеспечивающих сопоставимую стойкость. Например, 64-битному ключу симметричного шифрования примерно соответствует 512-битный ключ RSA, а 128-битному — ключ RSA длиной более 2300 бит.

2.5. ХЭШ-ФУНКЦИИ

В рассмотренных в разделе 2.4 алгоритмах формирования ЭЦП длина подписи получается равной или даже большей, чем длина самого сообщения. Очевидно, что удостоверять подобным образом большой документ неудобно. Поэтому подписывается, как правило,

не само сообщение, а его «дайджест» — значение фиксированной длины, зависящее от подписываемого сообщения. Для формирования дайджеста используется *хэш-функция* (от англ. «hash function») — односторонняя функция, преобразующая строку произвольной длины в строку фиксированной длины. В криптографии используются хэш-функции 2 классов:

- хэш-функции без ключа;
- хэш-функции с ключом.

2.5.1. Хэш-функции без ключа

Хэш-функции без ключа делятся на слабые и сильные. *Слабой хэш-функцией* называется односторонняя функция $H(x)$, удовлетворяющая следующим условиям:

1) аргумент может быть строкой бит произвольной длины;

2) значение функции $H(x)$ должно быть строкой бит фиксированной длины;

3) значение $H(x)$ легко вычислить;

4) для любого фиксированного аргумента x вычислительно невозможно найти другой $x' \neq x$, такой что $H(x')=H(x)$.

Пара значений $x' \neq x$: $H(x')=H(x)$ называется *коллизией* хэш-функции.

Сильной хэш-функцией называется односторонняя функция $H(x)$, удовлетворяющая условиям 1 – 3 и последнему условию в следующей формулировке:

4') вычислительно невозможно любую пару значений $x' \neq x$, таких что $H(x')=H(x)$.

Любая сильная хэш-функция соответствует и требованиям для слабой, обратное в общем случае неверно. Для иллюстрации различия в сложности поиска коллизий слабой и сильной хэш-функции рассмотрим атаку с использованием «парадокса дней рождения»[1]. За-

[1] Парадокс дней рождения — известный пример из теории вероятности — утверждение, что если дана группа из 23 или более человек, то вероятность

фиксируем значение аргумента x и будем перебирать случайным образом $x' \neq x$ в поисках ситуации, когда $H(x')=H(x)$. Если предположить, что значения хэш-функции равномерно распределены, а число возможных значений $H(x)$ равно N, то потребуется *в среднем* перебор $N/2$ вариантов. Если же мы захотим найти какую-либо коллизию вообще, то задача оказывается проще: с вероятностью 0,63 для определения нужной пары значений потребуется опробовать $\sqrt{N}$ вариантов.

Чтобы минимизировать стоимость создания криптографических хэш-функций, разработчики часто используют один из существующих алгоритмов шифрования. Пусть $E(m,k)$ обозначает шифрование сообщения m на ключе k, а v_0 — стартовый вектор. Представим хэшируемое сообщение M в виде последовательности блоков $m_1, \ldots, m_t$ и будем их использовать в качестве раундовых ключей. Тогда $H(m)$ вычисляется следующим образом:

$$\begin{aligned} &h_0 = v_0, \\ &h_i = E(h_{i-1}, m_i),\ i = 1 \ldots t, \\ &H(m) = h_t. \end{aligned} \qquad (2.27)$$

Однако в варианте с использованием в качестве $E(m,k)$ алгоритма DES, хэш-функция оказалась недостаточно стойкой из-за подверженности атаке, основанной на «парадоксе дней рождения». И было предложено улучшить эту схему, например, следующим образом:

$$\begin{aligned} &h_0 = v_0, \\ &h_i = E(h_{i-1}, m_i) \oplus h_{i-1},\ i=1 \ldots t, \\ &H(m) = h_t. \end{aligned} \qquad (2.28)$$

Существует и ряд специально разработанных алгоритмов хеширования, один из которых — SHA-1.

того, что хотя бы у двух из них дни рождения (число и месяц) совпадут, превышает 1/2. Данное утверждение кажется противоречащим здравому смыслу, так как вероятность одному родиться в определенный день года довольно мала, а вероятность того, что двое родились в конкретный день — еще меньше.

2.5.2. Алгоритм SHA-1

Алгоритм SHA (Secure Hash Algorithm) разработан в США как часть стандарта SHS (Secure Hash Standard), опубликованного в 1993 году. Но в нем были обнаружены уязвимости, которые привели к необходимости модифицировать алгоритм. Через два года была опубликована новая версия — SHA-1, получившая на сегодняшний день широкое распространение.

Получая на входе сообщение произвольной длины менее 2^{64} бит, SHA-1 формирует 160-битное выходное сообщение (дайджест). Вначале преобразуемое сообщение *M* дополняется до длины, кратной 512 битам. Заполнитель формируется следующим образом: в конец преобразуемого сообщения добавляется 1, потом — столько нулей, сколько необходимо для получения сообщения, которое на 64 бита короче, чем кратное 512, после чего добавляют 64-битное представление длины исходного сообщения. Например, если сообщение длиной 800 бит, то 801-й бит = 1, потом добавляем нули до 960 бит, после чего в оставшихся 64-разрядах записывается число «800», в итоге хэшируем 1024-битное сообщение. Общая схема преобразования представлена на рис. 2.18. Перед началом преобразований инициализируется пять 32-битных переменных:

A = 0x67452301;

B = 0xEFCDAB89;

C = 0x98BADCFE;

D = 0x10325476;

E = 0xC3D2E1F0.

Эти значения присваиваются также переменным a_0, b_0, c_0, d_0, e_0. Преобразование производится над блоком сообщения размером 512 бит в 80 раундов. В процессе преобразования используются следующие нелинейные функции f_t:

$f_t(X,Y,Z)=(X \wedge Y) \vee ((\neg X) \wedge Z)$ для t = 0–19;

$f_t(X,Y,Z)=X \oplus Y \oplus Z$ для t = 20–39 и t = 60–79;

$f_t(X,Y,Z)=(X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z)$ для t = 40–59.

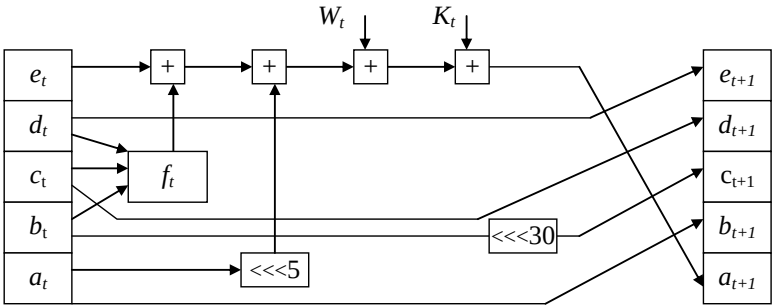


Рис. 2.18. Схема раунда алгоритма SHA-1

В процессе преобразования используются четыре константы:

K_t = 0x5A827999 для t = 0–19;

K_t = 0x6ED9EBA1 для t = 20–39;

K_t = 0x8F1BBCDC для t = 40–59;

K_t = 0xCA62C1D6 для t = 60–79.

Блок исходного сообщения M представляется в виде 16-ти 32-разрядных подблоков $M_0, \ldots, M_{15}$, которые используются для формирования значений W_t:

$W_t = M_t$ для t = 0–15;

$W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) <<< 1$ для t = 16–79.

Обозначение «<<< X» — циклический сдвиг влево на X разрядов, «+» — сложение по модулю 2^{32}.

После преобразования очередного 512-битного блока полученные значения *a, b, c, d, e* складываются со значениями *A, B, C, D, E* соответственно, и начинается обработка следующего блока (или полученное значение в виде сцепления *a, b, c, d, e* подается на выход, если обработанный блок был последним).

Таким образом, на выходе получаем 160-битный дайджест исходного сообщения.

2.5.3. Хэш-функции с ключом

Хэш-функцией с ключом называется односторонняя функция *H(k,x)* со следующими свойствами:

- аргумент x функции $H(k,x)$ может быть строкой бит произвольной длины;

- значение функции должно быть строкой бит фиксированной длины;

- при любых данных k и x легко вычислить $H(k,x)$;

- для любого x должно быть практически невозможно вычислить $H(k,x)$, не зная k;

- должно быть практически невозможно определить k, даже при большом числе известных пар $\{x, H(k,x)\}$ или вычислить по этой информации $H(k,x')$ для $x' \neq x$.

Часто такие функции также называются *кодами аутентификации сообщений* (англ. «Message Authentication Code», сокр. MAC). В отечественной литературе используется также термин *имитозащитная вставка* (или просто *имитовставка*).

Хэш-функцию с ключом можно построить на базе криптографической хэш-функции без ключа или алгоритма шифрования.

Пусть $H(x)$ — хэш-функция без ключа. Можно внедрить ключ в процесс хэширования, и получить хэш-функцию с ключом $H(k,x)$. Ниже представлены возможные варианты построения:

$$\begin{aligned} &H(k,x) = H(k|x), \\ &H(k,x) = H(x|k), \\ &H(k,x) = H(k_1|x|k_2), \text{ где } k = k_1|k_2. \end{aligned} \qquad (2.29)$$

Символ | в (2.28) обозначает конкатенацию, объединение строк аргументов.

Другой пример — построение хэш-функции с помощью шифра DES. Входной текст m разбивается на блоки $m_1,\ldots,m_t$ по 64 бита, которые преобразуются следующим образом (k — ключ шифрования):

$$\begin{aligned} &c_0=0, \\ &c_i=DES_k(m_i \oplus c_{i-1}),\ i=1,\ldots,t, \\ &H(k,m)= c_t. \end{aligned} \qquad (2.30)$$

2.6. ИНФРАСТРУКТУРА ОТКРЫТЫХ КЛЮЧЕЙ. ЦИФРОВЫЕ СЕРТИФИКАТЫ

Использование методов асимметричной криптографии сделало возможным безопасный обмен криптографическими ключами между отправителем и получателем, которые никогда друг друга не встречали и, возможно, находятся за многие километры друг от друга.

Но возникает другая проблема — как убедиться в том, что имеющийся у Вас открытый ключ другого абонента на самом деле принадлежит ему. Иными словами, возникает проблема аутентификации ключа. Без этого на криптографический протокол может быть осуществлена атака типа «человек посередине» (англ. «man in the middle»).

Идею данной атаки поясняет следующий пример. Абонент А (Алиса) хочет послать абоненту В (Боб) зашифрованное сообщение и берет его отрытый ключ из общедоступного справочника. Но, на самом деле, ранее нарушитель Е (Ева) подменил в справочнике открытый ключ Боба своим открытым ключом. Теперь Ева может расшифровать те сообщения, которые Алиса формирует для Боба, ознакомиться с их содержимым, возможно, зашифровать их на настоящем ключе Боба и переслать ему (рис. 2.19).

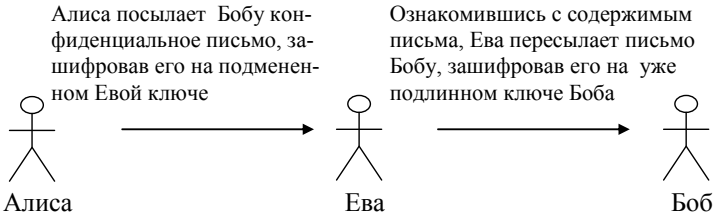


Рис. 2.19. Атака типа «man in the middle»

Избежать подобной атаки можно, подтвердив подлинность используемого ключа. Но Алиса и Боб лично никогда не встречались и передать, например, дискету с ключом из рук в руки не могут. Поэто-

му решение задачи подтверждения подлинности берет на себя третья доверенная сторона — некий арбитр, которому доверяют оба абонента. Заверяется ключ с помощью цифрового сертификата.

На самом деле, подобный способ применяется и вне компьютерных систем. Когда для подтверждения подлинности человека используется паспорт, то в роли третьей доверенной стороны выступает государство (от имени которого действовали в выдавшем паспорт отделе милиции).

Но вернемся к цифровым сертификатам. Для подтверждения подлинности открытых ключей создается инфраструктура открытых ключей (англ. «Public Key Infrastructure», сокр. PKI). PKI представляет собой набор средств, мер и правил, предназначенных для управления ключами, политикой безопасности и обменом защищенными сообщениями. Структура PKI представлена на рис. 2.20.

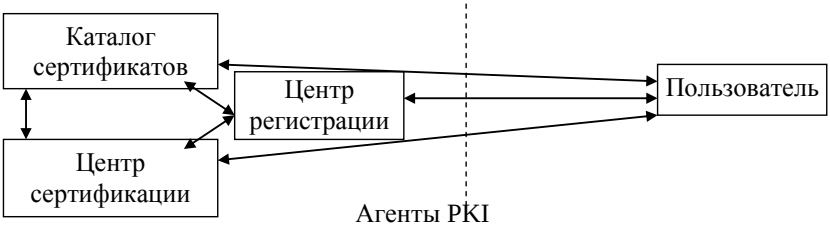


Рис. 2.20. Структура PKI

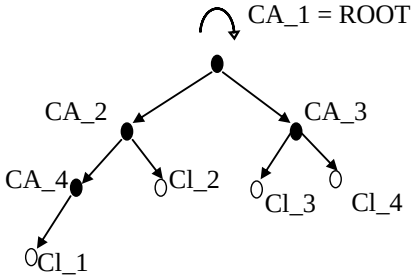


Рис. 2.21. Иерархия центров сертификации и клиентов

Таким образом, центры сертификации и пользователи формируют древовидную иерархическую структуру (рис. 2.21). В вершине этого дерева находится корневой центр сертификации, на рис. 2.21 — CA_1. Его особенность заключается в том, что он использует самоподписанный сертификат, т. е. сам заверяет свой ключ.

В приведенном примере CA_1 заверяет электронной подписью сертификаты подчиненных центров сертификации CA_2 и CA_3, а те, в свою очередь, подписывают сертификаты пользователей и центров более низкого уровня.

Перейдем к рассмотрению самих сертификатов. Наибольшее распространение получили цифровые сертификаты, формат которых определен стандартом X.509. На данный момент принята третья версия стандарта. Формат сертификата изображен на рис. 2.22 [11].

Номер версии содержит числовое значение, соответствующее номеру версии (для сертификата версии 1 равен 0 и т. д.). В первой версии X.509 не было уникальных номеров («ID Изготовителя», «ID Субъекта») и полей расширения. Во второй версии добавились указанные идентификаторы, в третьей — расширения.

Серийный номер — уникальный номер, присваиваемый каждому сертификату.

Алгоритм подписи — идентификатор алгоритма, используемого при подписании сертификата. Должен совпадать с полем *Алгоритм ЭЦП*.

Изготовитель — имя центра сертификации, выдавшего сертификат. Записывается в формате Relative Distinguished Name — RDN (варианты перевода названия — «относительное отдельное имя», «относительное характерное имя»). Данный формат используется в службах каталога, в частности, в протоколе LDAP. При записи Relative Distinguished Name используются специальные ключевые слова: CN (англ. «Common Name») — общее имя; OU (англ. «Organization Unit») — организационная единица; DC (англ. «Domain Component») — составная часть доменного имени.

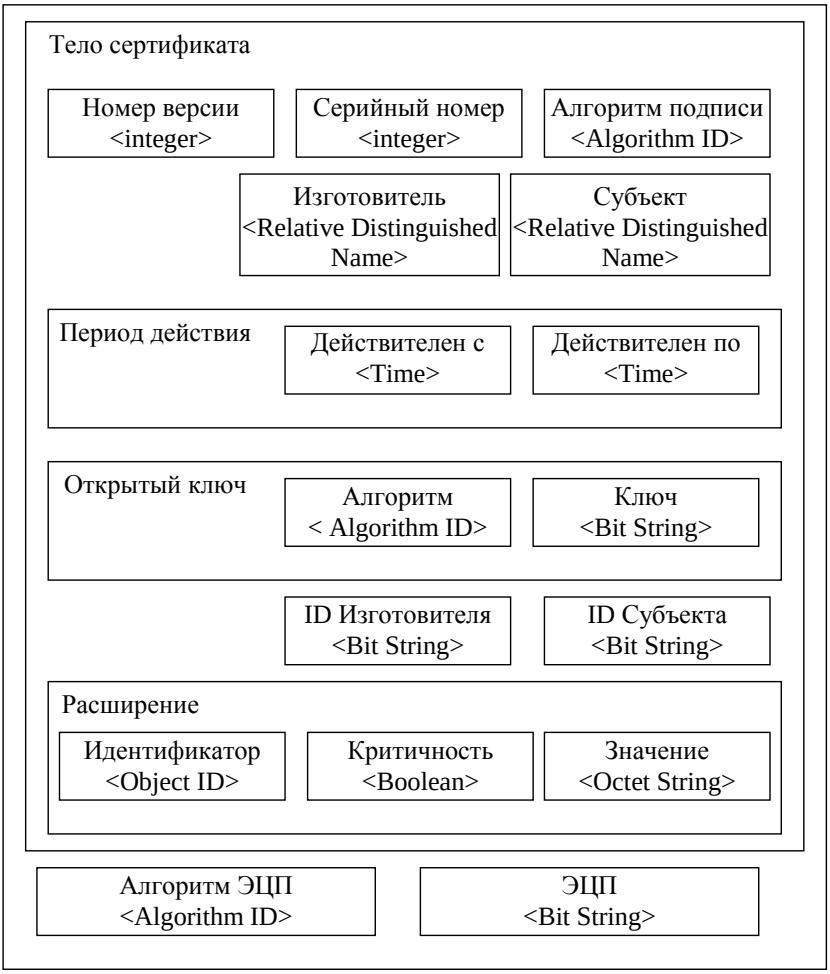


Рис. 2.22. Сертификат формата X.509 v.3

Например, в сертификате Microsoft Windows Hardware Compatibility, который находится в хранилище сертификатов Windows'XP значение данного поля следующее:

CN = Microsoft Root Authority,

OU = Microsoft Corporation,

OU = Copyright (c) 1997 Microsoft Corp.

Как видно, указывается имя центра сертификации, компания, которой он принадлежит и т. д.

Субъект — имя владельца сертификата, представленное в том же формате RDN. Для указанного в предыдущем примере сертификата значения данного поля:

CN = Microsoft Windows Hardware Compatibility,

OU = Microsoft Corporation,

OU = Microsoft Windows Hardware Compatibility Intermediate CA,

OU = Copyright (c) 1997 Microsoft Corp.

Период действия описывает временной интервал, в течение которого центр сертификации гарантирует отслеживание статуса сертификата (сообщит абонентам сети о факте досрочного отзыва сертификата и т. д.). Период задается датами начала и окончания действия.

Открытый ключ — составное поле, содержащее идентификатор алгоритма, для которого предназначается данный открытый ключ, и собственно сам открытый ключ в виде набора битов.

ID Изготовителя и *ID Субъекта* содержат уникальные идентификаторы центра сертификации и пользователя (на случай совпадения имен различных CA или пользователей).

Расширения — дополнительный атрибут, связанный с субъектом, изготовителем или открытым ключом и предназначенный для управления процессами сертификации. Более подробно он описан ниже.

Алгоритм электронной цифровой подписи (ЭЦП) — идентификатор алгоритма, используемый для подписи сертификата. Должен совпадать со значением поля *Алгоритм подписи*.

ЭЦП — само значение электронно-цифровой подписи для данного сертификата.

Расширения могут определять следующие дополнительные параметры:

- идентификатор пары открытый/секретный ключ центра сертификации (изготовителя), если центр имеет несколько различных ключей для подписи сертификатов;

- идентификатор конкретного ключа пользователя (субъекта), если пользователь имеет несколько сертификатов;

- назначение ключа, например, ключ для шифрования данных, проверки ЭЦП данных, для проверки ЭЦП сертификатов и т. д.;

- уточнение периода использования — можно сократить время действия сертификата, указанное в поле *Период действия* (период, в течение которого статус сертификата отслеживается, станет больше, чем разрешенное время использования сертификата);

- политики использования сертификата;

- выбор соответствия политик использования сертификата для центра сертификации и пользователя, если имеются различные варианты;

- альтернативное имя пользователя и центра сертификации;

- указания, является ли пользователь сам центром сертификации, и насколько глубоко разрешается разворачивать сертификационный путь.

Предположим, что ключевые пары сгенерированы, открытые ключи заверены сертификатами и размещены в каталоге, реализованном с помощью web-сервера, ftp-сервера, службы каталога или другой технологии. Теперь, если абонент *A* желает проверить подпись абонента *B* под полученным сообщением (или зашифровать для *B* сообщение с помощью его открытого ключа), он выполняет следующие действия [8]:

1) запрашивает в сетевом справочнике сертификат C_B открытого ключа подписи (шифрования) абонента *B*;

2) проверяет достоверность сертификата C_B (см. ниже);

3) в случае успеха проверяет подпись под сообщением (зашифровывает сообщение) с помощью открытого ключа, извлеченного из C_B.

Процедура проверки достоверности сертификата C_B состоит в следующем:

1) проверяется срок действия сертификата C_B, если он закончился, сертификат считается недостоверным;

2) из C_B извлекается имя ЦС, подписавшего этот сертификат, обозначим его D;

3) если $D = B$, то сертификат самоподписанный, он считается достоверным только, если $D = ROOT$ (хотя, возможно, в некоторых сетях право выдавать самоподписанные сертификаты имеет не один ROOT, это — политика сети);

4) если же $D \neq B$, то из справочника запрашивается сертификат C_D открытого ключа подписи абонента D, проверяется на достоверность сертификат C_D;

5) в случае отрицательного ответа принимается решение о недостоверности сертификата C_B, иначе из C_D извлекается открытый ключ K_D;

6) с помощью K_D проверяется подпись под сертификатом C_B, по результатам проверки этой подписи судят о достоверности C_B.

Если ключи шифрования досрочно вышли из обращения (причины могут быть различны — пользователь увольняется из компании, секретный ключ скомпрометирован и т. д.), центр сертификации извещает об этом остальных пользователей сети путем выпуска списка отозванных сертификатов (англ. «Certificate Revocation List», сокр. CRL). Структура CRL представлена на рис. 2.23. Поля списка содержат следующую информацию.

Номер версии определяет номер версии формата CRL. Текущая используемая версия — вторая.

Алгоритм подписи — идентификатор алгоритма, с помощью которого подписан CRL. Должен совпадать по значению с полем *Алгоритм ЭЦП*.

Изготовитель — имя центра сертификации в формате RDN.

Выпущен — дата выпуска CRL.

Следующий — дата, до которой будет выпущен следующий CRL.

Расширения описывают центр сертификации, точку для поиска CRL данного центра, номер данного списка и т. д.

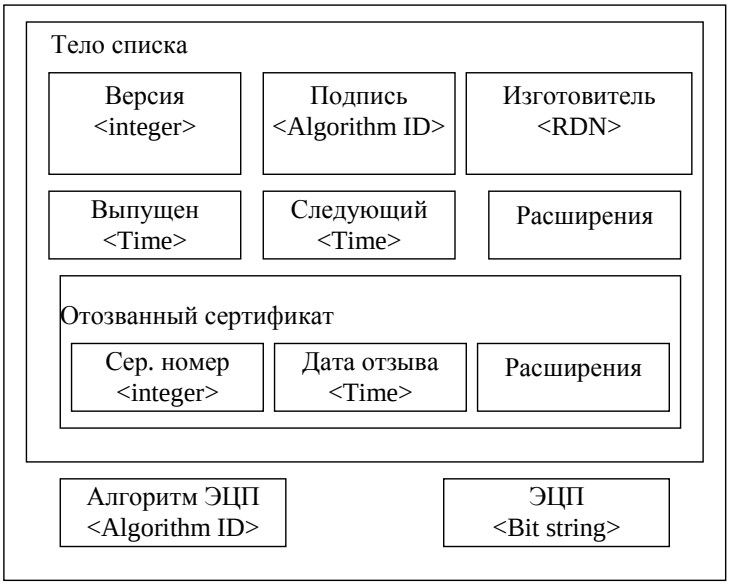


Рис. 2.23. Структура списка отозванных сертификатов

Отозванный сертификат — таких полей будет столько, сколько сертификатов отзывается — содержит номер отзываемого сертификата, дату, с которой сертификат отозван, описание причины отзыва.

Алгоритм ЭЦП — идентификатор алгоритма ЭЦП, используемого для подписи списка.

ЭЦП — сама электронная цифровая подпись.

Проблемы с CRL заключаются в том, что может возникнуть ситуация, когда ключ уже отозван, но CRL еще не выпущен, т. е. пользователи не могут получить информацию о компрометации ключа.

Кроме того, распространение CRL идет по запросу клиента, и нарушитель может препятствовать их получению.

Другая проблема PKI — самоподписанные сертификаты. Сертификат корневого ЦС должен раздаваться всем абонентам сети в начале работы и сохраняться в защищенном от подделки хранилище. Иначе нарушитель может попробовать навязать свой сертификат в качестве сертификата корневого центра.

Мы рассмотрели случай реализации *иерархической модели* PKI, при которой центры сертификации организованы в древовидную структуру с корневым центром сертификации на верху иерархии. На практике также встречаются другие варианты организации:

- *одиночный центр сертификации*, который выдает себе самоподписанный сертификат — данная модель часто реализуется в небольших организациях, но она имеет отмеченный выше недостаток, связанный с самоподписанными сертификатами;

- *одноранговая модель*, при которой независимые центры сертификации взаимно сертифицируют друг друга.

Надо отметить, что сфера применения цифровых сертификатов сейчас достаточно широка. В частности, они используются для распределения открытых ключей в протоколах защиты электронной почты S/MIME или PGP, с помощью цифровых сертификатов проверяется подлинность участников соединения по протоколу SSL и т. д.

3. ЗАЩИТА ИНФОРМАЦИИ В IP-СЕТЯХ

На сегодняшний день стек сетевых протоколов TCP/IP является наиболее широко используемым как в глобальных, так и в локальных компьютерных сетях. Именно поэтому методы и средства защиты передаваемых данных в IP-сетях представляют особый интерес.

В этом разделе будут рассмотрены криптографические протоколы, позволяющие защищать электронную почту, передаваемые данные на транспортном и сетевом уровне. Кроме того, учитывая боль-

шую роль межсетевых экранов в решении задач обеспечения сетевой безопасности, будет рассмотрен этот класс средств защиты.

3.1. ПРОТОКОЛ ЗАЩИТЫ ЭЛЕКТРОННОЙ ПОЧТЫ S/MIME

Протокол Secure Multipurpose Internet Mail Extensions (S/MIME) предназначен для защиты данных, передаваемых в формате MIME, в основном — электронной почты. Он был предложен в 1995 году компанией RSA Data Security Inc. Дальнейшие работы над протоколом велись рабочей группой организации Internet Engineering Task Force (IETF), разрабатывающей стандарты сети Интернет. На данный момент последней является версия 3.1 этого протокола, описываемая в документах RFC 3850, 3851, 3852. Протокол S/MIME предоставляет следующие криптографические услуги безопасности (криптографические сервисы):

- проверка целостности сообщения;
- установление подлинности отправителя (аутентификация);
- обеспечение секретности передаваемых данных (шифрование).

Нужно отметить, что сам по себе формат MIME описывает порядок форматирования писем, содержащих различные типы данных (обычный текст, текст в формате html, видео и графические файлы различный типов и т. д.). При использовании S/MIME добавляются новые типы (например, application/pkcs7-mime и application/pkcs7-signature). Это позволяет указать на то, что данные в этом разделе являются зашифрованными, подписанными и т. д. Протокол позволяет обычным почтовым клиентам защищать исходящую почту и интерпретировать криптографические сервисы, добавленные во входящую почту (расшифровывать сообщения, проверять их целостность и т. д.).

Стандарт определяет использование симметричных криптоалгоритмов для шифрования содержимого почтовых сообщений и алгоритма с открытым ключом для защиты передаваемого вместе с письмом ключа симметричного шифрования.

Протокол S/MIME позволяет использовать различные криптоалгоритмы, причем их список может расширяться. Изначально из сим-

метричных шифров могли использоваться RC2, DES или TripleDES. Для формирования дайджестов — алгоритмы MD5 и SHA1, причем версия 3 стандарта рекомендует использовать именно последний алгоритм (из-за того, что он формирует более длинный дайджест и считается более надежным). Защита симметричного ключа шифрования и ЭЦП в версии 2 осуществляется с помощью алгоритма RSA с ключом от 512 до 1024 бит. Версия 3 добавляет возможность использовать другие алгоритмы, например алгоритм Диффи–Хеллмана с ключом длиной до 2048 бит. Распределение и аутентификация открытых ключей производится с помощью цифровых сертификатов формата X.509. Таким образом, чтобы защищать переписку с помощью этого протокола, оба абонента должны сгенерировать ключевые пары и удостоверить открытые ключи с помощью сертификатов. На рис. 3.1 приведен фрагмент письма, содержащий открытый текст «This is a clear-signed message.» и подпись к нему.

S/MIME поддерживается многими почтовыми клиентами: Microsoft Outlook, Mozilla, The Bat! и т. д. Более широкое применение протокола сдерживается необходимостью наличия сертификатов у абонентов и плохой совместимостью с системами Web-почты.

```
Content-Type: multipart/signed;
          protocol="application/pkcs7-signature";
          micalg=sha1; boundary=boundary42

        --boundary42
        Content-Type: text/plain

        This is a clear-signed message.

        --boundary42
        Content-Type: application/pkcs7-signature; name=smime.p7s
        Content-Transfer-Encoding: base64
        Content-Disposition: attachment; filename=smime.p7s

        ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfHfYT6
        4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbB9HGTrfvbnj
        n8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
        7GhIGfHfYT64VQbnj756

        --boundary42--
```

Рис. 3.1. Фрагмент электронного письма с подписью

Альтернативой S/MIME является PGP (англ. «Pretty Good Privacy») — компьютерная программа, созданная Филиппом Циммерманном (Philip Zimmermann) в 1991 году. Данная программа положила основу работе над стандартом OpenPGP, последняя версия которого описана в RFC 4880. По функциональности S/MIME и PGP во многом сходны.

3.2. ПРОТОКОЛЫ SSL И TLS

Протокол Secure Sockets Layer (SSL) был разработан корпорацией Netscape Communications для обеспечения аутентификации, целостности и секретности трафика на сеансовом уровне модели OSI (с точки зрения четырехуровневой модели стека протоколов TCP/IP — на прикладном уровне). В январе 1999 года на смену SSL v3.0 пришел протокол TLS v1.0 (Transport Layer Security) последняя версия TLS v.1.2 описывается в RFC 5246. С точки зрения выполняемых действий, различия между этими протоколами SSL и TLS весьма невелики, в то же время, они несовместимы друг с другом [11].

SSL обеспечивает защищенное соединение, которое могут использовать протоколы более высокого уровня — HTTP, FTP, SMTP и т. д. Наиболее широко он используется для защиты данных, передаваемых по HTTP (режим HTTPS). Для этого должны использоваться SSL-совместимые web-сервер и браузер.

Протокол предусматривает два этапа взаимодействия клиента и сервера:

1) установление SSL-сессии (процедура «рукопожатия», от англ. «handshake»), на этом этапе может производиться аутентификация сторон соединения, распределение ключей сессии, определяются настраиваемые параметры соединения;

2) защищенное взаимодействие.

Протоколом SSL используются следующие криптоалгоритмы:

- асимметричные алгоритмы RSA и Диффи–Хеллмана;
- алгоритмы вычисления хэш-функций MD5 и SHA1;

- алгоритмы симметричного шифрования RC2, RC4, DES, TripleDES, IDEA.

В протоколе SSL v 3.0 и TLS перечень поддерживаемых алгоритмов является расширяемым. Для подтверждения подлинности открытых ключей используются цифровые сертификаты формата X.509.

Протокол SSL позволяет проводить следующие варианты аутентификации сторон взаимодействия:

- аутентификация сервера без аутентификации клиента (односторонняя аутентификация) — это наиболее часто используемый режим, позволяющий установить подлинность сервера, но не проводящий проверки клиента (ведь подобная проверка требует и от клиента наличия сертификата);

- взаимная аутентификация сторон (проверяется подлинность как клиента, так и сервера);

- отказ от аутентификации — полная анонимность; в данном случае SSL обеспечивает шифрование канала и проверку целостности, но не может защитить от атаки путем подмены участников взаимодействия.

Рассмотрим более подробно процедуру рукопожатия в режиме аутентификации сервера без аутентификации клиента. Она включает следующие шаги [13].

1. Клиент посылает серверу запрос на установление защищенного соединения, в котором передает, в частности, следующие параметры:

- дату и текущее время;

- сгенерированную клиентом случайную последовательность (RAND_CL);

- перечень поддерживаемых клиентом алгоритмов шифрования, хеширования и сжатия (если сжатие используется).

2. Сервер обрабатывает запрос от клиента и передает ему согласованный набор параметров:

- идентификатор SSL-сессии;

- идентификаторы криптографических алгоритмов из числа предложенных клиентом, которые будут использоваться в данной сессии (если по какой-либо причине предложенные алгоритмы или их параметры не удовлетворяют требованиям сервера, сессия закрывается);

- цифровой сертификат сервера формата X.509;

- случайную последовательность (RAND_SERV).

3. Клиент проверяет полученный сертификат и соответствие роли ключа его назначению, описанному в сертификате. При отрицательном результате проверки сессия закрывается, а при положительном клиент выполняет следующие действия:

- генерирует случайную 48-байтную последовательность, называемую Pre_MasterSecret, предназначенную для расчета общего секретного ключа;

- шифрует значение Pre_MasterSecret с использованием открытого ключа сервера, взятого из сертификата, и посылает криптограмму серверу;

- с помощью согласованной с сервером хэш-функции формирует общий секретный ключ (MasterSecret), используя в качестве параметров последовательность Pre_MasterSecret, посланную ранее серверу случайную последовательность RAND_CL и полученную от него случайную последовательность RAND_SERV;

- используя значение MasterSecret, вычисляет криптографические параметры SSL-сессии: формирует общие с сервером сеансовые секретные ключи для симметричного шифрования и вычисления хэш-функций;

- переходит в режим защищенного взаимодействия.

4. Сервер, используя свой секретный ключ, расшифровывает полученное значение Pre_MasterSecret и выполняет те же операции, что и клиент:

- с помощью согласованной с клиентом хэш-функции формирует общий секретный мастер-ключ (MasterSecret), используя в качестве

параметров значение Pre_MasterSecret, а также посланную клиенту случайную последовательность RAND_SERV и полученную от него случайную последовательность RAND_CL;

- используя значение MasterSecret, вычисляет криптографические параметры SSL-сессии: формирует общие с клиентом сеансовые секретные ключи для симметричного шифрования и вычисления хэш-функций;

- переходит в режим защищенного взаимодействия.

Так как при формировании параметров SSL-сессии и клиент, и сервер пользовались одними и теми же исходными данными (согласованными алгоритмами, общей секретной последовательностью Pre_MasterSecret и случайными последовательностями RAND_CL и RAND_SERV), то очевидно, что в результате описанных выше действий они выработали одинаковые сеансовые секретные ключи. Для проверки идентичности параметров SSL-сессии клиент и сервер посылают друг другу тестовые сообщения, содержание которых известно каждой из сторон:

- клиент формирует сообщение из собственных посылок в адрес сервера на шаге 1 и посылок, полученных от сервера на шаге 2, внося элемент случайности в виде последовательности MasterSecret, уникальной для данной сессии; формирует код для проверки целостности сообщения (MAC), шифрует сообщение на общем сеансовом секретном ключе и отправляет серверу;

- сервер, в свою очередь, формирует сообщение из собственных посылок в адрес клиента на шаге 2, посылок, полученных от клиента на шаге 1, и последовательности MasterSecret; формирует код для проверки целостности сообщения (MAC), шифрует сообщение на общем сеансовом секретном ключе и отправляет клиенту;

- в случае успешной расшифровки и проверки каждой из сторон целостности полученных тестовых сообщений SSL-сессия считается установленной, и стороны переходят в штатный режим защищенного взаимодействия.

Необязательная вторая фаза рукопожатия позволяет аутентифицировать клиента. Она заключается в том, что сервер шлет запрос клиенту, клиент аутентифицирует себя, возвращая подписанное сообщение (запрос сервера) и свой цифровой сертификат.

В процессе защищенного взаимодействия с установленными криптографическими параметрами SSL-сессии выполняются следующие действия:

- каждая сторона при передаче сообщения формирует имитовставку (MAC) для последующей проверки целостности сообщения и затем зашифровывает исходное сообщение вместе с MAC по сеансовому секретному ключу;

- каждая сторона при приеме сообщения расшифровывает его и проверяет на целостность (вычисляется текущее значение MAC и сверяется со значением, полученным вместе с сообщением); в случае обнаружения нарушения целостности сообщения, SSL-сессия закрывается.

Протоколы SSL и TLS получили широкое распространение, прежде всего благодаря их использованию для защиты трафика, передаваемого по протоколу HTTP в сети Интернет. В то же время предоставляемые SSL услуги не являются прозрачными для приложений, т. е. сетевые приложения, которые хотят воспользоваться возможностями SSL, должны включать в себя реализацию протокола (или подключать ее в виде каких-то внешних модулей).

3.3. ПРОТОКОЛЫ IPSEC И РАСПРЕДЕЛЕНИЕ КЛЮЧЕЙ

Протокол IPSec или, если точнее, набор протоколов, разработан организацией IETF как базовый протокол обеспечения безопасности на уровне IP-соединения. Он является дополнением к использующемуся сейчас протоколу IP ver.4 и составной частью IP ver.6. Возможности, предоставляемые протоколами IPSec:

- контроль доступа;
- контроль целостности данных;
- аутентификация данных;

- защита от повторений;
- обеспечение конфиденциальности.

Основная задача IPSec — создание между двумя компьютерами, связанными через общедоступную (небезопасную) IP-сеть, безопасного туннеля (рис. 3.2), по которому передаются конфиденциальные или чувствительные к несанкционированному изменению данные. Подобный туннель создается с использованием криптографических методов защиты информации. Протокол работает на сетевом уровне модели OSI, и, соответственно, он «прозрачен» для приложений. Иными словами, на работу приложений (таких как web-сервер, браузер, СУБД и т. д.) не влияет, используется ли защита передаваемых данных с помощью IPSec или нет.

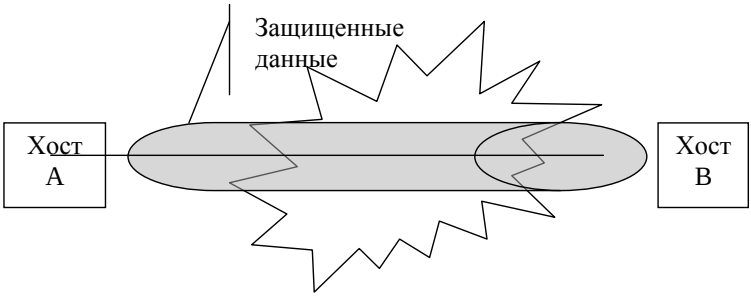


Рис. 3.2. Туннель безопасности

Архитектура IPSec является открытой, что позволяет использовать для защиты передаваемых данных новые криптографические алгоритмы, например, соответствующие национальным стандартам. Для этого необходимо, чтобы взаимодействующие стороны поддерживали эти алгоритмы, и они были бы стандартным образом зарегистрированы в описании параметров соединения.

Процесс защищенной передачи данных регулируется правилами безопасности, принятыми в системе. Параметры создаваемого туннеля описывает информационная структура, называемая контекст защиты или ассоциация безопасности (от англ. «Security Association», сокр.

SA). Как уже отмечалось выше, IPSec является набором протоколов, и состав контекста защиты может различаться. В зависимости от конкретного протокола в него входит:

- IP-адрес получателя;

- указание на протоколы безопасности, используемые при передаче данных;

- ключи, необходимые для шифрования и формирования имитовставки (если это требуется);

- указание на метод форматирования, определяющий, каким образом создаются заголовки;

- индекс параметров защиты (от англ. «Security Parameter Index», сокр. SPI) — идентификатор, позволяющий найти нужный SA.

Обычно контекст защиты является однонаправленным, а для передачи данных по туннелю в обе стороны задействуются два SA. Каждый хост имеет свою базу контекстов защиты, из которой выбирается нужный элемент либо на основании значения SPI, либо по IP-адресу получателя.

Два протокола, входящие в состав IPSec, это:

1) протокол аутентифицирующего заголовка — AH (от англ. «Authentication Header»), обеспечивающий проверку целостности и аутентификацию передаваемых данных; последняя версия протокола описана в документе RFC 4302 (предыдущие — RFC 1826, 2402);

2) протокол инкапсулирующей защиты данных — ESP (от англ. «Encapsulating Security Payload»), обеспечивающий конфиденциальность и, опционально, проверку целостности и аутентификацию; описан в RFC 4303 (предыдущие версии — RFC 1827, 2406).

Оба эти протокола имеют два режима работы — транспортный и туннельный, последний определен в качестве основного. Туннельный режим используется, если хотя бы один из соединяющихся узлов является шлюзом безопасности. В этом случае создается новый IP-заголовок, а исходный IP-пакет полностью инкапсулируется в новый.

Транспортный режим ориентирован на соединение хост-хост. При использовании ESP в транспортном режиме защищаются только данные IP-пакета, заголовок не затрагивается. При использовании AH защита распространяется на данные и часть полей заголовка. Более подробно режимы работы описаны ниже.

3.3.1. Протокол AH

В IP ver.4 аутентифицирующий заголовок располагается после IP-заголовка. Представим исходный IP-пакет как совокупность IP-заголовка, заголовка протокола следующего уровня (как правило, это TCP или UDP, на рис. 3.3 он обозначен как ULP — от англ. «Upper-Level Protocol») и данных.

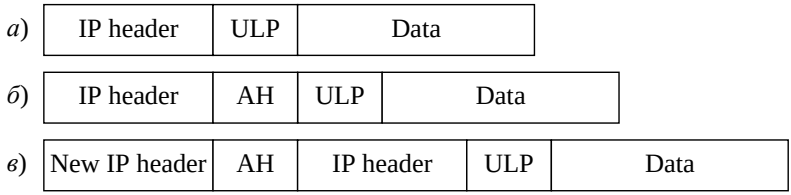


Рис. 3.3. *а*) исходный IP-пакет, *б*) IP-пакет при использовании AH в транспортном режиме, *в*) IP-пакет при использовании AH в туннельном режиме

На рис. 3.3 представлен исходный пакет и варианты его изменения при использовании протокола AH в разных режимах. В транспортном режиме заголовок исходного IP-пакета остается на своем месте, но в нем модифицируются некоторые поля. Например, меняется поле Next Header, указывающее на то, заголовок какого протокола следует за IP-заголовком. В туннельном режиме создается новый IP-заголовок, после которого идет заголовок AH, а за ним — полностью исходный IP-пакет.

Аутентификация производится путем создания имитовставки (MAC) для чего используется хэш-функция и секретный ключ. Во всех реализациях AH обязательно должно поддерживаться использо-

вание алгоритмов HMAC-MD5-96 (используется по умолчанию) и HMAC-SHA-1-96, представляющих собой хэш-функции с ключом, основанные на хэш-функциях MD5 и SHA-1, соответственно. Но могут использоваться и другие, «факультативные» алгоритмы хэширования. Полученное значение, называемое в описании протокола ICV (от англ. «Integrity Check Value» — значение контроля целостности), помещается в поле *Authentication Data* (рис. 3.4). Это поле переменной длины, так как разные алгоритмы хеширования формируют разные по длине дайджесты.

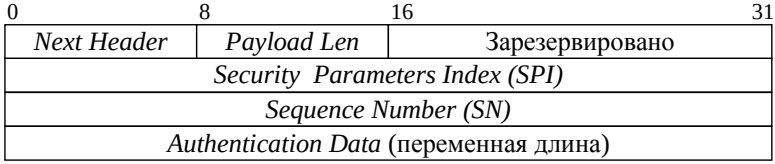


Рис. 3.4. Структура заголовка протокола AH

При использовании AH в транспортном режиме ICV рассчитывается для ULP, данных и неизменяемых полей IP-заголовка. Изменяемые поля, такие как поле TTL, указывающее на время жизни пакета и изменяемое при прохождении маршрутизаторов, при расчете значения хэш-функции принимаются равными 0. В туннельном режиме аутентифицируется весь исходный IP-пакет и неизменяемые поля нового заголовка. Рассмотрим формат заголовка AH (рис. 3.4).

Первые 8 бит заголовка (поле *Next Header*) содержат номер, соответствующий протоколу следующего уровня. Номер для каждого протокола назначает организация IANA (Internet Assigned Numbers Authority). Например, номер TCP — 6, ESP — 50, AH — 51 и т. д.

Поле *Payload Len* указывает длину заголовка AH в 32-битных словах. Далее 16 бит зарезервировано.

Поле *SPI* содержит значение индекса параметров защиты, по которому получатель сможет найти нужный контекст защиты (SA).

Поле *Sequence Number* было введено в RFC 2402. Значение счетчика, содержащееся в этом поле, может использоваться для защиты от атак путем повторных посылок перехваченных пакетов. Если функция защиты от повторов активирована (а это указывается в SA), отправитель последовательно наращивает значение поля для каждого пакета, передаваемого в рамках данной ассоциации (соединения, использующего единый SA).

Поле *Authentication Data*, как уже указывалось ранее, хранит значение ICV.

3.3.2. Протокол ESP

Если AH обеспечивает защиту от угроз целостности передаваемых данных, то ESP также может обеспечивать и конфиденциальность.

Так же как и AH, ESP может работать в транспортном и туннельном режимах. На рис. 3.5 изображены варианты его использования (штриховкой выделены фрагменты пакета, которые защищаются с помощью шифрования). Для ESP определен следующий перечень обязательных алгоритмов, которые должны поддерживаться во всех реализациях:

- для формирования имитовставки — HMAC-MD5-96 (используется по умолчанию) и HMAC-SHA-1-96;
- для шифрования — DES (в режиме CBC; используется по умолчанию) и NULL (отсутствие шифрования).

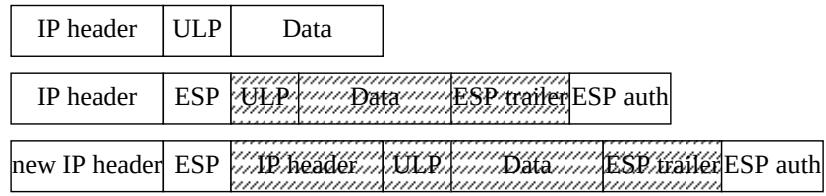


Рис. 3.5. *а*) исходный IP-пакет, *б*) ESP в транспортном режиме, *в*) ESP в туннельном режиме

Кроме того, зарегистрированы и могут быть реализованы еще ряд алгоритмов шифрования — Triple DES, CAST-128, RC5, IDEA, Blowfish, ARCFour (общедоступная версия RC4) [13].

Рассмотрим формат заголовка ESP (рис. 3.6). Он начинается с двух 32-разрядных значений — *SPI* и *SN*. Роль их такая же, как в протоколе AH — *SPI* идентифицирует контекст защиты, использующийся для создания данного туннеля; *SN* позволяет защититься от повторов пакетов. *SN* и *SPI* не шифруются. Следующим идет поле, содержащее зашифрованные данные. После них — поле заполнителя, который нужен для того, чтобы выровнять длину шифруемых полей до значения кратного размеру блока алгоритма шифрования.

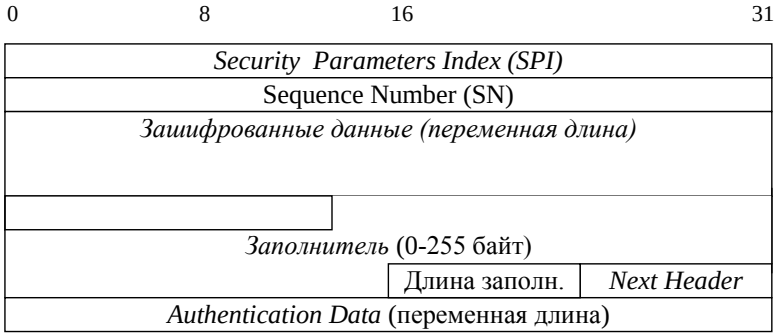


Рис. 3.6. Структура заголовка ESP

После заполнителя идут поля, содержащие значение длины заполнителя и указание на протокол более высокого уровня. Четыре перечисленных поля (данные, заполнитель, длина, следующий протокол) защищаются шифрованием.

Если ESP используется и для аутентификации данных, то завершает пакет поле переменной длины, содержащее ICV. В отличие от AH, в ESP при расчете значения имитовставки, поля IP-заголовка (нового — для туннельного режима, модифицированного старого — для транспортного) не учитываются.

При совместном использовании протоколов AH и ESP, после IP-заголовка идет AH, после него — ESP. В этом случае ESP решает задачи обеспечения конфиденциальности, AH — обеспечения целостности и аутентификации источника соединения.

Рассмотрим ряд дополнительных вопросов, связанных с использованием IPSec. Начнем с того, откуда берется информация о параметрах соединения — контекстах защиты или SA. Создание базы SA может производиться различными путями. В частности, она может создаваться администратором безопасности вручную или формироваться с использованием специальных протоколов — SKIP, ISAKMP (Internet Security Association and Key Management Protocol) и IKE (Internet Key Exchange).

3.3.3. Протокол SKIP

Протокол SKIP (Simple Key management for Internet Protocol) был разработан корпорацией SUN Microsystems в 1994 году. Он создавался как IP-совместимый протокол, обеспечивающий управление ключами и криптозащиту передаваемых данных на сетевом уровне модели OSI [13]. По нумерации IANA этому протоколу присвоен номер 57.

Первоначально SKIP выглядел следующим образом. Устанавливающие защищенное взаимодействие абоненты должны иметь аутентифицированные открытые ключи. По алгоритму Диффи–Хеллмана они вычисляют общий секретный ключ K_{ij}. Он используется для защиты ключевой информации.

Отправитель генерирует временный ключ K_p, используемый для шифрования данных одного пакета или небольшой их группы. На нем зашифровывается исходный IP-пакет и инкапсулируется в SKIP-пакет (рис. 3.7). Шифрование данных на временном ключе, а не на общем секретном ключе K_{ij}, повышает надежность, так как в этом случае нарушителю труднее набрать нужный для реализации атаки на K_{ij} объем шифрованных данных.

а)

IP-заголовок	Данные

SKIP-заголовок

б)

Новый IP- заголовок	K_p зашифр. на K_{ij}; имитовст.	Исходный пакет, зашифр. на K_p

Рис. 3.7. Пакет до (*а*) и после (*б*) применения протокола SKIP

Ключ K_p зашифровывается на ключе K_{ij}, и криптограмма помещается в SKIP-заголовок, там же резервируется место под имитовставку.

Формируется новый IP-заголовок, и с помощью хэш-функции с ключом для всего пакета рассчитывается значение имитовставки, которое помещается в SKIP-заголовок.

Впоследствии протокол SKIP был усовершенствован. В частности, были внесены изменения, позволяющие использовать SKIP совместно с ESP. Тогда SKIP отвечает за передачу ключевой информации и описание параметров соединения, а ESP решает задачи криптографической защиты данных. Рассмотрим теперь эту версию [14].

Итак, стороны распределили общий секретный ключ K_{ij}, и отправитель сгенерировал временный ключ K_p. На его основе с помощью хэш-функции H будут выработаны ключ для шифрования пакета E_K_p и ключ для аутентификации A_K_p. В отличие от первоначального варианта, при передаче ключ K_p шифруется не на K_{ij}, а на модифицированном при помощи счетчика n и хеш-фукнции ключе K_{ijn}. Расчет производится в соответствии с формулой:

$$\begin{aligned} E\_Kp &= H(K_p|\text{Crypt Alg}|02h)|H(K_p|\text{Crypt Alg}|00h), \\ A\_Kp &= H(K_p|\text{MAC Alg}|03h)|H(K_p|\text{MAC Alg}|01h), \\ K_{ijn} &= H(K'_{ij}|n|01h)|H(K'_{ij}|n|00h), \end{aligned} \qquad (3.1)$$

где K'_{ij} — младшие 256 бит K_{ij}, Crypt Alg и MAC Alg — значения соответствующих полей заголовка SKIP (рис. 3.8).

Рассмотрим поля заголовка (рис. 3.8). *Ver* — номер версии протокола. Следующие за ним 4 бита зарезервированы (*Rsvd*). Далее — идентификаторы пространств имен источника и получателя *Source*

NSID и Dest NSID. Если они равны 0, то в полях *Source MKID* и *Dest MKID* ставятся IP-адреса источника и получателя соответственно. После поля *Dest NSID* идет поле *Next Header*, содержащее номер протокола, следующего за SKIP. Далее идет 32-разрядное поле счетчика *Counter n*. Как отмечается в описаниях, правила для работы со счетчиком *n* отнесены на усмотрение разработчика, но для обеспечения совместимости версий предлагается считать, что *n* — время в часах, отсчитанное от 00:00 01.01.95. Как правило, если значение счетчика *n* пришедшего пакета отличается более чем на 1 от текущего, то пакет отбрасывается.

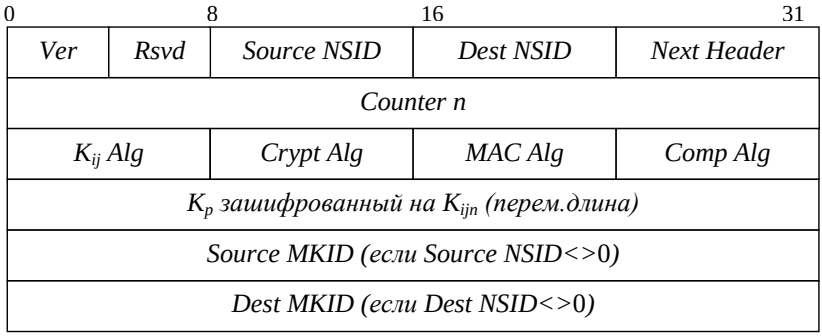


Рис. 3.8. Формат заголовка SKIP

Далее в заголовке идут байтовые идентификаторы алгоритмов: шифрования ключа K_p — *K_{ij} Alg*, шифрования данных в пакете — *Crypt Alg*, аутентификации данных — *MAC Alg*, сжатия (если используется) — *Comp Alg*. После идентификаторов в SKIP-заголовок помещается ключ K_p, зашифрованный на ключе K_{ijn} (размер этого поля зависит от используемых алгоритмов шифрования ключа и данных). Далее идут идентификаторы отправителя и получателя в выбранном пространстве имен — *Source MKID* и *Dest MKID*. Наличие нескольких идентификаторов позволяет более гибко настраивать использование протоколов безопасности. Например, если на одном компьютере работают различные приложения, можно описать политику, указываю-

щую, какие алгоритмы и ключи использовать для защиты данных каждого из них.

В случае совместного использования протоколов SKIP и ESP заголовок SKIP размещается после IP-заголовка перед заголовком ESP (рис. 3.9).

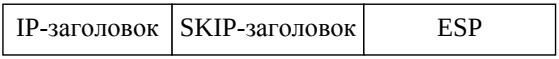


Рис. 3.9. Совместное использование SKIP и ESP

В этом случае протокол ESP использует параметры соединения, определяемые протоколом SKIP, а указывает на это значение SPI в заголовке ESP: SKIP_SPI=1.

3.3.4. Протоколы ISAKMP и IKE

Протокол ISAKMP (Internet Security Association Key Management Protocol — протокол управления ключами и контекстами безопасности в Internet) был разработан IETF для решения задач согласования параметров и управления ключами при формировании защищенного канала. ISAKMP описывает общую схему взаимодействия, но не содержит конкретных криптоалгоритмов распределения ключей. Поэтому он используется совместно с протоколом OAKLEY, основанном на алгоритме Диффи–Хеллмана [13]. Протокол IKE (англ. «Internet Key Exchange» — обмен ключами в Internet) определяет совместное использование протоколов ISAKMP с OAKLEY и SKEMI (англ. «Secure Key Exchange Mechanism for Internet» — безопасный механизм обмена ключами в Интернет) для решения данной задачи. Сравнивая протоколы SKIP и IKE, надо отметить, что последний более сложен в реализации, но считается более надежным и гибким.

Протокол ISAKMP определяет две фазы согласования параметров взаимодействия:

- согласование глобальных параметров защищенного канала (в терминах IPSec такие параметры называются управляющим контекстом);

- согласование параметров каждого защищенного соединения (они образуют контекст защиты — SA).

С точки зрения модели стека TCP/IP, протокол ISAKMP является протоколом прикладного уровня. В случае его использования совместно с IPSec спецификация устанавливает использование в качестве протокола нижнего уровня UDP с номером порта 500.

Протокол ISAKMP определяет следующую последовательность действий по формированию управляющего контекста (стороны взаимодействия, например, это могут быть два шлюза безопасности, называются *«Инициатор»* и *«Партнер»*):

1) *«Инициатор»* → *«Партнер»*: заявка на контекст, включающая предлагаемые алгоритмы и их параметры;

2) *«Партнер»* → *«Инициатор»*: принимаемые алгоритмы и параметры (из списка, полученного на шаге 1; для каждой функции защиты — генерация и распределение ключей, шифрование, аутентификация — используется один алгоритм и его параметры);

3) *«Инициатор»* → *«Партнер»*: ключевой материал и одноразовый номер инициатора;

4) *«Партнер»* → *«Инициатор»*: ключевой материал и одноразовый номер партнера;

5) *«Инициатор»* → *«Партнер»*: подписанное инициатором зашифрованное сообщение, содержащее его идентификатор;

6) *«Партнер»* → *«Инициатор»*: подписанное партнером зашифрованное сообщение, содержащее его идентификатор.

Если используется протокол OAKLEY, на шаге 3 и 4 стороны отправляют друг другу свои открытые ключи вместе со случайными числами, служащими для защиты от повтора. Для обеспечения контроля подлинности открытых ключей могут быть использованы сертификаты X.509. В соответствии со схемой Диффи–Хеллмана рассчи-

тывается общий секретный ключ. На его основе вырабатываются значения:

SKEYID_d — ключевой материал, используемый для генерации временных ключей для защищенных соединений;

SKEYID_a — сеансовые ключи, используемые для аутентификации сторон и согласовываемых параметров;

SKEYID_e — сеансовые ключи, используемые для шифрования согласовываемых параметров.

Пятый и шестой шаги служат для обмена идентификационной информацией, защищенной и заверенной на ключах *SKEYID_e* и *SKEYID_a*.

Такой порядок взаимодействия реализуется при использовании основного или базового режима (англ. «Main Mode»). Он более медленный, но и более безопасный. Существует также «агрессивный» режим (англ. «Aggressive Mode») при котором для увеличения скорости взаимодействия ряд параметров передается в открытом виде, и уменьшено число шагов взаимодействия (с 6 до 3 за счет того, что на первом и втором шаге сразу передается больше параметров) [11].

Сообщение ISAKMP состоит из заголовка и следующих за ним полей данных. Формат заголовка приведен на рис. 3.10.

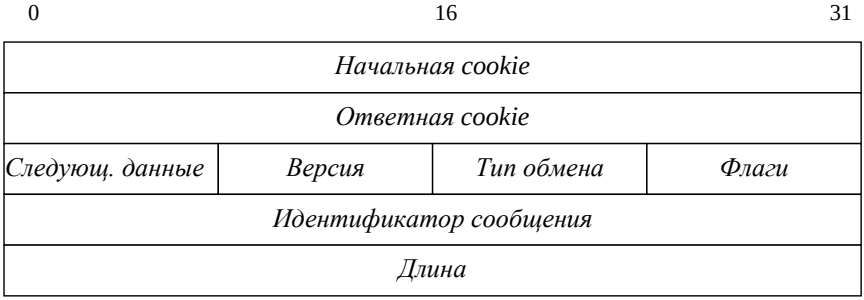


Рис. 3.10. Заголовок ISAKMP

Исходя из значения текущего времени, стороны формируют идентификаторы (*cookie*), которые включают в заголовок пакета (на

шаге 1 присутствует только идентификатор стороны-инициатора соединения, на последующих — идентификаторы обеих сторон). Присутствие этих идентификаторов позволяет защититься от атак путем повторных посылок перехваченных сообщений.

Поле «*Следующие данные*» содержит идентификатор, указывающий на тип содержимого области данных. Например, 1 — контекст защиты (SA), 2 — предложение (используется при согласовании параметров); 6 — сертификат, 7 — запрос сертификата и т. д.

Тип обмена указывает на тип информационного обмена (режим, в котором работает протокол). Например, 0 — нет обмена, 1 — базовый, 4 — агрессивный и т. д.

Флаги позволяют указать дополнительные настройки. Например, один из флагов указывает на то, что все данные, идущие за заголовком, зашифрованы.

Идентификатор сообщения — уникальный идентификатор сообщения.

Длина — длина сообщения (заголовка и данных).

Итак, описанная фаза протокола позволяет сформировать общий защищенный канал и согласовать его параметры.

После создания общего защищенного канала параметры каждого защищенного соединения, создаваемого в рамках этого канала, согласуются на основе сформированных глобальных параметров канала и образуют контекст защиты. Каждое защищенное соединение является однонаправленным, и в нем может использоваться один из двух протоколов — ESP или AH (кроме того, на базе общего защищенного канала могут быть созданы защищенные соединения, использующие отличный от IPSec протокол). Если предполагается организовать защищаемое ESP двустороннее взаимодействие, понадобятся два соединения (и два SA), если использовать и протокол AH, и ESP, то нужны четыре SA.

В состав согласуемых параметров, образующих SA, входят [13]:

- номер протокола криптозащиты (AH, ESP, другой);

- номера алгоритмов криптозащиты и их параметры;

- режим протокола (транспортный или туннельный);

- сеансовые ключи (действующие для текущего соединения);

- срок существования защищенного соединения (может задаваться временем или объемом переданного трафика; например, срок существования канала — 6 часов, соединения в рамках этого канала — 1 час);

- максимальный размер пакетов;

- дополнительные параметры (параметры счетчика и т. д.) для защиты от повтора, задержки, удаления пакетов сообщения.

Пользователями защищенных соединений, как правило, являются прикладные процессы. И между двумя узлами сети может существовать произвольное число соединений, сформированных в рамках одного защищенного канала.

Защищенное соединение, соответствующее спецификациям протокола IPSec, идентифицируется целевым IP-адресом, используемым протоколом криптозащиты (ESP или AH) и индексом SPI.

Для выработки параметров «*Инициатор*» и «*Партнер*» обмениваются следующими сообщениями.

1) «*Инициатор*» → «*Партнер*» (защищенное сообщение):

- заявка на создание защищенного соединения (предлагаемые алгоритмы и их параметры);

- одноразовый номер инициатора.

2) «*Партнер*» → «*Инициатор*» (защищенное сообщение):

- принимаемые алгоритмы и параметры;

- одноразовый номер партнера.

3) «*Инициатор*» → «*Партнер*» (защищенное сообщение):

- одноразовый номер инициатора;

- одноразовый номер партнера.

Для защиты передаваемых сообщений используются ключи, выработанные при формировании защищенного канала: *SKEYID_a* — для аутентификации, *SKEYID_e* — для шифрования. Для аутентифи-

кации используется хэш-функция, в качестве аргументов которой выступают аутентифицируемое сообщение и ключ *SKEYID_a*.

Временные ключи защищенного соединения генерируются путем применения хэш-функции к значению *SKEYID_d* с дополнительными параметрами, в число которых входят одноразовые идентификаторы инициатора и партнера.

Принимающая сторона соединения задает для формируемого SA номер SPI, по которому он будет идентифицироваться.

3.3.5. Протоколы IPSec и трансляция сетевых адресов

При подключении сетей организаций к Интернет часто используется механизм трансляции сетевых адресов — NAT (от англ. «Network Address Translation»). Это позволяет уменьшить число зарегистрированных IP-адресов, используемых в данной сети. Внутри сети используются незарегистрированные «частные» адреса (как правило, из диапазонов, специально выделенных для этой цели, например, адреса вида 192.168.x.y для сетей класса C). Если пакет из такой сети передается в Интернет, то маршрутизатор, внешнему интерфейсу которого назначен по крайней мере один зарегистрированный ip-адрес, модифицирует ip-заголовки сетевых пакетов, подставляя вместо частных адресов зарегистрированный адрес. Порядок, по которому производится подстановка, описывается в специальной таблице. При получении ответа в соответствии с таблицей делается обратная замена, и пакет переправляется во внутреннюю сеть.

Рассмотрим пример использования NAT (рис. 3.11). В данном случае во внутренней сети используются частные адреса 192.168.0.x. С компьютера с адресом 192.168.0.2 обращаются во внешнюю сеть к компьютеру с адресом 195.242.2.2. Пусть это будет подключение к web-серверу (протокол HTTP, который использует TCP порт 80).

При прохождении пакета через маршрутизатор, выполняющий трансляцию адресов, ip-адрес отправителя (192.168.0.2) будет заменен на адрес внешнего интерфейса маршрутизатора (195.201.82.146), а в

таблицу трансляции адресов будет добавлена запись, аналогичная приведенной в табл. 3.1.

Получив представление о механизме работы NAT, разберемся, какие сложности могут возникнуть в случае использования IPSec.

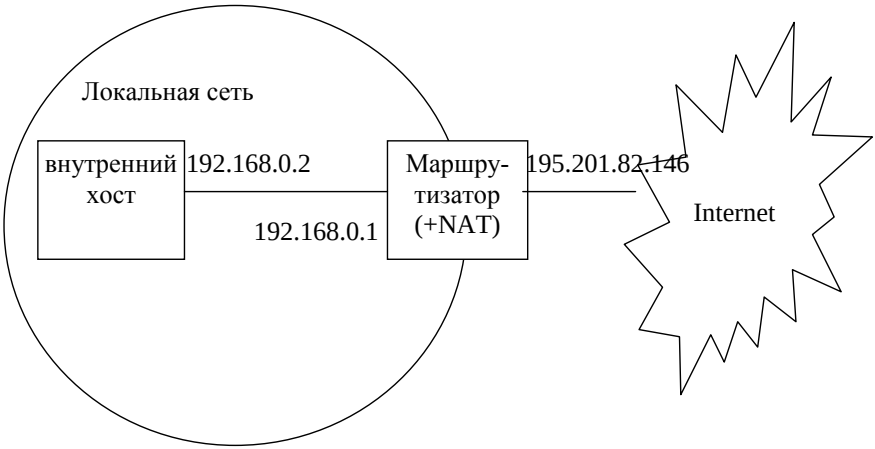


Рис. 3.11. Пример использования механизма NAT

Таблица 3.1

Таблица трансляции адресов

Протокол	Внутренний локальный ip-адрес: порт	Внутренний зарегистрированный ip-адрес: порт.	Внешний ip-адрес: порт
TCP	192.168.0.2: 2001	195.201.82.146: 2161	195.242.2.2 : 80

Предположим, с хоста с адресом 192.168.0.2 пытаются установить защищенное соединение с внешним хостом 195.242.2.2, используя протокол аутентифицирующего заголовка (AH). При прохождении маршрутизатора ip-адрес отправителя меняется, как было описано выше. Протокол AH определяет, что значение имитовставки рассчитывается, включая неизменяемые поля IP-заголовка, в частности, адрес отправителя. Сторона-получатель, обнаружит неверное (из-за трансляции адресов) значение имитовставки и отбросит пакет.

Таким образом, механизм NAT и протокол AH несовместимы. В то же время протокол ESP, который не контролирует целостность ip-заголовка, может использоваться совместно с трансляцией адресов.

Кроме того, RFC 2709 определяет расширение NAT — IPC-NAT (англ. «IPSec policy Controlled NAT» — NAT, управляемый правилами IPSec). Оно позволяет решить указанную проблему путем создания IP-IP туннеля, одной из конечных точек которого является узел NAT. В этом случае вместо модификации IP-адреса отправителя в заголовке исходного пакета NAT-устройство помещает без изменений весь исходный пакет (который аутентифицирован AH) в новый IP-пакет, в заголовке которого в качестве адреса отправителя ставится адрес NAT-устройства. На стороне получателя из полученного пакета изымают исходный пакет и далее обрабатывают его как обычно.

Отдельно необходимо решать вопрос с распределением ключей. Если для этой цели используется протокол IKE (а он использует транспортный протокол UDP, порт 500), потребуется специально организовать пересылку соответствующих данных во внутреннюю сеть. В случае если задействовать UDP-порт 500 не представляется возможным, можно использовать описываемый в документах RFC 3947, 3948 механизм NAT-T (от англ. «NAT traversal»), определяющий инкапсуляцию IKE и IPSec трафика в пакеты UDP. При этом задействуется порт 4500.

3.4. МЕЖСЕТЕВЫЕ ЭКРАНЫ

Межсетевой экран (МЭ) — это средство защиты информации, осуществляющее анализ и фильтрацию проходящих через него сетевых пакетов. В зависимости от установленных правил МЭ пропускает или уничтожает пакеты, разрешая или запрещая таким образом сетевые соединения. МЭ является классическим средством защиты периметра компьютерной сети: он устанавливается на границе между внутренней (защищаемой) и внешней (потенциально опасной) сетями и контролирует соединения между узлами этих сетей. Но бывают и другие схемы подключения, которые будут рассмотрены ниже.

Английский термин, используемый для обозначения МЭ — «firewall». Поэтому в литературе межсетевые экраны иногда также называют файервол или брандмауэр (немецкий термин, аналог firewall).

Как уже было отмечено, фильтрация производится на основании правил. Наиболее безопасным при формировании правил для МЭ считается подход «запрещено все, что явно не разрешено». В этом случае сетевой пакет проверяется на соответствие разрешающим правилам, а если таковых не найдется — отбрасывается. Но в некоторых случаях применяется и обратный принцип: «разрешено все, что явно не запрещено». Тогда проверка производится на соответствие запрещающим правилам, и если таких не будет найдено, пакет будет пропущен.

Фильтрацию можно производить на разных уровнях эталонной модели сетевого взаимодействия OSI. По этому признаку МЭ делятся на следующие классы [13]:

- экранирующий маршрутизатор;
- экранирующий транспорт (шлюз сеансового уровня);
- экранирующий шлюз (шлюз прикладного уровня).

Экранирующий маршрутизатор (или пакетный фильтр) функционирует на сетевом уровне модели OSI, но для выполнения проверок может использовать информацию и из заголовков протоколов транспортного уровня. Соответственно, фильтрация может производиться по ip-адресам отправителя и получателя, а также по TCP и UDP портам. Такие МЭ отличает высокая производительность и относительная простота — функциональностью пакетных фильтров обладают сейчас даже наиболее простые и недорогие аппаратные маршрутизаторы. В то же время, они не защищают от многих атак, например, связанных с подменой участников соединений.

Шлюз сеансового уровня работает на сеансовом уровне модели OSI и также может контролировать информацию сетевого и транспортного уровней. Соответственно, в дополнение к перечисленным

выше возможностям подобный МЭ может контролировать процесс установки соединения и проводить проверку проходящих пакетов на принадлежность разрешенным соединениям.

Шлюз прикладного уровня может анализировать пакеты на всех уровнях модели OSI от сетевого до прикладного, что обеспечивает наиболее высокий уровень защиты. В дополнение к ранее перечисленным появляются такие возможности, как аутентификация пользователей, анализ команд протоколов прикладного уровня, проверка передаваемых данных (на наличие компьютерных вирусов, соответствие политике безопасности) и т. д.

Рассмотрим теперь вопросы, связанные с установкой МЭ. На рис. 3.12 представлены типовые схемы подключения МЭ. В первом случае (рис. 3.12, *а*) МЭ устанавливается после маршрутизатора и защищает всю внутреннюю сеть. Такая схема применяется, если требования в области защиты от несанкционированного межсетевого доступа примерно одинаковы для всех узлов внутренней сети. Например, «разрешать соединения, устанавливаемые из внутренней сети во внешнюю, и пресекать попытки подключения из внешней сети во внутреннюю».

В случае если требования для разных узлов различны (например, нужно разместить почтовый сервер, к которому могут подключаться «извне»), подобная схема установки межсетевого экрана не является достаточно безопасной. Если в нашем примере нарушитель в результате реализации сетевой атаки получит контроль над указанным почтовым сервером, через него он может получить доступ и к другим узлам внутренней сети.

В подобных случаях иногда перед МЭ создается открытый сегмент сети предприятия (рис. 3.12, *б*), а МЭ защищает остальную внутреннюю сеть. Недостаток данной схемы заключается в том, что подключения к узлам открытого сегмента МЭ не контролирует.

Более предпочтительным в данном случае является использование МЭ с тремя сетевыми интерфейсами (рис. 3.12, *в*).

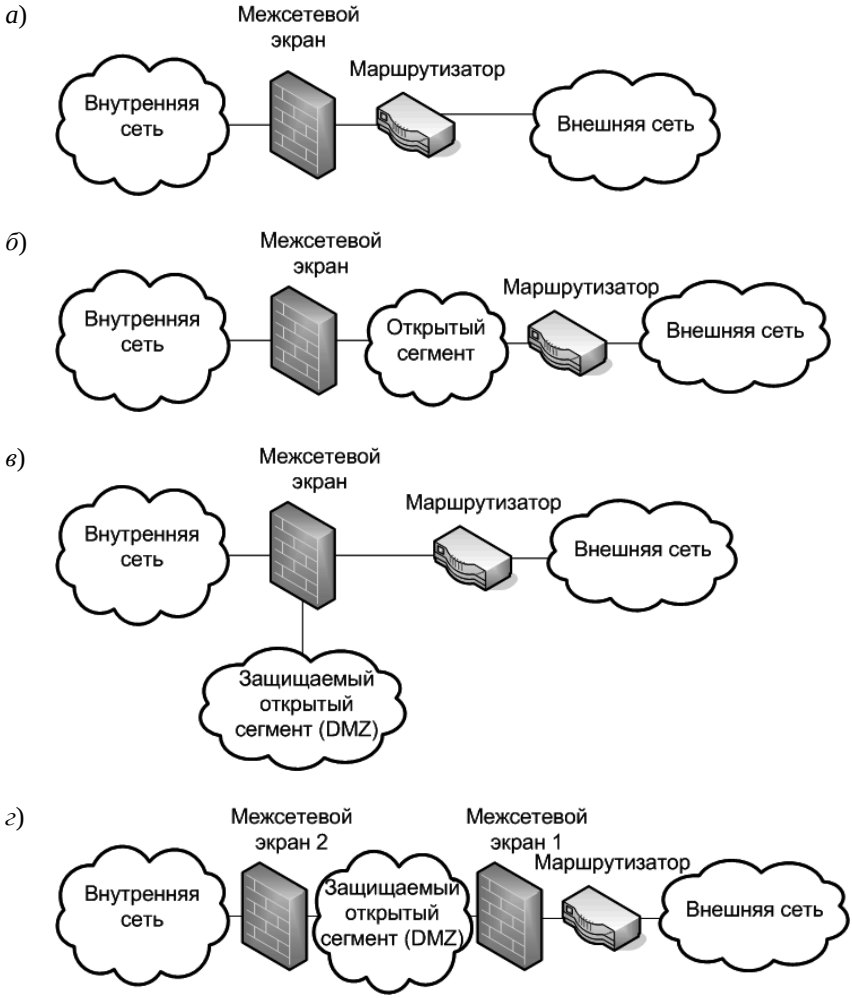


Рис. 3.12. Типовые схемы подключения межсетевых экранов: *а*) подключение МЭ с двумя сетевыми интерфейсами; *б*) подключение МЭ с двумя сетевыми интерфейсами при выделении открытого сегмента внутренней сети; *в*) подключение МЭ с тремя сетевыми интерфейсами; *г*) подключение двух МЭ

МЭ с тремя сетевыми интерфейсами конфигурируется таким образом, чтобы правила доступа во внутреннюю сеть были более

строгими, чем в открытый сегмент. В то же время, и те, и другие соединения могут контролироваться МЭ. Открытый сегмент в этом случае иногда называется «демилитаризованной зоной» — DMZ.

Еще более надежной считается схема, в которой для защиты сети с DMZ задействуются два независимо конфигурируемых МЭ (рис. 3.12, *г*). В этом случае МЭ 2 реализует более жесткий набор правил фильтрации по сравнению с МЭ1. И даже успешная атака на первый МЭ не сделает внутреннюю сеть беззащитной.

В последнее время стал широко использоваться вариант установки программного МЭ непосредственно на защищаемый компьютер. Иногда такой МЭ называют «персональным». Подобная схема позволяет защититься от угроз исходящих не только из внешней сети, но из внутренней. Особенно актуально применение персональных МЭ при непосредственном подключении компьютера к потенциально опасной сети. Например, при подключении домашнего компьютера к Интернет.

Завершая этот раздел, хочется отметить, что тема защиты информации в компьютерных сетях очень обширна и многие ее аспекты в данном разделе даже не затрагивались. Дополнительные сведения по данной тематике можно получить из специальной литературы, в частности, из изданий, перечисленных в библиографическом списке.

4. АНАЛИЗ И УПРАВЛЕНИЕ РИСКАМИ В СФЕРЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

4.1. ВВЕДЕНИЕ В ПРОБЛЕМУ

Целью данного раздела является изучение современных методик анализа и управления рисками, связанными с информационной безопасностью (ИБ).

Риском в сфере ИБ будем называть потенциальную возможность понести убытки из-за нарушения безопасности информационной системы (ИС). Зачастую понятие риска смешивают с понятием угрозы, которое рассматривалось ранее, в разделе 1.1. Необходимо

отметить, что от угрозы риск отличает наличие количественной оценки возможных потерь и (возможно) оценки вероятности наступления нежелательного события.

Разберемся, зачем нужно исследовать риски в сфере ИБ, и что это может дать при разработке системы обеспечения ИБ для ИС. Для любого проекта, требующего финансовых затрат на его реализацию, весьма желательно уже на начальной стадии определить, что мы будем считать признаком завершения работы и как будем оценивать результаты проекта. Для задач, связанных с обеспечением ИБ, это также весьма актуально.

На практике наибольшее распространение получили два подхода к обоснованию проекта подсистемы обеспечения безопасности.

Первый из них основан на проверке соответствия уровня защищенности ИС требованиям одного из стандартов в области информационной безопасности. Это может быть класс защищенности в соответствии с требованиями руководящих документов Гостехкомиссии РФ (сейчас это ФСТЭК России), профиль защиты, разработанный в соответствии со стандартом ISO-15408 (см. раздел 1.5), или какой-либо другой набор требований. Тогда критерий достижения цели в области безопасности — это выполнение заданного набора требований. Критерий эффективности — минимальные суммарные затраты на выполнение поставленных функциональных требований: $\sum c_i \to \min$, где c_i — затраты на i-е средство защиты.

Основной недостаток данного подхода заключается в том, что в случае, когда требуемый уровень защищенности жестко не задан (например, через законодательные требования), определить «наиболее эффективный» уровень защищенности ИС достаточно сложно.

Второй подход к построению системы обеспечения ИБ связан с оценкой и управлением рисками. Изначально он произошел из принципа «разумной достаточности», примененного к сфере обеспечения ИБ. Этот принцип быть описан следующим набором утверждений:

- абсолютно непреодолимую систему защиты создать невозможно;
- необходимо соблюдать баланс между затратами на защиту и получаемым эффектом, в т. ч. и экономическим, заключающимся в снижении потерь от нарушений безопасности;
- стоимость средств защиты не должна превышать стоимости защищаемой информации (или других ресурсов — аппаратных, программных);
- затраты нарушителя на несанкционированный доступ к информации должны превышать тот эффект, который он получит, осуществив подобный доступ.

Но вернемся к рискам. В данном случае, рассматривая ИС в ее исходном состоянии, мы оцениваем размер ожидаемых потерь от инцидентов, связанных с информационной безопасностью (как правило, берется определенный период времени, например, год). После этого делается оценка того, как предлагаемые средства и меры обеспечения безопасности влияют на снижение рисков, и сколько они стоят. Если представить некоторую идеальную ситуацию, то идею подхода отображает приведенный на рис. 4.1 график [15].

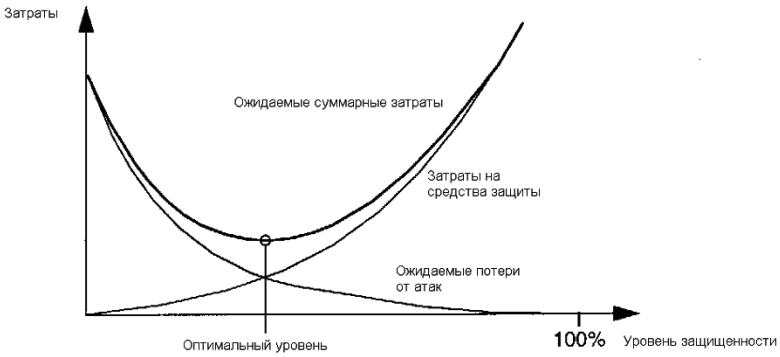


Рис. 4.1. Идеализированный график соотношения «затраты на защиту — ожидаемые потери»

По мере того, как затраты на защиту растут, размер ожидаемых потерь падает. Если обе функции имеют вид, представленный на рис. 4.1, то можно определить минимум функции «Ожидаемые суммарные результаты», который нам и требуется.

К сожалению, на практике точные зависимости между затратами и уровнем защищенности определить не представляется возможным, поэтому аналитический метод определения минимальных затрат в представленном виде неприменим.

Для того чтобы перейти к рассмотрению вопросов описания риска, введем еще одно определение. *Ресурсом* или *активом* будем называть именованный элемент ИС, имеющий (материальную) ценность и подлежащий защите.

Тогда риск может быть идентифицирован следующим набором параметров:

- угроза, с возможной реализацией которой связан данный риск;
- ресурс, в отношении которого может быть реализована угроза (ресурс может быть информационный, аппаратный, программный и т. д.);
- уязвимость, через которую может быть реализована данная угроза в отношении данного ресурса.

Важно также определить то, как мы узнаем, что нежелательное событие произошло. Поэтому в процессе описания рисков, обычно также указывают события-«*триггеры*», являющиеся идентификаторами рисков, произошедших или ожидающихся в скором времени (например, увеличение время отклика web-сервера может свидетельствовать о производимой на него одной из разновидностей атак на «отказ в обслуживании»).

Исходя из сказанного выше, в процессе оценки риска надо оценить стоимость ущерба и частоту возникновения нежелательных событий и вероятность того, что подобное событие нанесет урон ресурсу. Размер ущерба от реализации угрозы в отношении ресурса зависит от стоимости ресурса, который подвергается риску, и степени разру-

шительности воздействия на ресурс, выражаемой в виде коэффициента разрушительности. Как правило, указанный коэффициент лежит в диапазоне от 0 до 1. Таким образом, получаем оценку потери от разовой реализации угрозы, представимую в виде произведения:

$$Потери=(Стоим.\ Рес.)\times(Коэфф.\ Разруш.). \quad (4.1)$$

Далее необходимо оценить частоту возникновения рассматриваемого нежелательного события (за какой-то фиксированный период времени, например, за год) и вероятность успешной реализации угрозы. В результате, стоимость риска может быть вычислена по формуле:

$$Стоим.Риска=(Частота) \times (Вероятн.) \times (Стоим.\ Рес.) \times \times (Коэфф.\ Разруш). \quad (4.2)$$

Примерно такая формула используется во многих методиках анализа рисков, некоторые из которых будут рассмотрены в дальнейшем. Ожидаемый ущерб сравнивается с затратами на меры и средства защиты, после чего принимается решение в отношении данного риска. Он может быть:

- принят;
- снижен (например, за счет внедрения средств и механизмов защиты, уменьшающих вероятность реализации угрозы или коэффициент разрушительности);
- устранен (за счет отказа от использования подверженного угрозе ресурса);
- перенесен (например, застрахован, в результате чего в случае реализации угрозы безопасности потери будет нести страховая компания, а не владелец ИС).

4.2. УПРАВЛЕНИЕ РИСКАМИ. МОДЕЛЬ БЕЗОПАСНОСТИ С ПОЛНЫМ ПЕРЕКРЫТИЕМ

Идеи управления рисками во многом восходят к модели безопасности с полным перекрытием, разработанной в 1970-х годах [16].

Модель системы безопасности с полным перекрытием строится исходя из постулата, что система безопасности должна иметь, по

крайней мере, одно средство для обеспечения безопасности на каждом возможном пути воздействия нарушителя на ИС. В модели точно определяется каждая область, требующая защиты, оцениваются средства обеспечения безопасности с точки зрения их эффективности и их вклад в обеспечение безопасности во всей вычислительной системе.

Считается, что несанкционированный доступ к каждому из множества защищаемых объектов (ресурсов ИС) **O** сопряжен с некоторой «величиной ущерба» для владельца ИС, и этот ущерб может быть определен количественно.

С каждым объектом, требующим защиты, связывается некоторое множество действий, к которым может прибегнуть нарушитель для получения несанкционированного доступа к объекту. Потенциальные злоумышленные действия по отношению ко всем объектам $o \in \mathbf{O}$ формируют множество угроз ИБ **T**. Каждый элемент данного множества характеризуется вероятностью появления.

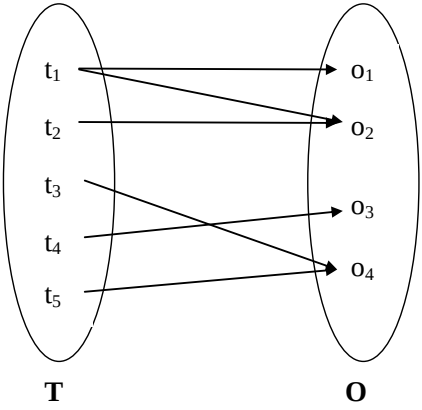


Рис. 4.2. Двудольный граф «угроза–объект»

Множество отношений «угроза–объект» образует двухдольный граф (рис. 4.2), в котором ребро (t_i,o_j) существует тогда и только тогда, когда угроза t_i является средством получения доступа к объекту o_j. Следует отметить, что связь между угрозами и объектами не явля-

ется связью типа «один к одному» — угроза может распространяться на любое число объектов, а объект может быть уязвим со стороны более чем одной угрозы. Цель защиты состоит в том, чтобы «перекрыть» каждое ребро данного графа и воздвигнуть барьер для доступа по этому пути.

Завершает модель третий набор, описывающий средства обеспечения безопасности **M**, которые используются для защиты информации в ИС. В идеальном случае каждое средство $m_k \in \mathbf{M}$ должно устранять некоторое ребро (t_i,o_j). В действительности m_k выполняет функцию «барьера», обеспечивая некоторую степень сопротивления попыткам проникновения. Это сопротивление — основная характеристика, присущая всем элементам набора **M**. Набор **M** средств обеспечения безопасности преобразует двудольный граф в трехдольный (рис. 4.3).

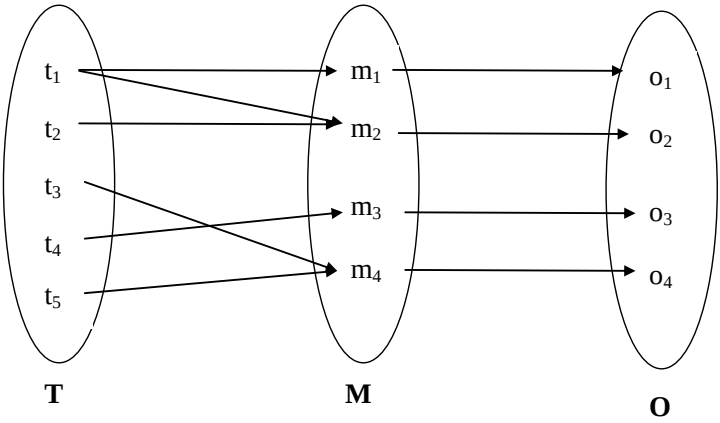


Рис. 4.3. Трехдольный граф
«угроза–средство безопасности–объект»

В защищенной системе все ребра представляются в форме (t_i,m_k) и (m_k,o_j). Любое ребро в форме (t_i,o_j) определяет незащищенный объект. Следует отметить, что одно и то же средство обеспечения безопасности может противостоять реализации более чем одной угрозы и

(или) защищать более одного объекта. Отсутствие ребра (t_i,o_j) не гарантирует полного обеспечения безопасности (хотя наличие такого ребра дает потенциальную возможность несанкционированного доступа за исключением случая, когда вероятность появления t_i равна нулю).

Далее в рассмотрение включается теоретико-множественная модель защищенной системы — система обеспечения безопасности Клементса. Она описывает систему в виде пятикортежного набора **S = {O,T,M,V,B}**, где **O** — набор защищаемых объектов; **T** — набор угроз; **M** — набор средств обеспечения безопасности; **V** — набор уязвимых мест — отображение **TxO** на набор упорядоченных пар $V_i = (t_i,o_j)$, представляющих собой пути проникновения в систему; **B** — набор барьеров — отображение **V**x**M** или **T**x**O**x**M** на набор упорядоченных троек $b_i = (t_i,o_j,m_k)$, представляющих собой точки, в которых требуется осуществлять защиту в системе.

Таким образом, система с полным перекрытием — это система, в которой имеются средства защиты на каждый возможный путь проникновения. Если в такой системе $\exists (t_i,o_j) \in V$, то $\exists (t_i,o_j,m_k) \in B$.

Далее производятся попытки количественно определить степень безопасности системы, сопоставляя каждой дуге весовой коэффициент.

Модель системы безопасности с полным перекрытием описывает требования к составу подсистемы защиты ИС. Но в ней не рассматривается вопрос стоимости внедряемых средств защиты и соотношения затрат на защиту и получаемого эффекта. Кроме того, определить полное множество «путей проникновения» в систему на практике может оказаться достаточно сложно. А именно от того, как полно описано это множество, зависит то, насколько полученный результат будет адекватен реальному положению дел.

4.3. УПРАВЛЕНИЕ ИНФОРМАЦИОННОИ БЕЗОПАСНОСТЬЮ. СТАНДАРТЫ ISO/IEC 17799/27002 И 27001

Международные стандарты ISO/IEC 17799 (новая версия вышла под номером 27002) и 27001 посвящены вопросам управления информационной безопасностью, и так как они взаимосвязаны, рассматривать их будем в одном разделе.

В 1995 году Британским институтом стандартов (BSI) был опубликован стандарт BS 7799 Part 1 «Code of Practice for Information Security Management» (название обычно переводится как «Практические правила управления информационной безопасностью»). На его основе в 2000 году был принят уже международный стандарт ISO/IEC 17799:2000 «Information technology. Code of practice for information security management». Следующая дополненная версия была принята в 2005 году и обозначается ISO/IEC 17799:2005. А в 2007-м году данный стандарт был переиздан под номером ISO/IEC 27002. Как следует из названия, он описывает рекомендуемые меры в области управления информационной безопасностью и, в целом, не предназначался для проведения сертификации систем на его соответствие.

В 1999 году была опубликована вторая часть стандарта: BS 7799 Part 2 «Information Security Management Systems — Specification with guidance for use» (Системы управления информационной безопасностью — спецификации с руководством по использованию). На его базе был разработан стандарт ISO/IEC 27001:2005 «Information Technology. Security techniques. Information security management systems. Requirements», на соответствие которому может проводиться сертификация.

В России на данный момент действуют стандарты ГОСТ Р ИСО/МЭК 17799-2005 «Информационная технология. Практические правила управления информационной безопасностью» (аутентичный перевод ISO/IEC 17799:2000) и ГОСТ Р ИСО/МЭК 27001-2006 «Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требо-

вания» (перевод ISO/IEC 27001:2005). Несмотря на некоторые внутренние расхождения, связанные с разными версиями и особенностями перевода, наличие стандартов позволяет привести систему управления информационной безопасностью в соответствие их требованиям и, при необходимости, сертифицировать.

4.3.1. ГОСТ Р ИСО/МЭК 17799:2005 «Информационная технология. Практические правила управления информационной безопасностью»

Рассмотрим теперь содержание стандарта ИСО/МЭК 17799 [17]. Во введении указывается, что «информация, поддерживающие ее процессы, информационные системы и сетевая инфраструктура являются существенными активами организации. Конфиденциальность, целостность и доступность информации могут существенно способствовать обеспечению конкурентоспособности, ликвидности, доходности, соответствия законодательству и деловой репутации организации». Таким образом, можно говорить о том, что данный стандарт рассматривает вопросы информационной безопасности, в том числе, и с точки зрения экономического эффекта.

Указываются три группы факторов, которые необходимо учитывать при формировании требований в области информационной безопасности:

- оценка рисков организации. Посредством оценки рисков происходит выявление угроз активам организации, оценка уязвимости соответствующих активов и вероятности возникновения угроз, а также оценка возможных последствий;
- юридические, законодательные, регулирующие и договорные требования, которым должны удовлетворять организация, ее торговые партнеры, подрядчики и поставщики услуг;
- специфический набор принципов, целей и требований, разработанных организацией в отношении обработки информации.

После того как определены требования, идет этап выбора и внедрения мероприятий по управлению информационной безопасно-

стью, которые обеспечат снижение рисков до приемлемого уровня. Выбор мероприятий по управлению информационной безопасностью должен основываться на соотношении стоимости их реализации, эффекта от снижения рисков и возможных убытков в случае нарушения безопасности. Также следует принимать во внимание факторы, которые не могут быть представлены в денежном выражении, например, потерю репутации. Возможный перечень мероприятий приводится в стандарте, но отмечается, что он может быть дополнен или сформирован самостоятельно исходя из потребностей организации.

Кратко перечислим разделы стандарта и предлагаемые в них мероприятия по защите информации. Первая их группа касается политики безопасности. Требуется, чтобы она была разработана, утверждена руководством организации, издана и доведена до сведения всех сотрудников. Она должна определять порядок работы с информационными ресурсами организации, обязанности и ответственность сотрудников. Политика периодически пересматривается, чтобы соответствовать текущему состоянию системы и выявленным рискам.

Следующий раздел затрагивает организационные вопросы, связанные с обеспечением информационной безопасности. Стандарт рекомендует создавать управляющие советы (с участием высшего руководства компании) для утверждения политики безопасности, назначения ответственных лиц, распределения обязанностей и координации внедрения мероприятий по управлению информационной безопасностью в организации. Также должен быть описан процесс получения разрешений на использование в организации средств обработки информации (в т. ч. нового программного обеспечения и аппаратуры), чтобы это не привело к возникновению проблем с безопасностью. Требуется определить и порядок взаимодействия с другими организациями по вопросам информационной безопасности, проведения консультаций с «внешними» специалистами, независимой проверки (аудита) информационной безопасности.

При предоставлении доступа к информационным системам специалистам сторонних организаций необходимо особое внимание уделить вопросам безопасности. Должна быть проведена оценка рисков, связанных с разными типами доступа (физическим или логическим, т. е. удаленным) таких специалистов к различным ресурсам организации. Необходимость предоставления доступа должна быть обоснована, а в договоры со сторонними лицами и организациями должны быть включены требования, касающиеся соблюдения политики безопасности. Аналогичным образом предлагается поступать и в случае привлечения сторонних организаций к обработке информации (аутсорсинга).

Следующий раздел стандарта посвящен вопросам классификации и управления активами. Для обеспечения информационной безопасности организации необходимо, чтобы все основные информационные активы были учтены и закреплены за ответственными владельцами. Начать предлагается с проведения инвентаризации. В качестве примера приводится следующая классификация активов:

- информационные (базы данных и файлы данных, системная документация и т. д.);
- программное обеспечение (прикладное программное обеспечение, системное программное обеспечение, инструментальные средства разработки и утилиты);
- физические активы (компьютерное оборудование, оборудование связи, носители информации, другое техническое оборудование, мебель, помещения);
- услуги (вычислительные услуги и услуги связи, основные коммунальные услуги).

Далее предлагается классифицировать информацию, чтобы определить ее приоритетность, необходимость и степень ее защиты. При этом можно оценить соответствующую информацию с учетом того, насколько она критична для организации, например, с точки зрения обеспечения ее целостности и доступности. После этого пред-

лагается разработать и внедрить процедуру маркировки при обработке информации. Для каждого уровня классификации следует определять процедуры маркировки, для того чтобы учесть следующие типы обработки информации:

- копирование;
- хранение;
- передачу по почте, факсом и электронной почтой;
- передачу голосом, включая мобильный телефон, голосовую почту, автоответчики;
- уничтожение.

Раздел 6 рассматривает вопросы безопасности, связанные с персоналом. Стандартом определяется, чтобы обязанности по соблюдению требований безопасности распределялись на стадии подбора персонала, включались в трудовые договоры, и проводился их мониторинг в течение всего периода работы сотрудника. В частности, при приеме в постоянный штат рекомендуется проводить проверку подлинности представляемых претендентом документов, полноту и точность резюме, представляемые им рекомендации. Рекомендуется, чтобы сотрудники подписывали соглашение о конфиденциальности, уведомляющее о том, какая информация является конфиденциальной или секретной. Должна быть определена дисциплинарная ответственность сотрудников, нарушивших политику и процедуры безопасности организации. Там, где необходимо, эта ответственность должна сохраняться и в течение определенного срока после увольнения с работы.

Пользователей необходимо обучать процедурам безопасности и правильному использованию средств обработки информации, чтобы минимизировать возможные риски. Кроме того, должен быть определен порядок информирования о нарушениях информационной безопасности, с которым необходимо ознакомить персонал. Аналогичная процедура должна задействоваться в случаях сбоев программного

обеспечения. Подобные инциденты требуется регистрировать и проводить их анализ для выявления повторяющихся проблем.

Следующий раздел стандарта посвящен вопросам физической защиты и защиты от воздействия окружающей среды. Указывается, что «средства обработки критичной или важной служебной информации необходимо размещать в зонах безопасности, обозначенных определенным периметром безопасности, обладающим соответствующими защитными барьерами и средствами контроля проникновения. Эти зоны должны быть физически защищены от неавторизованного доступа, повреждения и воздействия». Кроме организации контроля доступа в охраняемые зоны, должны быть определены порядок проведения в них работ и, при необходимости, процедуры организации доступа посетителей. Необходимо также обеспечивать безопасность оборудования (включая и то, что используется вне организации), чтобы уменьшить риск неавторизованного доступа к данным и защитить их от потери или повреждения. К этой же группе требований относится обеспечение защиты от сбоев электропитания и защиты кабельной сети. Также должен быть определен порядок технического обслуживания оборудования, учитывающий требования безопасности, и порядок безопасной утилизации или повторного использования оборудования. Например, списываемые носители данных, содержащие важную информацию, рекомендуется физически разрушать или перезаписывать безопасным образом, а не использовать стандартные функции удаления данных.

С целью минимизации риска неавторизованного доступа или повреждения бумажных документов, носителей данных и средств обработки информации рекомендуется внедрить политику «чистого стола» в отношении бумажных документов и сменных носителей данных, а также политику «чистого экрана» в отношении средств обработки информации. Оборудование, информацию или программное обеспечение можно выносить из помещений организации только на основании соответствующего разрешения.

Название раздела 8 – «Управление передачей данных и операционной деятельностью». В нем требуется, чтобы были установлены обязанности и процедуры, связанные с функционированием всех средств обработки информации. Например, должны контролироваться изменения конфигурации в средствах и системах обработки информации. Требуется реализовать принцип разграничения обязанностей в отношении функций управления, выполнения определенных задач и областей.

Рекомендуется провести разделение сред разработки, тестирования и промышленной эксплуатации программного обеспечения (ПО). Правила перевода ПО из статуса разрабатываемого в статус принятого к эксплуатации должны быть определены и документально оформлены.

Дополнительные риски возникают при привлечении сторонних подрядчиков для управления средствами обработки информации. Такие риски должны быть идентифицированы заранее, а соответствующие мероприятия по управлению информационной безопасностью согласованы с подрядчиком и включены в контракт.

Для обеспечения необходимых мощностей по обработке и хранению информации необходим анализ текущих требований к производительности, а также прогноз будущих. Эти прогнозы должны учитывать новые функциональные и системные требования, а также текущие и перспективные планы развития информационных технологий в организации. Требования и критерии для принятия новых систем должны быть четко определены, согласованы, документально оформлены и опробованы.

Необходимо принимать меры предотвращения и обнаружения внедрения вредоносного программного обеспечения, такого как компьютерные вирусы, сетевые «черви», «троянские кони» и логические бомбы. Отмечается, что защита от вредоносного программного обеспечения должна основываться на понимании требований безопасно-

сти, соответствующих мерах контроля доступа к системам и надлежащем управлении изменениями.

Должен быть определен порядок проведения вспомогательных операций, к которым относится резервное копирование программного обеспечения и данных, регистрация событий и ошибок и, где необходимо, мониторинг состояния аппаратных средств. Мероприятия по резервированию для каждой отдельной системы должны регулярно тестироваться для обеспечения уверенности в том, что они удовлетворяют требованиям планов по обеспечению непрерывности бизнеса.

Для обеспечения безопасности информации в сетях и защиты поддерживающей инфраструктуры требуется внедрение средств контроля безопасности и защита подключенных сервисов от неавторизованного доступа.

Особое внимание уделяется вопросам безопасности носителей информации различного типа: документов, компьютерных носителей информации (лент, дисков, кассет), данных ввода/вывода и системной документации от повреждений. Рекомендуется установить порядок использования сменных носителей компьютерной информации (порядок контроля содержимого, хранения, уничтожения и т. д.). Как уже отмечалось выше, носители информации по окончании использования следует надежно и безопасно утилизировать.

С целью обеспечения защиты информации от неавторизованного раскрытия или неправильного использования необходимо определить процедуры обработки и хранения информации. Эти процедуры должны быть разработаны с учетом категорирования информации и действовать в отношении документов, вычислительных систем, сетей, переносных компьютеров, мобильных средств связи, почты, речевой почты, речевой связи вообще, мультимедийных устройств, использования факсов и любых других важных объектов, например, бланков, чеков и счетов. Системная документация может содержать определенную важную информацию, поэтому тоже должна защищаться.

Процесс обмена информацией и программным обеспечением между организациями должен быть под контролем и соответствовать действующему законодательству. В частности, должна обеспечиваться безопасность носителей информации при пересылке, определена политика использования электронной почты и электронных офисных систем. Следует уделять внимание защите целостности информации, опубликованной электронным способом, например, информации на Web-сайте. Также необходим соответствующий формализованный процесс авторизации, прежде чем такая информация будет сделана общедоступной.

Следующий раздел стандарта посвящен вопросам контроля доступа. В нем требуется, чтобы правила контроля доступа и права каждого пользователя или группы пользователей однозначно определялись политикой безопасности. Пользователи и поставщики услуг должны быть оповещены о необходимости выполнения данных требований.

При использовании парольной аутентификации необходимо осуществлять контроль в отношении паролей пользователей. В частности, пользователи должны подписывать документ о необходимости соблюдения полной конфиденциальности паролей. Требуется обеспечить безопасность процесса получения пароля пользователем и, если это используется, управления пользователями своими паролями (принудительная смена пароля после первого входа в систему и т. д.).

Доступ как к внутренним, так и к внешним сетевым сервисам должен быть контролируемым. Пользователям следует обеспечивать непосредственный доступ только к тем сервисам, в которых они были авторизованы. Особое внимание должно уделяться проверке подлинности удаленных пользователей. Исходя из оценки риска, важно определить требуемый уровень защиты для выбора соответствующего метода аутентификации. Также должна контролироваться безопасность использования сетевых служб.

Многие сетевые и вычислительные устройства имеют встроенные средства удаленной диагностики и управления. Меры обеспечения безопасности должны распространяться и на эти средства.

В случае, когда сети используются совместно несколькими организациями, должны быть определены требования политики контроля доступа, учитывающие это обстоятельство. Также может потребоваться внедрение дополнительных мероприятий по управлению информационной безопасностью, чтобы ограничивать возможности пользователей по подсоединению.

На уровне операционной системы следует использовать средства информационной безопасности для ограничения доступа к компьютерным ресурсам. Это относится к идентификации и аутентификации терминалов и пользователей. Рекомендуется, чтобы все пользователи имели уникальные идентификаторы, которые не должны содержать признаков уровня привилегии пользователя. В системах управления паролем должны быть предусмотрены эффективные интерактивные возможности поддержки необходимого их качества. Использование системных утилит должно быть ограничено и тщательным образом контролироваться.

Желательно предусматривать сигнал тревоги на случай, когда пользователь может стать объектом насилия[1] (если такое событие оценивается как вероятное). При этом необходимо определить обязанности и процедуры реагирования на сигнал такой тревоги.

Терминалы, обслуживающие системы высокого риска, при размещении их в легкодоступных местах должны отключаться после определенного периода их бездействия для предотвращения доступа неавторизованных лиц. Также может вводиться ограничение периода времени, в течение которого разрешены подсоединения терминалов к компьютерным сервисам.

[1] В качестве примера можно назвать пароли для входа «под принуждением». Если пользователь вводит такой пароль, система отображает процесс обычного входа пользователя, после чего имитируется сбой, чтобы нарушители не смогли получить доступ к данным.

На уровне приложений также необходимо применять меры обеспечения информационной безопасности. В частности, это может быть ограничение доступа для определенных категорий пользователей. Системы, обрабатывающие важную информацию, должны быть обеспечены выделенной (изолированной) вычислительной средой.

Для обнаружения отклонения от требований политики контроля доступа и обеспечения доказательства на случай выявления инцидентов нарушения информационной безопасности необходимо проводить мониторинг системы. Результаты мониторинга следует регулярно анализировать. Журнал аудита может использоваться для расследования инцидентов, поэтому достаточно важной является правильная установка (синхронизация) компьютерных часов.

При использовании переносных устройств, например, ноутбуков, необходимо принимать специальные меры противодействия компрометации служебной информации. Необходимо принять формализованную политику, учитывающую риски, связанные с работой с переносными устройствами, в особенности в незащищенной среде.

Десятый раздел стандарта называется «Разработка и обслуживание систем». Уже на этапе разработки информационных систем необходимо обеспечить учет требований безопасности. А в процессе эксплуатации системы требуется предотвращать потери, модификацию или неправильное использование пользовательских данных. Для этого в прикладных системах рекомендуется предусмотреть подтверждение корректности ввода и вывода данных, контроль обработки данных в системе, аутентификацию сообщений, протоколирование действий пользователя.

Для обеспечения конфиденциальности, целостности и аутентификации данных могут быть использованы криптографические средства защиты.

Важную роль в процессе защиты информации играет обеспечение целостности программного обеспечения. Чтобы свести к минимуму повреждения информационных систем, следует строго контро-

лировать внедрение изменений. Периодически возникает необходимость внести изменения в операционные системы. В этих случаях необходимо провести анализ и протестировать прикладные системы с целью обеспечения уверенности в том, что не оказывается никакого неблагоприятного воздействия на их функционирование и безопасность. Насколько возможно, готовые пакеты программ рекомендуется использовать без внесения изменений.

Связанным вопросом является противодействие «троянским» программам и использованию скрытых каналов утечки. Одним из методов противодействия является использование программного обеспечения, полученного от доверенных поставщиков, и контроль целостности системы.

В случаях, когда для разработки программного обеспечения привлекается сторонняя организация, необходимо предусмотреть меры по контролю качества и правильности выполненных работ.

Следующий раздел стандарта посвящен вопросам управления непрерывностью бизнеса. На начальном этапе предполагается идентифицировать события, которые могут быть причиной прерывания бизнес-процессов (отказ оборудования, пожар и т. п.). При этом нужно провести оценку последствий, после чего разработать планы восстановления. Адекватность планов должна быть подтверждена тестированием, а сами они должны периодически пересматриваться, чтобы учитывать происходящие в системе изменения.

Заключительный раздел посвящен вопросам соответствия требованиям. В первую очередь, это касается соответствия системы и порядка ее эксплуатации требованиям законодательства. Сюда относятся вопросы соблюдения авторского права (в том числе, на программное обеспечение), защиты персональной информации (сотрудников, клиентов), предотвращения нецелевого использования средств обработки информации. При использовании криптографических средств защиты информации, они должны соответствовать действующему законодательству. Также должна быть досконально прорабо-

тана процедура сбора доказательств на случай судебных разбирательств, связанных с инцидентами в области безопасности информационной системы.

Сами информационные системы должны соответствовать политике безопасности организации и используемым стандартам. Безопасность информационных систем необходимо регулярно анализировать и оценивать. В то же время требуется соблюдать меры безопасности и при проведении аудита безопасности, чтобы это не привело к нежелательным последствиям (например, сбой критически важного сервера из-за проведения проверки).

Подводя итог можно отметить, что в стандарте рассмотрен широкий круг вопросов, связанных с обеспечением безопасности информационных систем, и по ряду направлений даются практические рекомендации.

4.3.2. ГОСТ Р ИСО/МЭК 27001-2006 «Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования»

Разработчики стандарта отмечают, что он был подготовлен в качестве модели для разработки, внедрения, функционирования, мониторинга, анализа, поддержки и улучшения системы менеджмента информационной безопасности (СМИБ). СМИБ (англ. — information security management system; ISMS) определяется как часть общей системы менеджмента, основанная на использовании методов оценки бизнес-рисков для разработки, внедрения, функционирования, мониторинга, анализа, поддержки и улучшения информационной безопасности. Система менеджмента включает в себя организационную структуру, политики, деятельность по планированию, распределение ответственности, практическую деятельность, процедуры, процессы и ресурсы.

Стандарт предполагает использование процессного подхода для разработки, внедрения, обеспечения функционирования, мониторин-

га, анализа, поддержки и улучшения СМИБ организации. Он основан на модели «Планирование (Plan) – Осуществление (Do) – Проверка (Check) – Действие (Act)» (PDCA), которая может быть применена при структурировании всех процессов СМИБ. На рис. 4.4 показано, как СМИБ, используя в качестве входных данных требования ИБ и ожидаемые результаты заинтересованных сторон, с помощью необходимых действий и процессов выдает выходные данные по результатам обеспечения информационной безопасности, которые соответствуют этим требованиям и ожидаемым результатам.

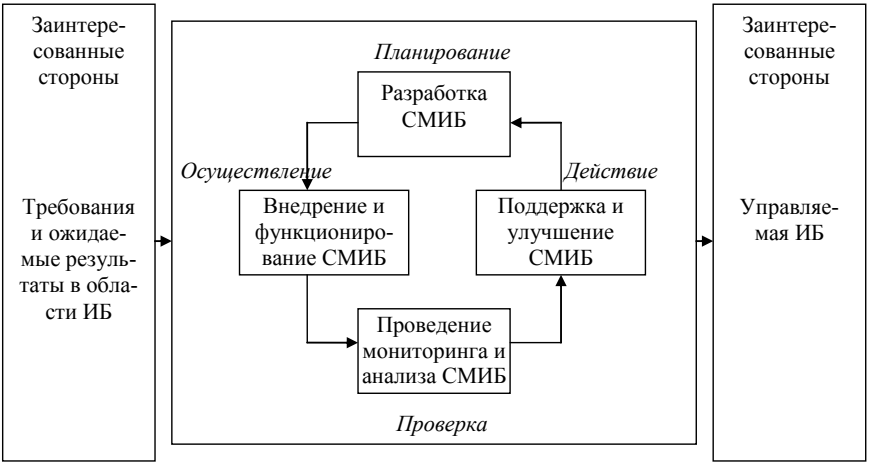


Рис. 4.4. Этапы построения и использования СМИБ

На этапе «*Разработка системы менеджмента информационной безопасности*» организация должна осуществить следующее:

- определить область и границы действия СМИБ;
- определить политику СМИБ на основе характеристик бизнеса, организации, ее размещения, активов и технологий;
- определить подход к оценке риска в организации;
- идентифицировать риски;
- проанализировать и оценить риски;
- определить и оценить различные варианты обработки рисков;

- выбрать цели и меры управления для обработки рисков;
- получить утверждение руководством предполагаемых остаточных рисков;
- получить разрешение руководства на внедрение и эксплуатацию СМИБ;
- подготовить Положение о применимости.

Этап «*Внедрение и функционирование системы менеджмента информационной безопасности*» предполагает, что организация должна:

- разработать план обработки рисков, определяющий соответствующие действия руководства, ресурсы, обязанности и приоритеты в отношении менеджмента рисков ИБ;
- реализовать план обработки рисков для достижения намеченных целей управления, включающий в себя вопросы финансирования, а также распределение функций и обязанностей;
- внедрить выбранные меры управления;
- определить способ измерения результативности выбранных мер управления;
- реализовать программы по обучению и повышению квалификации сотрудников;
- управлять работой СМИБ;
- управлять ресурсами СМИБ;
- внедрить процедуры и другие меры управления, обеспечивающие быстрое обнаружение событий ИБ и реагирование на инциденты, связанные с ИБ.

Третий этап «*Проведение мониторинга и анализа системы менеджмента информационной безопасности*» требует:

- выполнять процедуры мониторинга и анализа;
- проводить регулярный анализ результативности СМИБ;
- измерять результативность мер управления для проверки соответствия требованиям ИБ;

- пересматривать оценки рисков через установленные периоды времени, анализировать остаточные риски и установленные приемлемые уровни рисков, учитывая изменения;
- проводить внутренние аудиты СМИБ через установленные периоды времени;
- регулярно проводить руководством организации анализ СМИБ в целях подтверждения адекватности ее функционирования и определения направлений совершенствования;
- обновлять планы ИБ с учетом результатов анализа и мониторинга;
- регистрировать действия и события, способные повлиять на результативность или функционирование СМИБ.

И наконец, этап «*Поддержка и улучшение системы менеджмента информационной безопасности*» предполагает, что организация должна регулярно проводить следующие мероприятия:

- выявлять возможности улучшения СМИБ;
- предпринимать необходимые корректирующие и предупреждающие действия, использовать на практике опыт по обеспечению ИБ, полученный как в собственной организации, так и в других организациях;
- передавать подробную информацию о действиях по улучшению СМИБ всем заинтересованным сторонам, при этом степень ее детализации должна соответствовать обстоятельствам и, при необходимости, согласовывать дальнейшие действия;
- обеспечивать внедрение улучшений СМИБ для достижения запланированных целей.

Далее в стандарте приводятся требования к документации, которая должна включать положения политики СМИБ и описание области функционирования, описание методики и отчет об оценке рисков, план обработки рисков, документирование связанных процедур. Также должен быть определен процесс управления документами СМИБ, включающий актуализацию, использование, хранение и уничтожение.

Для предоставления свидетельств соответствия требованиям и результативности функционирования СМИБ необходимо вести и поддерживать в рабочем состоянии учетные записи и записи о выполнении процессов. В качестве примеров называются журналы регистрации посетителей, отчеты о результатах аудита и т. п.

Стандарт определяет, что руководство организации ответственно за обеспечение и управление ресурсами, необходимыми для создания СМИБ, а также организацию подготовки персонала.

Как уже ранее отмечалось, организация должна в соответствии с утвержденным графиком проводить внутренние аудиты СМИБ, позволяющие оценить ее функциональность и соответствие стандарту. А руководство должно проводить анализ системы менеджмента информационной безопасности.

Также должны проводиться работы по улучшению системы менеджмента информационной безопасности: повышению ее результативности и уровня соответствия текущего состояния системы и предъявляемым к ней требованиям.

В приложении к стандарту перечисляются рекомендуемые меры управления, взятые из ранее рассмотренного стандарта ISO/IEC 17799:2005.

4.4. МЕТОДИКИ ПОСТРОЕНИЯ СИСТЕМ ЗАЩИТЫ ИНФОРМАЦИИ

4.4.1. Модель Lifecycle Security

Роль анализа рисков для создания корпоративной системы защиты информации в компьютерной сети предприятия можно наглядно показать на примере модели Lifecycle Security [19] (название можно перевести как «жизненный цикл безопасности»), разработанной компанией Axent, впоследствии приобретенной Symantec.

Lifecycle Security — это обобщенная схема построения комплексной защиты компьютерной сети предприятия. Выполнение описываемого в ней набора процедур позволяет системно решать задачи,

связанные с защитой информации, и дает возможность оценить эффект от затраченных средств и ресурсов. С этой точки зрения, идеология Lifecycle Security может быть противопоставлена тактике «точечных решений», заключающейся в том, что все усилия сосредотачиваются на внедрении отдельных частных решений (например, межсетевых экранов или систем аутентификации пользователей по смарт-картам). Без предварительного анализа и планирования подобная тактика может привести к появлению в компьютерной системе набора разрозненных продуктов, которые не стыкуются друг с другом и не позволяют решить проблемы предприятия в сфере информационной безопасности.

Lifecycle Security включает в себя 7 основных компонентов, которые можно рассматривать как этапы построения системы защиты (рис. 4.5).

Рис. 4.5. Компоненты модели Lifecycle Security

Политики безопасности, стандарты, процедуры и метрики. Этот компонент определяет рамки, в которых осуществляются мероприятия по обеспечению безопасности информации, и задает крите-

рии оценки полученных результатов. Стоит отметить, что под стандартами здесь понимаются не только государственные и международные стандарты в сфере информационной безопасности, но и корпоративные стандарты, которые в ряде случаев могут оказать очень существенное влияние на создаваемую систему защиты информации. Также хочется остановиться на обязательном введении метрики, позволяющей оценить состояние системы до и после проведения работ по защите информации. Метрика определяет, в чем и как измеряется защищенность системы, и позволяет соотнести сделанные затраты и полученный эффект.

Анализ рисков. Этот этап является отправной точкой для установления и поддержания эффективного управления системой защиты. Проведение анализа рисков позволяет подробно описать состав и структуру информационной системы (если по каким-то причинам это не было сделано ранее), расположить имеющиеся ресурсы по приоритетам, основываясь на степени их важности для нормальной работы предприятия, оценить угрозы и идентифицировать уязвимости системы.

Стратегический план построения системы защиты. Результаты анализа рисков используются как основа для разработки стратегического плана построения системы защиты. Наличие подобного плана помогает распределить по приоритетам бюджеты и ресурсы и в последующем осуществить выбор продуктов и разработать стратегию их внедрения.

Выбор и внедрение решений. Хорошо структурированные критерии выбора решений в сфере защиты информации и наличие программы внедрения уменьшают вероятность приобретения продуктов, становящихся «мертвым грузом», мешающим развитию информационной системы предприятия. Кроме непосредственно выбора решений, также должно учитываться качество предоставляемых поставщиками сервисных и обучающих услуг. Кроме того, необходимо чет-

ко определить роль внедряемого решения в выполнении разработанных планов и достижении поставленных целей в сфере безопасности.

Обучение персонала. Знания в области компьютерной безопасности и технические тренинги необходимы для построения и обслуживания безопасной вычислительной среды. Усилия, затраченные на обучение персонала, значительно повышают шансы на успех мероприятий по защите сети.

Мониторинг защиты. Он помогает обнаруживать аномалии или вторжения в ваши компьютеры и сети и является средством контроля над системой защиты, чтобы гарантировать эффективность программ защиты информации.

Разработка методов реагирования в случае инцидентов и восстановление. Без наличия заранее разработанных и «отрепетированных» процедур реагирования на инциденты в сфере безопасности невозможно гарантировать, что в случае обнаружения атаки ей будут противопоставлены эффективные меры защиты, и работоспособность системы будет быстро восстановлена.

Все компоненты программы взаимосвязаны и предполагается, что процесс совершенствования системы защиты идет непрерывно.

Остановимся более подробно на этапе анализа рисков. По мнению разработчиков модели Lifecycle Security, он должен проводиться в следующих случаях:

- до и после обновления или существенных изменений в структуре системы;
- до и после перехода на новые технологии;
- до и после подключения к новым сетям (например, подключения локальной сети филиала к сети головного офиса);
- до и после подключения к глобальным сетям (в первую очередь, Интернет);
- до и после изменений в порядке ведения бизнеса (например, при открытии электронного магазина);
- периодически, для проверки эффективности системы защиты.

Ключевые моменты этапа анализа рисков:

1. Подробное документирование компьютерной системы предприятия. При этом особое внимание необходимо уделять критически важным приложениям.
2. Определение степени зависимости организации от нормального функционирования фрагментов компьютерной сети, конкретных узлов, от безопасности хранимых и обрабатываемых данных.
3. Определение уязвимых мест компьютерной системы.
4. Определение угроз, которые могут быть реализованы в отношении выявленных уязвимых мест.
5. Определение и оценка всех рисков, связанных с эксплуатацией компьютерной системы.

Особо хочется обратить внимание на связь анализа рисков с другими компонентами модели. С одной стороны, наличие метрики защищенности и определение значений, характеризующих состояние системы до и после мероприятий по защите информации, накладывают определенные требования на процедуру анализа рисков. Ведь на базе полученных результатов и оценивается состояние системы. С другой стороны, они дают те начальные условия, исходя из которых разрабатывается план построения системы защиты сети. И результаты анализа рисков должны быть сформулированы в виде, пригодном для выполнения как первой, так и второй функции.

4.4.2. Модель многоуровневой защиты

Понятие многоуровневой защиты или эшелонированной обороны (от англ. «Defense in depth») пришло в информационные технологии из военных руководств.

С точки зрения информационной безопасности, модель многоуровневой защиты определяет набор уровней защиты информационной системы. Модель часто используется корпорацией Майкрософт в руководствах по безопасности. Корректная организация защиты на каждом из выделенных уровней позволяет уберечь систему от реализации угроз информационной безопасности.

Перечень выделяемых уровней незначительно различается в различных документах, один из возможных вариантов представлен на рис. 4.6 [20].

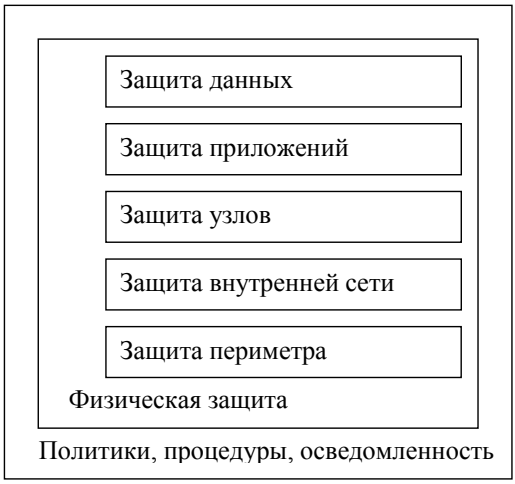


Рис. 4.6. Модель многоуровневой защиты

Как уже отмечалось выше, политика безопасности должна описывать все аспекты работы системы с точки зрения обеспечения информационной безопасности. Поэтому *уровень политики безопасности* можно рассматривать как базовый. Этот уровень также подразумевает наличие документированных организационных мер защиты (процедур) и порядка информирования о происшествиях, обучение пользователей в области информационной безопасности и прочие меры аналогичного характера (например, рекомендуемые стандартом ISO/IEC 17799).

Уровень физической защиты включает меры по ограничению физического доступа к ресурсам системы — защита помещений, контроль доступа, видеонаблюдение и т. д. Сюда же относятся средства защиты мобильных устройств, используемых сотрудниками в служебных целях.

Уровень защиты периметра определяет меры безопасности в «точках входа» в защищаемую сеть из внешних, потенциально опасных. Классическим средством защиты периметра является межсетевой экран, который на основании заданных правил определяет, может ли проходящий сетевой пакет быть пропущен в защищаемую сеть (см. раздел 3.4). Другие примеры средств защиты периметра — системы обнаружения вторжений, средства антивирусной защиты для шлюзов безопасности и т. д.

Уровень защиты внутренней сети «отвечает» за обеспечение безопасности передаваемого внутри сети трафика и сетевой инфраструктуры. Примеры средств и механизмов защиты на этом уровне — создание виртуальных локальных сетей (VLAN) с помощью управляемых коммутаторов, защита передаваемых данных с помощью протокола IPSec и т. д. Нередко внутри сети также используют средства, характерные для защиты периметра, например, межсетевые экраны, в том числе и персональные (устанавливаемые на защищаемый компьютер). Связано это с тем, что использование беспроводных сетевых технологий и виртуальных частных сетей (VPN) приводит к «размыванию» периметра сети. Например, если атакующий смог подключиться к точке беспроводного доступа внутри защищаемой сети, его действия уже не будут контролироваться межсетевым экраном, установленным «на границе» сети, хотя формально атака будет производиться с внешнего по отношению к нашей сети компьютера. Поэтому иногда при анализе рассматривают «*уровень защиты сети*», включающий и защиту периметра, и внутренней сети.

Следующим на схеме идет *уровень защиты узлов*. Здесь рассматриваются атаки на отдельный узел сети и, соответственно, меры защиты от них. Может учитываться функциональность узла и отдельно рассматриваться защита серверов и рабочих станций. В первую очередь, необходимо уделять внимание защите на уровне операционной системы — настройкам, повышающим безопасность конфигурации (в том числе, отключению не использующихся или потенциально

опасных служб), организации установки исправлений и обновлений, надежной аутентификации пользователей. Исключительно важную роль играет антивирусная защита.

Уровень защиты приложений отвечает за защиту от атак, направленных на конкретные приложения — почтовые серверы, web-серверы, серверы баз данных. В качестве примера можно назвать SQL-инъекции — атаки на сервер БД, заключающиеся в том, что во входную текстовую строку включаются операторы языка SQL, что может нарушить логику обработки данных и привести к получению нарушителем конфиденциальной информации. Сюда же можно отнести модификацию приложений компьютерными вирусами. Для защиты от подобных атак используются настройки безопасности самих приложений, установка обновлений, средства антивирусной защиты.

Уровень защиты данных определяет порядок защиты обрабатывающихся и хранящихся в системе данных от несанкционированного доступа и других угроз. В качестве примеров контрмер можно назвать разграничение доступа к данным средствами файловой системы, шифрование данных при хранении и передаче.

В процессе идентификации рисков определяется, что является целью нарушителя, и на каком уровне или уровнях защиты можно ему противостоять. Соответственно выбираются и контрмеры. Защита от угрозы на нескольких уровнях снижает вероятность ее реализации, а значит, и уровень риска.

4.4.3. Методика управления рисками, предлагаемая Майкрософт

Ниже представлено краткое описание подхода к управлению рисками, предлагаемого корпорацией Майкрософт. Данное описание базируется на материалах «Руководства по управлению рисками» [20].

Управление рисками рассматривается как одна из составляющих общей программы управления, предназначенной для руководства компаний и позволяющей контролировать ведение бизнеса и принимать обоснованные решения.

Рис. 4.7. Процесс управления рисками безопасности, предлагаемый корпорацией Майкрософт

Процесс управления рисками безопасности, предлагаемый Майкрософт, включает следующие четыре этапа (рис. 4.7):

Этап «оценка рисков» включает в себя следующие мероприятия:

- *планирование сбора данных*: обсуждение основных условий успешной реализации и подготовка рекомендаций;
- *сбор данных о рисках*: описание процесса сбора и анализа данных;
- *выделение наиболее приоритетных рисков*: подробное описание шагов по качественной и количественной оценке рисков.

Этап «поддержка принятия решений»:

- *определение функциональных требований*: определение функциональных требований для снижения рисков;
- *выбор возможных решений для контроля*: описание подхода к выбору решений по нейтрализации риска;
- *экспертиза решения*: проверка предложенных элементов контроля на соответствие функциональным требованиям;
- *оценка снижения риска*: оценка снижения подверженности воздействию или вероятности рисков;
- *оценка стоимости решения*: оценка прямых и косвенных затрат, связанных с решениями по нейтрализации риска;

- *выбор стратегии нейтрализации риска*: определение наиболее экономически эффективного решения по нейтрализации риска путем анализа соотношения затрат и получаемого результата.

Этап «реализация контроля» включает мероприятия по развертыванию и использованию решений для контроля (например, внедрение новых средств защиты), снижающих риск для организации:

- *поиск целостного подхода*: включение персонала, процессов и технологий в решение по нейтрализации риска.
- *организация по принципу многоуровневой защиты*: упорядочение решений по нейтрализации риска в рамках предприятия.

Этап «оценка эффективности программы» предполагает проведение анализа эффективности процесса управления рисками и проверки того, обеспечивают ли элементы контроля надлежащий уровень безопасности:

- *разработка системы показателей рисков*: оценка уровня и изменения риска.
- *оценка эффективности программы*: оценка программы управления рисками для выявления возможностей усовершенствования.

В руководстве [20] особо отмечается, что термины «управление рисками» и «оценка рисков» не являются взаимозаменяемыми. Под управлением рисками понимаются общие мероприятия по снижению риска в рамках организации до приемлемого уровня. Управление рисками представляет собой непрерывный процесс, но производимые оценки чаще всего делаются для годичного интервала. Под оценкой рисков понимается процесс выявления и приоритизации рисков для бизнеса, являющийся составной частью управления рисками.

При описании риска делается указание на то, какое влияние он оказывает на бизнес, и насколько вероятно данное событие. Компоненты, описывающие риск, изображены на рис. 4.8.

На начальном этапе проведения оценки рискам присваиваются значения в соответствии со шкалой: «высокий», «средний» и «низ-

кий». После этого для выявленных наиболее существенных рисков проводится количественная оценка.

Рис. 4.8. Компоненты «полной формулировки» риска

Подробно предлагаемая Майкрософт методика оценки рисков будет рассмотрена в разделе 4.5.5. Перед внедрением в организации процесса управления рисками безопасности необходимо проверить «уровень зрелости» организации (табл. 4.1).

Организациям, в которых отсутствуют формальные политики или процессы, относящиеся к управлению рисками безопасности, будет очень трудно сразу внедрить все аспекты рассматриваемого процесса. Если окажется, что уровень зрелости является достаточно низким, рассматриваемый процесс можно внедрять последовательными этапами на протяжении нескольких месяцев (например, начав с пилотного проекта в отдельном подразделении). Продемонстрировав эффективность процесса управления рисками безопасности на примере пилотного проекта, группа управления рисками безопасности может перейти к внедрению данного процесса в других подразделениях, постепенно охватывая всю организацию.

Таблица 4.1

Уровни зрелости управления рисками безопасности

Уровень	Состояние	Определение
0	Отсутствует	Политика или процесс не документированы. Ранее организация не знала о деловых рисках, связанных с управлением рисками, и не рассматривала данный вопрос
1	Узкоспециализированный	Некоторые члены организации признают значимость управления рисками, однако операции по управлению рисками являются узкоспециализированными. Политики и процессы в организации не документированы, процессы не являются полностью повторяемыми. В результате проекты по управлению рисками являются хаотичными и некоординируемыми, а получаемые результаты не измеряются и не подвергаются аудиту
2	Повторяемый	Организации известно об управлении рисками. Процесс управления рисками является повторяемым, но развит слабо. Процесс документирован не полностью, однако соответствующие операции выполняются регулярно, и организация стремится внедрить всеобъемлющий процесс управления рисками с привлечением высшего руководства. В организации не проводится формальное обучение и информирование по управлению рисками; ответственность за выполнение соответствующих мероприятий возложена на отдельных сотрудников
3	Наличие определенного процесса	Организация приняла формальное решение об интенсивном внедрении управления рисками для управления программой защиты информации. В организации разработан базовый процесс с четко определенными целями и документированными процессами достижения и оценки результатов. Проводится обучение всего персонала основам управления рисками. Организация активно внедряет документированные процессы управления рисками

Окончание табл. 4.1

Уровень	Состояние	Определение
4	Управляемый	На всех уровнях организации имеется глубокое понимание управления рисками. В организации существуют процедура управления рисками и четко определенный процесс, широко распространена информация об управлении рисками, доступно подробное обучение, существуют начальные формы измерений показателей эффективности. Программе управления рисками выделен достаточный объем ресурсов, результаты управления рисками оказывают положительное влияние на работу многих подразделений организации, а группа управления рисками безопасности может постоянно совершенствовать свои процессы и средства. В организации используются некоторые технологические средства, помогающие в управлении рисками, однако большая часть (если не подавляющее большинство) процедур оценки рисков, определения элементов контроля и анализа выгод и затрат выполняется вручную
5	Оптимизированный	Организация выделила на управление рисками безопасности значительные ресурсы, а сотрудники пытаются прогнозировать, какие проблемы могут встретиться в течение следующих месяцев и лет, и каким образом их нужно будет решать. Процесс управления рисками глубоко изучен и в значительной степени автоматизирован путем применения различных средств (разработанных в организации или приобретенных у сторонних разработчиков). При возникновении проблем в системе безопасности выявляется основная причина возникшей проблемы, и предпринимаются необходимые действия для снижения риска ее повторного возникновения. Сотрудники организации могут проходить обучение, обеспечивающее различные уровни подготовки

4.5. МЕТОДИКИ И ПРОГРАММНЫЕ ПРОДУКТЫ ДЛЯ ОЦЕНКИ РИСКОВ

В данном разделе будут приведены краткие описания ряда распространенных методик анализа рисков. Их можно разделить на:

- методики, использующие оценку риска на качественном уровне (например, по шкале «высокий», «средний», «низкий»). К таким методикам, в частности, относится FRAP;
- количественные методики (риск оценивается через числовое значение, например размер ожидаемых годовых потерь). К этому классу относится методика RiskWatch;
- методики, использующие смешанные оценки (такой подход используется в CRAMM, методике Майкрософт и т. д.).

4.5.1. Методика CRAMM

Это одна из первых методик анализа рисков в сфере ИБ — работа над ней была начата в середине 80-х гг. центральным агентством по компьютерам и телекоммуникациям (CCTA) Великобритании.

В основе метода CRAMM лежит комплексный подход к оценке рисков, сочетающий количественные и качественные методы анализа. Метод является универсальным и подходит как для крупных, так и для малых организаций, как правительственного, так и коммерческого сектора. Версии программного обеспечения CRAMM, ориентированные на разные типы организаций, отличаются друг от друга своими базами знаний (авторами методики они называются профилями, англ. «profiles»). Для коммерческих организаций имеется Коммерческий профиль (Commercial Profile), для правительственных организаций — Правительственный профиль (Government profile). Правительственный вариант профиля также позволяет проводить аудит на соответствие требованиям американского стандарта ITSEC («Оранжевая книга»).

Исследование ИБ системы с помощью CRAMM проводится в три стадии [21, 22, 23].

На *первой стадии* анализируется все, что касается идентификации и определения ценности ресурсов системы. Она начинается с решения задачи определения границ исследуемой системы: собираются сведения о конфигурации системы и о том, кто отвечает за физические и программные ресурсы, кто входит в число пользователей системы, как они ее применяют или будут применять.

Проводится идентификация ресурсов: физических, программных и информационных, содержащихся внутри границ системы. Каждый ресурс необходимо отнести к одному из предопределенных классов. Затем строится модель информационной системы с позиции ИБ. Для каждого информационного процесса, имеющего, по мнению пользователя, самостоятельное значение и называемого пользовательским сервисом, строится дерево связей используемых ресурсов. Построенная модель позволяет выделить критичные элементы.

Ценность физических ресурсов в CRAMM определяется стоимостью их восстановления в случае разрушения.

Ценность данных и программного обеспечения определяется в следующих ситуациях:

- недоступность ресурса в течение определенного периода времени;
- разрушение ресурса — потеря информации, полученной со времени последнего резервного копирования, или ее полное разрушение;
- нарушение конфиденциальности в случаях несанкционированного доступа штатных сотрудников или посторонних лиц;
- модификация — рассматривается для случаев мелких ошибок персонала (ошибки ввода), программных ошибок, преднамеренных ошибок;
- ошибки, связанные с передачей информации: отказ от доставки, недоставка информации, доставка по неверному адресу.

Для оценки возможного ущерба CRAMM рекомендует использовать следующие параметры:

- ущерб репутации организации;
- нарушение действующего законодательства;
- ущерб для здоровья персонала;
- ущерб, связанный с разглашением персональных данных отдельных лиц;
- финансовые потери от разглашения информации;
- финансовые потери, связанные с восстановлением ресурсов;
- потери, связанные с невозможностью выполнения обязательств;
- дезорганизация деятельности.

Для данных и программного обеспечения выбираются применимые к данной ИС критерии, дается оценка ущерба по шкале со значениями от 1 до 10.

В описаниях CRAMM в качестве примера приводится в [21] такая шкала оценки по критерию «Финансовые потери, связанные с восстановлением ресурсов»:

- 2 балла — менее $ 1000;
- 6 баллов — от $ 1000 до $ 10 000;
- 8 баллов — от $ 10 000 до $ 100 000;
- 10 баллов — свыше $ 100 000.

При низкой оценке по всем используемым критериям (3 балла и ниже) считается, что рассматриваемая система требует базового уровня защиты (для этого уровня не требуется подробной оценки угроз ИБ) и вторая стадия исследования пропускается.

На *второй стадии* анализа рассматривается все, что относится к идентификации и оценке уровней угроз для групп ресурсов и их уязвимостей. В конце стадии заказчик получает идентифицированные и оцененные уровни рисков для своей системы. На этой стадии оцениваются зависимость пользовательских сервисов от определенных групп ресурсов и существующий уровень угроз и уязвимостей, вычисляются уровни рисков и анализируются результаты.

Ресурсы группируются по типам угроз и уязвимостей. Например, в случае существования угрозы пожара или кражи в качестве

группы ресурсов разумно рассмотреть все ресурсы, находящиеся в одном месте (серверный зал, комната средств связи и т. д.). Оценка уровней угроз и уязвимостей производится на основе исследования косвенных факторов.

Программное обеспечение CRAMM для каждой группы ресурсов и каждого из 36 типов угроз генерирует список вопросов, допускающих однозначный ответ. Уровень угроз оценивается, в зависимости от ответов, как очень высокий, высокий, средний, низкий и очень низкий. Уровень уязвимости оценивается, в зависимости от ответов, как высокий, средний и низкий.

На основе этой информации рассчитываются уровни рисков в дискретной шкале с градациями от 1 до 7. Полученные уровни угроз, уязвимостей и рисков анализируются и согласовываются с заказчиком.

CRAMM объединяет угрозы и уязвимости в матрице риска. Рассмотрим, как получается эта матрица, и что каждый из уровней риска означает.

Таблица 4.2

Шкала оценки уровней угрозы (частота возникновения)

Описание	Значение
инцидент происходит в среднем не чаще, чем каждые 10 лет	очень низкий (very low)
инцидент происходит в среднем один раз в 3 года	низкий (low)
инцидент происходит в среднем раз в год	средний (medium)
инцидент происходит в среднем один раз в четыре месяца	высокий (high)
инцидент происходит в среднем раз в месяц	очень высокий (very high)

Основной подход для решения этой проблемы состоит в рассмотрении [12]:

- уровня угрозы (шкала приведена в табл. 4.2);
- уровня уязвимости (шкала приведена в табл. 4.3);

- размера ожидаемых финансовых потерь (пример приведен в табл. 4.4).

Таблица 4.3

Шкала оценки уровня уязвимости (вероятность успешной реализации угрозы)

Описание	Значение
В случае возникновения инцидента, вероятность развития событий по наихудшему сценарию меньше 0,33	низкий (low)
В случае возникновения инцидента, вероятность развития событий по наихудшему сценарию от 0,33 до 0,66	средний (medium)
В случае возникновения инцидента, вероятность развития событий по наихудшему сценарию выше 0,66	высокий (high)

Таблица 4.4

Матрица ожидаемых годовых потерь

		0,1	**0,1**	**0,1**	**0,34**	**0,34**	**0,34**	**1**	**1**	**1**	**3,33**	**3,33**	**3,33**	**10**	**10**	**10**
		0,1	**0,5**	**1**	**0,1**	**0,5**	**1**	**0,1**	**0,5**	**1**	**0,1**	**0,5**	**1**	**0,1**	**0,5**	**1**
1	**1000**	10	50	…												10^4
2	**10000**	100														
3	**30000**	3×10^2														
4	$\mathbf{10^5}$	10^3														
5	$\mathbf{3\times10^5}$	3×10^3														
6	$\mathbf{10^6}$	10^4														…
7	$\mathbf{3\times10^6}$	3×10^4						…								
8	$\mathbf{10^7}$	10^5														
9	$\mathbf{3\times10^7}$	3×10^5														
10	$\mathbf{10^8}$	10^6														10^9

Исходя из оценок стоимости ресурсов защищаемой ИС, оценок угроз и уязвимостей, определяются «ожидаемые годовые потери». В табл. 4.4 приведен пример матрицы оценки ожидаемый потерь [24]. В ней второй столбец слева содержит значения стоимости ресурса, верхняя строка заголовка таблицы — оценку частоты возникновения угрозы в течение года (уровня угрозы), нижняя строка заголовка — оценку вероятности успеха реализации угрозы (уровня уязвимости). Результат получается путем перемножения указанных оценок.

Значения ожидаемых годовых потерь (англ. «Annual Loss of Expectancy» — ALE) переводятся в CRAMM в баллы, показывающие уровень риска, согласно шкале, представленной в табл. 4.5 (в этом примере из [24] размер потерь приводится в фунтах стерлингов). В соответствии с приведенной ниже матрицей, выводится оценка риска (табл. 4.6).

Таблица 4.5

Шкала оценки уровня риска

Уровень риска (CRAMM Measure of risk)	Ожидаемые годовые потери (Annual Loss of Expectancy)
1	< £ 1 000
2	< £ 10 000
3	< £ 100 000
4	< £ 1 000 000
5	< £ 10 000 000
6	< £ 100 000 000
7	< £ 1 000 000 000

Таблица 4.6

Матрица оценки риска

Threat / Vuln. / Asset Value	Very Low Low	Very Low Medium	Very Low High	Low Low	Low Medium	Low High	Medium Low	Medium Medium	Medium High	High Low	High Medium	High High	Very High Low	Very High Medium	Very High High
1	1	1	1	1	1	1	1	1	2	1	2	2	2	2	3
2	1	1	2	1	2	2	2	3	3	2	3	3	3	3	4
3	1	2	2	2	2	3	2	3	3	3	3	4	3	4	4
4	2	2	3	2	3	3	3	3	4	3	4	4	4	4	5
5	2	3	3	3	3	4	3	4	4	4	4	5	4	5	5
6	3	3	4	3	4	4	4	4	5	4	5	5	5	5	6
7	3	4	4	4	4	5	4	5	5	5	5	6	5	6	6
8	4	4	5	4	5	5	5	5	6	5	6	6	6	6	7
9	4	5	5	5	5	6	5	6	6	6	6	7	7	7	7
10	5	5	6	5	6	6	6	6	6	6	7	7	7	7	7

Третья стадия исследования заключается в поиске адекватных контрмер. По существу, это поиск варианта системы безопасности, наилучшим образом удовлетворяющей требованиям заказчика.

На этой стадии CRAMM генерирует несколько вариантов мер противодействия, адекватных выявленным рискам и их уровням.

Контрмеры можно объединить в три категории: около 300 рекомендаций общего плана; более 1000 конкретных рекомендаций; около 900 примеров того, как можно организовать защиту в данной ситуации.

Таким образом, CRAMM — пример методики расчета, при которой первоначальные оценки даются на качественном уровне, а потом производится переход к количественной оценке (в баллах).

4.5.2. Методика FRAP

Методика «Facilitated Risk Analysis Process (FRAP)» предлагаемая компанией Peltier and Associates (сайт в Интернет http://www.peltierassociates.com/) разработана Томасом Пелтиером (Thomas R. Peltier) и опубликована в [25] (фрагменты данной книги доступны на сайте, приведенное ниже описание построено на их основе). В методике обеспечение ИБ ИС предлагается рассматривать в рамках процесса управления рисками. Управление рисками в сфере ИБ — процесс, позволяющий компаниям найти баланс между затратами средств и сил на средства защиты и получаемым эффектом.

Управление рисками должно начинаться с оценки рисков: должным образом оформленные результаты оценки станут основой для принятия решений в области повышения безопасности системы.

После завершения оценки проводится анализ соотношения затрат и получаемого эффекта (англ. «cost/benefit analysis»), который позволяет определить те средства защиты, которые нужны для снижения риска до приемлемого уровня.

Ниже приведены основные этапы оценки рисков. Данный список во многом повторяет аналогичный перечень из других методик, но во FRAP более подробно раскрываются пути получения данных о системе и ее уязвимостях.

1) Определение защищаемых активов производится с использованием опросных листов, изучения документации на систему, использования инструментов автоматизированного анализа (сканирования) сетей.

2) Идентификация угроз. При составлении списка угроз могут использоваться разные подходы:

- заранее подготовленные экспертами перечни угроз (checklists), из которых выбираются актуальные для данной системы;
- анализ статистики происшествий в данной ИС и в подобных ей — оценивается частота их возникновения; по ряду угроз, например, угрозе возникновения пожара, подобную статистику можно получить у соответствующих государственных организаций;
- «мозговой штурм», проводимый сотрудниками компании.

3) Когда список угроз закончен, каждой из них сопоставляют вероятность возникновения. После чего оценивают ущерб, который может быть нанесен данной угрозой. Исходя из полученных значений, оценивается уровень угрозы.

При проведении анализа, как правило, принимают, что на начальном этапе в системе отсутствуют средства и механизмы защиты. Таким образом оценивается уровень риска для незащищенной ИС, что в последствии позволяет показать эффект от внедрения средств защиты информации (СЗИ).

Оценка производится для вероятности возникновения угрозы и ущерба от нее по следующим шкалам.

Вероятность (англ. «Probability»):

- «высокая» (англ. «High Probability») — очень вероятно, что угроза реализуется в течение следующего года;
- «средняя» (англ. «Medium Probability») — возможно угроза реализуется в течение следующего года;
- «низкая» (англ. «Low Probability») — маловероятно, что угроза реализуется в течение следующего года.

Ущерб (англ. «Impact») — вред (или величины потерь), наносимый активу:

- «высокий» (англ. «High Impact»): остановка критически важных бизнес-подразделений, которая приводит к существенному

ущербу для бизнеса, потере имиджа или неполучению существенной прибыли;

- средний (Medium Impact): кратковременное прерывание работы критических процессов или систем, которое приводит к ограниченным финансовым потерям в одном бизнес-подразделении;
- низкий (Low Impact): перерыв в работе, не вызывающий ощутимых финансовых потерь.

Оценка определяется в соответствии с правилом, задаваемым матрицей рисков, изображенной на рис. 4.9.

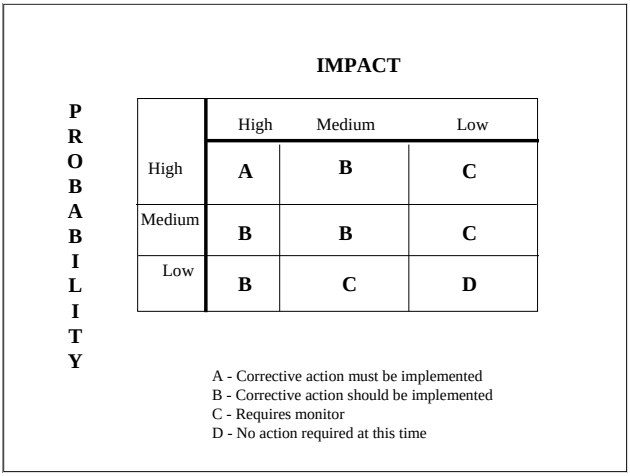


Рис. 4.9. Матрица рисков FRAP

Полученная оценка уровня риска может интерпретироваться следующим образом:

- уровень А — связанные с риском действия (например, внедрение СЗИ) должны быть выполнены немедленно и в обязательном порядке;
- уровень В — связанные с риском действия должны быть предприняты;

- уровень C — требуется мониторинг ситуации (но непосредственных мер по противодействию угрозе принимать, возможно, не надо);
- уровень D — никаких действий в данный момент предпринимать не требуется.

4) После того как угрозы идентифицированы и дана оценка риска, должны быть определены контрмеры, позволяющие устранить риск или свести его до приемлемого уровня. При этом должны приниматься во внимание законодательные ограничения, делающие невозможным или, наоборот, предписывающие в обязательном порядке, использование тех или иных средств и механизмов защиты. Чтобы определить ожидаемый эффект, можно провести оценку того же риска, но при условии внедрения предлагаемого СЗИ. Если риск снижен недостаточно, возможно, надо применить другое СЗИ. Вместе с определением средства защиты, надо определить, какие затраты повлечет его приобретение и внедрение (затраты могут быть как прямые, так и косвенные, см. ниже). Кроме того, необходимо оценить, безопасно ли само это средство, не создает ли оно новых уязвимостей в системе.

Чтобы использовать экономически эффективные средства защиты, нужно проводить анализ соотношения затрат и получаемого эффекта. При этом надо оценивать не только стоимость приобретения решения, но и стоимость поддержания его работы. В затраты могут включаться:

- стоимость реализации проекта, включая дополнительное программное и аппаратное обеспечение;
- снижение эффективности выполнения системой своих основных задач;
- внедрение дополнительных политик и процедур для поддержания средства;
- затраты на найм дополнительного персонала или переобучение имеющегося.

5) Документирование. Когда оценка рисков закончена, ее результаты должны быть подробно документированы в стандартизованном формате. Полученный отчет может быть использован при определении политик, процедур, бюджета безопасности и т. д.

4.5.3. Методика OCTAVE

OCTAVE (англ. «Operationally Critical Threat, Asset, and Vulnerability Evaluation») — методика поведения оценки рисков в организации, разрабатываемая институтом Software Engineering Institute (SEI) при университете Карнеги Меллон (Carnegie Mellon University). Полное описание методики доступно в Интернет на сайте www.cert.org/octave. Ей также посвящено много научных и научно-технических статей [26, 27].

Особенность данной методики заключается в том, что весь процесс анализа производится силами сотрудников организации, без привлечения внешних консультантов. Для этого создается смешанная группа, включающая как технических специалистов, так и руководителей разного уровня, что позволяет всесторонне оценить последствия для бизнеса возможных инцидентов в области безопасности и разработать контрмеры.

OCTAVE предполагает три фазы анализа:

1. Разработка профиля угроз, связанных с активом.
2. Идентификация инфраструктурных уязвимостей.
3. Разработка стратегии и планов безопасности.

Профиль угрозы включает в себя указания на актив (англ. «asset»), тип доступа к активу (англ. «access»), источник угрозы (англ. «actor»), тип нарушения или мотив (англ. «motive»), результат (англ. «outcome») и ссылки на описания угрозы в общедоступных каталогах. По типу источника угрозы в OCTAVE делятся на:

- угрозы, исходящие от человека-нарушителя, действующего через сеть передачи данных;
- угрозы, исходящие от человека-нарушителя, использующего физический доступ;

- угрозы, связанные со сбоями в работе системы;
- прочие.

Результатом может быть раскрытие (англ. «disclosure»), изменение (англ. «modification»), потеря или разрушение (англ. «loss/destruction») информационного ресурса или разрыв подключения, отказ в обслуживании (англ. «interruption»).

Методика OCTAVE предлагает при описании профиля использовать «деревья вариантов», пример подобного дерева для угроз класса 1 (угрозы, исходящие от человека-нарушителя, действующего через сеть передачи данных) приведен на рис. 4.10. При создании профиля угроз рекомендуется избегать обилия технических деталей — это задача второго этапа исследования. Главная задача первой стадии — стандартизованным образом описать сочетание угрозы и ресурса.

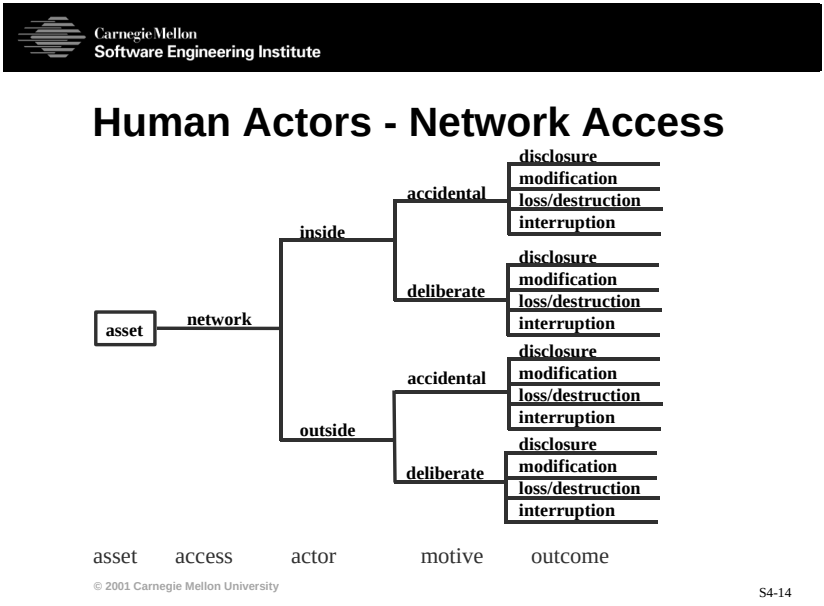


Рис. 4.10. Дерево вариантов, использующееся при описании профиля

Предположим, что на предприятии имеется информационный ресурс (актив) — база данных (БД) отдела кадров (англ. «HR Database»). Профиль, соответствующий угрозе кражи информации сотрудником, предприятия представлен в табл. 4.10.

Таблица 4.10

Пример профиля угрозы

Ресурс (Asset)	БД отдела кадров (HR Database)
Тип доступа (Access)	Через сеть передачи данных (Network)
Источник угрозы (Actor)	Внутренний (Inside)
Тип нарушения (Motive)	Преднамеренное (Deliberate)
Уязвимость (Vulnerability)	-
Результат (Outcome)	Раскрытие данных (Disclosure)
Ссылка на каталог уязвимостей (Catalog reference)	-

Вторая фаза исследования системы в соответствии с методикой — идентификация инфраструктурных уязвимостей. В ходе этой фазы определяется инфраструктура, поддерживающая существование выделенного ранее актива (например, если это БД отдела кадров, то нам для работы с ней нужен сервер, на котором база размещена, рабочая станция служащего отдела кадров и т. д.), и то окружение, которое может позволить получить к ней доступ (например, соответствующий сегмент локальной сети). Рассматриваются компоненты следующих классов: серверы; сетевое оборудование; СЗИ; персональные компьютеры; домашние персональные компьютеры «надомных» пользователей, работающих удаленно, но имеющих доступ в сеть организации; мобильные компьютеры; системы хранения; беспроводные устройства; прочее.

Группа, проводящая анализ для каждого сегмента сети, отмечает, какие компоненты в нем проверяются на наличие уязвимостей. Уязвимости проверяются сканерами безопасности уровня операционной системы, сетевыми сканерами безопасности, специализированными сканерами (для конкретных web-серверов, СУБД и проч.), с по-

мощью списков уязвимостей (англ. «checklists»), тестовых скриптов. Для каждого компонента определяется:

- список уязвимостей «высокой степени важности» (англ. «high-severity vulnerabilities»), которые надо устранить немедленно;
- список уязвимостей «средней степени важности» (англ. «middle-severity vulnerabilities»), которые надо устранить в ближайшее время;
- список уязвимостей «низкой степени важности» (low-severity vulnerabilities), в отношении которых не требуется немедленных действий.

По результатам стадии готовится отчет, в котором указывается, какие уязвимости обнаружены, какое влияние они могут оказать на выделенные ранее активы, какие меры надо предпринять для устранения уязвимостей.

Разработка стратегии и планов безопасности — третья стадия исследования системы. Она начинается с оценки рисков, которая проводится на базе отчетов по двум предыдущим этапам. В OCTAVE при оценке риска дается только оценка ожидаемого ущерба, без оценки вероятности. Шкала: «высокий», «средний», «низкий». Оценивается финансовый ущерб, ущерб репутации компании, жизни и здоровью клиентов и сотрудников, ущерб, который может вызвать судебное преследование в результате того или иного инцидента. Описываются значения, соответствующие каждой градации шкалы (например, для малого предприятия финансовый ущерб в $ 10000 — высокий, для более крупного — средний).

Далее разрабатывают планы снижения рисков нескольких типов:

- долговременные;
- на среднюю перспективу;
- списки задач на ближайшее время.

Для определения мер противодействия угрозам в методике предлагаются каталоги средств.

Хотелось бы еще раз подчеркнуть, что в отличие от прочих методик OCTAVE не предполагает привлечения для исследования безопасности ИС сторонних экспертов, а вся документация по OCTAVE общедоступна и бесплатна, что делает методику особенно привлекательной для предприятий с жестко ограниченным бюджетом, выделяемым на цели обеспечения ИБ.

4.5.4. Методика RiskWatch

Компания RiskWatch разработала собственную методику анализа рисков и семейство программных средств, в которых она в той либо иной мере реализуется [19, 28, 29, 30].

В семейство RiskWatch входят программные продукты для проведения различных видов аудита безопасности:

- RiskWatch for Physical Security — для анализа физической защиты ИС;
- RiskWatch for Information Systems — для информационных рисков;
- HIPAA-WATCH for Healthcare Industry — для оценки соответствия требованиям стандарта HIPAA (от англ. «US Healthcare Insurance Portability and Accountability Act»), актуальным в основном для медицинских учреждений, работающих на территории США;
- RiskWatch RW17799 for ISO 17799 — для оценки соответствия ИС требованиям международного стандарта ISO 17799.

В методе RiskWatch в качестве критериев для оценки и управления рисками используются ожидаемые годовые потери (англ. «Annual Loss Expectancy», ALE) и оценка возврата инвестиций (англ. «Return on Investment», ROI). RiskWatch ориентирована на точную количественную оценку соотношения потерь от угроз безопасности и затрат на создание системы защиты. В основе продукта RiskWatch находится методика анализа рисков, которая состоит из четырех этапов.

Первый этап — определение предмета исследования. Здесь описываются такие параметры, как тип организации, состав исследуемой

системы (в общих чертах), базовые требования в области безопасности. Для облегчения работы аналитика в шаблонах, соответствующих типу организации («коммерческая информационная система», «государственная/военная информационная система» и т. д.), есть списки категорий защищаемых ресурсов, потерь, угроз, уязвимостей и мер защиты. Из них нужно выбрать те, что реально присутствуют в организации (рис. 4.11).

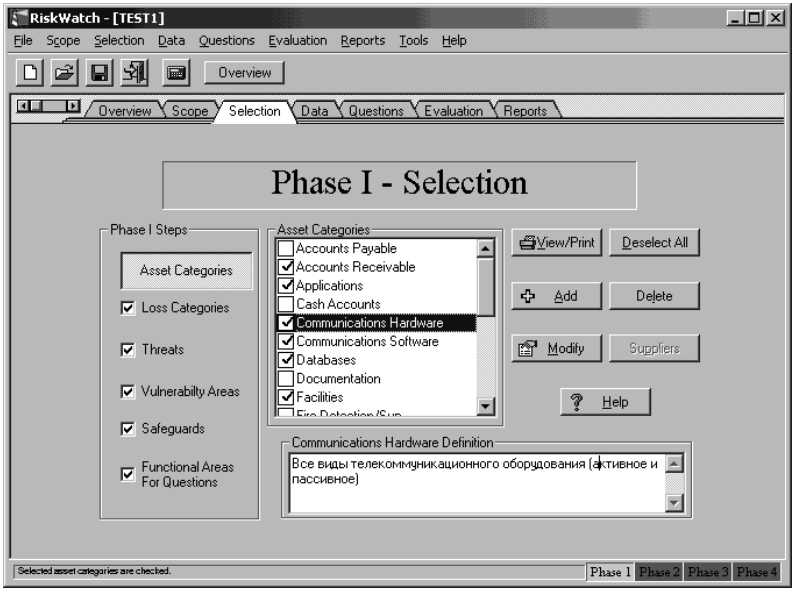


Рис. 4.11. Определение категорий защищаемых ресурсов

Например, категории потерь:

- задержки и отказ в обслуживании;
- раскрытие информации;
- прямые потери (например, от уничтожения оборудования огнем);
- жизнь и здоровье (персонала, заказчиков и т. д.);
- изменение данных;
- косвенные потери (например, затраты на восстановление);

- репутация.

Второй этап — ввод данных, описывающих конкретные характеристики системы. Данные могут вводиться вручную или импортироваться из отчетов, созданных инструментальными средствами исследования уязвимости компьютерных сетей. На этом этапе, в частности, подробно описываются ресурсы, потери и классы инцидентов. Классы инцидентов получаются путем сопоставления категории потерь и категории ресурсов.

Для выявления возможных уязвимостей используется вопросник, база которого содержит более 600 вопросов. Вопросы связаны с категориями ресурсов.

Также задается частота возникновения каждой из выделенных угроз, степень уязвимости и ценность ресурсов. Если для выбранного класса угроз в системе есть среднегодовые оценки возникновения (LAFE и SAFE), то используются они. Все это используется в дальнейшем для расчета эффекта от внедрения средств защиты.

Третий этап — количественная оценка риска. На этом этапе рассчитывается профиль рисков, и выбираются меры обеспечения безопасности. Сначала устанавливаются связи между ресурсами, потерями, угрозами и уязвимостями, выделенными на предыдущих шагах исследования. По сути, риск оценивается с помощью математического ожидания потерь за год. Например, если стоимость сервера $ 150000, а вероятность того, что он будет уничтожен пожаром в течение года, равна 0,01, то ожидаемые потери составят $ 1500.

Формула расчета ($m = p \times v$, где m — математическое ожидание, p — вероятность возникновения угрозы, v — стоимость ресурса) претерпела некоторые изменения в связи с тем, что RiskWatch использует определенные американским институтом стандартов NIST оценки, называемые LAFE и SAFE. LAFE (от англ. «Local Annual Frequency Estimate») показывает, сколько раз в год в среднем данная угроза реализуется в данном месте (например, в городе). SAFE (от англ. «Standard Annual Frequency Estimate») показывает, сколько раз в год в

среднем данная угроза реализуется в этой «части мира» (например, в Северной Америке). Вводится также поправочный коэффициент, который позволяет учесть, что в результате реализации угрозы защищаемый ресурс может быть уничтожен не полностью, а только частично.

Формулы (4.3) и (4.4) показывают варианты расчета показателя ALE

$$ALE = Asset\ Value \times Exposure\ Factor \times Frequency, \quad (4.3)$$

где *Asset Value* — стоимость рассматриваемого актива (данных, программ, аппаратуры и т. д.);

Exposure Factor — коэффициент воздействия — показывает, какая часть (в процентах) от стоимости актива подвергается риску;

Frequency — частота возникновения нежелательного события;

ALE — это оценка ожидаемых годовых потерь для одного конкретного актива от реализации одной угрозы.

Когда все активы и воздействия идентифицированы и собраны вместе, то появляется возможность оценить общий риск для ИС, как сумму всех частных значений.

Можно ввести показатели «ожидаемая годовая частота происшествия» (англ. «Annualized Rate of Occurrence» — ARO) и «ожидаемый единичный ущерб» (англ. «Single Loss Expectancy» — SLE), который может рассчитываться как разница первоначальной стоимости актива и его остаточной стоимости после происшествия (хотя подобный способ оценки применим не во всех случаях, например, он не подходит для оценки рисков, связанных с нарушением конфиденциальности информации). Тогда для отдельно взятого сочетания угроза-ресурс применима формула (4.4):

$$ALE = ARO \times SLE. \quad (4.4)$$

Дополнительно рассматриваются сценарии «что если:», которые позволяют описать аналогичные ситуации при условии внедрения средств защиты. Сравнивая ожидаемые потери при условии внедрения защитных мер и без них можно оценить эффект от таких мероприятий.

RiskWatch включает в себя базы с оценками LAFE и SAFE, а также с обобщенным описанием различных типов средств защиты.

Эффект от внедрения средств защиты количественно описывается с помощью показателя ROI (Return on Investment — возврат инвестиций), который показывает отдачу от сделанных инвестиций за определенный период времени. Рассчитывается он по формуле:

$$ROI = \sum_i NVP(Benefits_i) - \sum_j NVP(Costs_j), \qquad (4.5)$$

где $Costs_j$ — затраты на внедрение и поддержание j-меры защиты; $Benefits_i$ — оценка той пользы (т. е. ожидаемого снижения потерь), которую приносит внедрение данной меры защиты; *NPV* (Net Present Value) — чистая текущая стоимость.

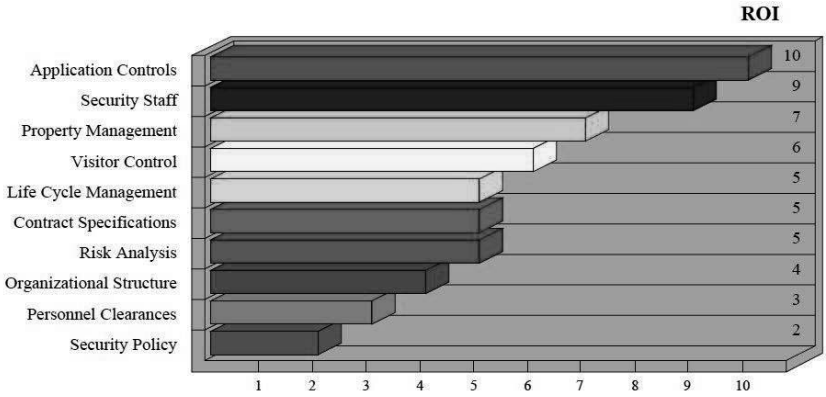


Рис. 4.12. Пример графика показателя ROI для различных мер защиты

Четвертый этап — генерация отчетов. Типы отчетов:

- краткие итоги;
- полные и краткие отчеты об элементах, описанных на стадиях 1 и 2;
- отчет от стоимости защищаемых ресурсов и ожидаемых потерях от реализации угроз;
- отчет об угрозах и мерах противодействия;
- отчет о ROI (фрагмент приведен на рис. 4.12);

- отчет о результатах аудита безопасности.

Таким образом, рассматриваемое средство позволяет оценить не только те риски, которые сейчас существуют у предприятия, но и ту выгоду, которую может принести внедрение физических, технических, программных и прочих средств и механизмов защиты. Подготовленные отчеты и графики дают материал, достаточный для принятия решений об изменении системы обеспечения безопасности предприятия.

4.5.5. Проведение оценки рисков в соответствии с методикой Майкрософт

Процесс управления рисками, предлагаемый корпорацией Майкрософт [20], разбивает этап оценки рисков на следующие три шага:

- *планирование*. Разработка основы для успешной оценки рисков;
- *координированный сбор данных*. Сбор информации о рисках в ходе координированных обсуждений рисков;
- *приоритизация рисков*. Ранжирование выявленных рисков на основе непротиворечивого и повторяемого процесса.

Для проведения оценки требуется собрать данные о:

- активах организации;
- угрозах безопасности;
- уязвимостях;
- текущей среде контроля (в принятой авторами перевода руководства [20] терминологии средства и меры защиты информации называются элементами контроля, соответственно, среда контроля — совокупность элементов);
- предлагаемых элементах контроля.

Активами считается все, что представляет ценность для организации. К материальным активам относится физическая инфраструктура (например, центры обработки данных, серверы и имущество). К нематериальным активам относятся данные и другая ценная для организации информация, хранящаяся в цифровой форме (например, бан-

ковские транзакции, расчеты платежей, спецификации и планы разработки продуктов). В некоторых организациях может оказаться полезным определение третьего типа активов — ИТ-служб. ИТ-служба представляет собой сочетание материальных и нематериальных активов. Например, это может быть корпоративная служба электронной почты.

Процесс управления рисками безопасности, предлагаемый корпорацией Майкрософт, определяет следующие три качественных класса активов:

- *высокое влияние на бизнес* (ВВБ) — влияние на конфиденциальность, целостность и доступность этих активов может причинить организации значительный или катастрофический ущерб. Например, к этому классу относятся конфиденциальные деловые данные;
- *среднее влияние на бизнес* (СВБ) — влияние на конфиденциальность, целостность и доступность этих активов может причинить организации средний ущерб. Средний ущерб не вызывает значительных или катастрофических изменений, однако нарушает нормальную работу организации до такой степени, что это требует проактивных элементов контроля для минимизации влияния в данном классе активов. К этому классу могут относиться внутренние коммерческие данные, такие как перечень сотрудников или данные о заказах предприятия;
- *низкое влияние на бизнес* (НВБ) — активы, не попадающие в классы ВВБ и СВБ, относятся к классу НВБ. К защите подобных активов не выдвигаются формальные требования, и она не требует дополнительного контроля, выходящего за рамки стандартных рекомендаций по защите инфраструктуры. Например, это могут быть общие сведения о структуре организации.

Далее определяется перечень угроз и уязвимостей, и выполняется оценка уровня потенциального ущерба, называемого степенью

подверженности актива воздействию. Оценка ущерба может проводиться по различным категориям:

- конкурентное преимущество;
- законы и регулятивные требования;
- операционная доступность;
- репутация на рынке.

Оценку предлагается проводить по следующей шкале:

- «высокая подверженность воздействию»: значительный или полный ущерб для актива;
- «средняя подверженность воздействию»: средний или ограниченный ущерб;
- «низкая подверженность воздействию»: незначительный ущерб или отсутствие такового.

Следующий шаг — оценка частоты возникновения угроз по шкале:

- «высокая»: вероятно возникновение одного или нескольких событий в пределах года.
- «средняя»: влияние (событие) может возникнуть в пределах двух-трех лет.
- «низкая»: возникновение влияния в пределах трех лет маловероятно.

Шаблон сбора данных

Определите активы, за разработку, поддержку, управление и сопровождение которых несет ответственность ваша группа

Название актива	Классификация актива (высокое, среднее или низкое влияние на деятельность)
1.	

Для каждого актива укажите следующие значения

Многоуровневая защита	Чего необходимо избежать (угрозы)	Пути возникновения (уязвимости)	Уровень подверженности воздействию (В, С, Н)	**Описания текущих элементов контроля**	Вероятность (В, С, Н)	**Назначение контроля, потенциальные новые**
Физический уровень						
Приложения						
Узлы						
Сеть						
Данные						

Рис. 4.13. Шаблон сбора данных

Данные собираются в шаблон (см. рис. 4.13). Набор шаблонов (в виде файлов Excel) для проведения анализа рисков доступен вместе с

текстом руководства на сайте Microsoft. Для пояснения методики ниже будут приводиться скриншоты, отображающие разные этапы заполнения шаблонов.

Для угроз указывается уровень воздействия в соответствии с концепцией многоуровневой защиты (уровни — физический, сети, хоста, приложения, данных).

В столбце текущие элементы контроля описывают использующиеся средства и меры защиты, противостоящие данной угрозе. На основе собранных данных заполняется таблица, пример которой представлен на рис. 4.14.

Актив				Подверженность воздействию			
Дата обнаружения	Название актива	Класс актива	Применимые уровни многоуровневой защиты	Описание угрозы	Описание уязвимости	Уровень подверженности воздействию (В, С, Н)	Уровень влияния (В, С, Н)
Дата	Информация о финансовых инвестициях заказчиков	ВВБ	Узлы	Несанкционированный доступ к информации о заказчиках путем хищения учетных данных консультантов по финансовым вопросам	Хищение учетных данных с управляемых компьютеров локальной сети вследствие несвоевременного обновления баз данных антивирусных средств или конфигураций узлов либо несвоевременной установки обновлений системы безопасности	С	В
Дата	Информация о финансовых инвестициях заказчиков	ВВБ	Узлы	Несанкционированный доступ к информации о заказчиках путем хищения учетных данных консультантов по финансовым вопросам	Хищение учетных данных с управляемых удаленных компьютеров вследствие несвоевременного обновления баз данных антивирусных средств или конфигураций узлов либо несвоевременной установки обновлений системы безопасности	С	В
Дата	Информация о финансовых инвестициях заказчиков	ВВБ	Данные	Несанкционированный доступ к информации о заказчиках путем хищения учетных данных консультантов по финансовым вопросам	Хищение учетных данных доверенным сотрудником с помощью подслушивания, методов социальной инженерии и других методов без использования технических средств	Н	С

Рис. 4.14. Пример заполненного шаблона

Следующий шаг этапа оценки рисков — приоритизация рисков, т. е. создание упорядоченного по приоритетам списка рисков. Формирование данного списка сначала предлагается выполнить на обобщенном уровне, после чего описания наиболее существенных рисков детализируются.

Исходя из значения класса актива и оценки подверженности актива воздействию по таблице, приведенной на рис. 4.15, определяется уровень влияния.

		Образец подверженности воздействию		
Класс актива	Выс.	Средн.	Выс.	Выс.
	Средн.	Низк.	Средн.	Выс.
	Низк.	Низк.	Низк.	Средн.
		Низк.	Средн.	Выс.
		Уровень подверженности воздействию		

Рис. 4.15. Определение уровня влияния по классу актива и уровню подверженности воздействию

Итоговый уровень риска определяется исходя из уровня влияния и оценки частоты возникновения риска (рис. 4.16). Полученные оценки заносятся в таблицу, пример которой приведен на рис. 4.17.

		Уровни в списке с обобщенными сведениями о рисках		
Влияние (из предыдущей таблицы)	Выс.	Средн.	Выс.	Выс.
	Средн.	Низк.	Средн.	Выс.
	Низк.	Низк.	Низк.	Средн.
		Низк.	Средн.	Выс.
		Уровень вероятности		

Рис. 4.16. Определение итогового уровня риска

Для детального изучения (составления «перечня на уровне детализации») отбираются риски, отнесенные по результатам оценки на обобщенном уровне к одной из трех групп:

- риски высокого уровня;
- граничные риски: риски среднего уровня, которые необходимо снижать;
- противоречивые риски: риск является новым и знаний об этом риске у организации недостаточно или различные заинтересованные лица оценивают этот риск по-разному.

Информация, полученная в ходе процесса сбора данных						
Актив				Подверженность воздействию		
Дата обнаружения	Название актива	Класс актива	Применимые уровни многоуровневой защиты	Описание угрозы	Описание уязвимости	Уро подвер возде (В,
пример	Информация о финансовых инвестициях заказчиков	ВВБ	Узел	Несанкционированный доступ к информации о заказчиках путем хищения учетных данных консультантов по финансовым вопросам	Хищение учетных данных с управляемых компьютеров локальной сети вследствие несвоевременного обновления баз данных антивирусных средств или конфигураций узлов либо несвоевременной установки обновлений системы безопасности	
пример	Информация о финансовых инвестициях заказчиков	ВВБ	Узел	Несанкционированный доступ к информации о заказчиках путем хищения учетных данных консультантов по финансовым вопросам	Хищение учетных данных с управляемых удаленных компьютеров вследствие несвоевременного обновления баз данных антивирусных средств или конфигураций узлов либо несвоевременной установки обновлений системы безопасности	
пример	Информация о финансовых инвестициях заказчиков	ВВБ	Данные	Несанкционированный доступ к информации о заказчиках путем хищения учетных данных консультантов по финансовым вопросам	Хищение учетных данных доверенным сотрудником с помощью подслушивания, методов социальной инженерии и других методов без использования технических средств	

сти	Уровень подверженности воздействию (В, С, Н)	Уровень влияния (В, С, Н)	Вероятность (В, С, Н)	Обобщенный уровень риска (В, С, Н)
емых компьютеров своевременного усных средств или временной установки асности	С	В	**С**	**В**
емых удаленных евременного ных средств или еменной установки пасности	С	В	**В**	**В**
ным сотрудником с дов социальной использования тв	Н	С	**Н**	**Н**

Рис. 4.17. Пример перечня рисков на обобщенном уровне

Формирование перечня рисков на уровне детализации является последней задачей процесса оценки рисков. В этом перечне каждому риску в итоге сопоставляется оценка в числовой (денежной) форме. Вновь определяются:

- величина влияния и подверженности воздействию;
- текущие элементы контроля;
- вероятности влияния;
- уровень риска.

Уровень подверженности воздействию оценивается по пятибалльной шкале. Шкала для угрозы целостности и конфиденциальности приведена на рис. 4.18, а для угрозы отказа в обслуживании — на рис. 4.19. В качестве итогового уровня подверженности воздействию предлагается выбрать максимальное значение.

Уровень подверженности воздействию	Конфиденциальность или целостность актива
5	Серьезные повреждения или полный выход актива из строя (например, видимые снаружи и влияющие на прибыльность или успешность ведения бизнеса)
4	Серьезные повреждения, не приводящие к полному выходу актива из строя (например, влияющие на прибыльность или успешность ведения бизнеса и, возможно, видимые снаружи)
3	Средние повреждения или ущерб (например, влияющие на внутренние рекомендации по ведению бизнеса и способные вызвать увеличение эксплуатационных затрат или уменьшение доходов)
2	Незначительные повреждения или ущерб (например, влияющие на внутренние рекомендации по ведению бизнеса, но не вызывающие существенного роста затрат)
1	Небольшие изменения в активе или отсутствие изменений

Рис. 4.18. Уровни подверженности воздействию для угроз конфиденциальности и целостности

Уровень подверженности воздействию	Дата выпуска	Описание
5	Прекращение работы	Большие эксплуатационные затраты или нарушение коммерческих обязательств
4	Прерывание работы	Значительное увеличение эксплуатационных затрат или задержка при выполнении коммерческих обязательств
3	Задержки в работе	Заметное влияние на величину эксплуатационных затрат и производительность.
2	Отвлечение от работы	Измеримое влияние на деятельность компании отсутствует; небольшое увеличение эксплуатационных затрат или затрат на инфраструктуру
1	Не влияет на обычный ход бизнес-операций	Измеримое влияние на эксплуатационные затраты, производительность и коммерческие обязательства отсутствует

Рис. 4.19. Уровни подверженности воздействию для доступности

После определения уровня подверженности воздействию производится оценка величины влияния. Каждому уровню подверженности воздействию сопоставляется значение в процентах, отражающее величину ущерба, причиненного активу, и называемое фактором подверженности воздействию. Майкрософт рекомендует использовать линейную шкалу подверженности воздействию от 100 до 20 %, которая может изменяться в соответствии с требованиями организации. Кроме того, каждой величине влияния сопоставляется качественная оценка: высокая, средняя или низкая. На рис. 4.20 показаны возможные значения для каждого класса влияния.

Класс влияния	Значение класса влияния (З)
ВВБ	10
СВБ	5
НВБ	2

Уровень подверженности воздействию	Фактор подверженности воздействию (ФПВ)	Уровень влияния (З * ФПВ)	Диапазон влияния	Обобщенное сравнение
5	100%		7 - 10	Выс.
4	80%		4 - 6	Средн.
3	60%		0 - 3	Низк.
2	40%			
1	20%			

Рис. 4.20. Определение величин влияния

Далее описываются «элементы контроля», используемые в организации для снижения вероятностей угроз и уязвимостей, определенных в формулировке влияния.

Следующая задача — определение вероятности влияния. Результирующий уровень вероятности определяется на основании двух значений. Первое значение определяет вероятность существования уязвимости в текущей среде. Второе значение определяет вероятность существования уязвимости, исходя из эффективности текущих элементов контроля. Каждое значение изменяется в диапазоне от 1 до 5. Определение оценки проводится на основе ответов на вопросы, перечень которых представлен на рис. 4.21, с последующим переходом к результирующей оценке (рис. 4.22).

Определения вероятностей для уязвимостей
Высокая
Большое число злоумышленников — любители и компьютерные хулиганы
Удаленное выполнение
Возможность использования анонимного доступа
Общеизвестный метод взлома
Автоматизированность
5, если выполняется хотя бы одно из условий
Средняя
Среднее число злоумышленников — специалисты и эксперты
Невозможность удаленного выполнения
Необходимость наличия привилегий уровня пользователя
Метод взлома не является общеизвестным
Атака не автоматизирована
3, если выполняется хотя бы одно из условий
Низкая
Небольшое число злоумышленников — необходима внутренняя информация
Невозможность удаленного выполнения
Необходимость наличия привилегий уровня администратора
Метод взлома не является общеизвестным
Атака не автоматизирована
1, если выполняются все условия

Рис. 4.21. Оценка уязвимости

Результирующая оценка уязвимости	
Атрибуты подверженности воздействию (выберите из числа указанных выше)	
высокая	5
средняя	3
низкая	1
уровень вероятности (1, 3 или 5)	

Рис. 4.22. Оценка уровня вероятности

При этом разработчики руководства указывают, что оценка вероятности взлома имеет субъективный характер, и предлагают при проведении оценки уточнять приведенный перечень.

Насколько эффективны текущие элементы контроля?	
Да — 0, Нет — 1	
Эффективно ли определена и реализована ответственность?	1,0
Эффективно ли осуществляется информирование?	1,0
Эффективно ли определены и реализованы процессы?	1,0
Эффективно ли существующие технологии или элементы контроля снижают угрозы?	1,0
Обеспечивают ли существующие методы аудита обнаружение злоупотреблений и недостатка контроля?	1,0
Сумма атрибутов контроля (0–5) =	

Рис. 4.23. Оценка эффективности текущего контроля

На рис. 4.23 приведена шкала оценки эффективности текущих мер и средств защиты. Меньший результат означает большую эффективность элементов контроля и их способность уменьшать вероятность взлома.

Полученные значения суммируются и заносятся в шаблон для уровня детализации. Пример заполненного шаблона представлен на рис. 4.24 (на рисунке в предпоследнем столбце первой строки следует читать «Уязвимость: 5, Контроль: 1», в предпоследнем столбце второй строки — «Уязвимость: 5, Контроль: 5»).

Уровень влияния × Уровень вероятности = Уровень риска

Диапазоны уровня влияния		**Диапазоны вероятности**	
Выс.	10 -- 7	10 -- 7	
Средн.	6 -- 4	6 -- 4	
Низк.	3 -- 0	3 -- 0	

	В	10	0	10	20	30	40	50	60	70	80	90	100
		9	0	9	18	27	36	45	54	63	72	81	90
		8	0	8	16	24	32	40	48	56	64	72	80
		7	0	7	14	21	28	35	42	49	56	63	70
Влияние		6	0	6	12	18	24	30	36	42	48	54	60
	С	5	0	5	10	15	20	25	30	35	40	45	50
		4	0	4	8	12	16	20	24	28	32	36	40
		3	0	3	6	9	12	15	18	21	24	27	30
		2	0	2	4	6	8	10	12	14	16	18	20
	Н	1	0	1	2	3	4	5	6	7	8	9	10
			0	1	2	3	4	5	6	7	8	9	10
			Н					**С**					**В**
							Вероятность						

Общий риск	**Уровень риска**
	Выс.
	Средн.
	Низк.

Рис. 4.25. Результирующее качественное ранжирование

На рис. 4.24 показаны уровни риска и соответствующие элементы данных. Уровень риска определяется как произведение оценок уровня влияния (со значением от 1 до 10) и уровня вероятности (со значением от 0 до 10). В результате уровень риска может принимать значения от 0 до 100. Переход от числовой оценки к оценке по шкале «высокий», «средний» или «низкий» можно сделать в соответствии с таблицей, представленной на рис. 4.25.

Базовый риск (текущий)									
Актив		Подверженность воздействию							
Название актива	Уровень класса влияния	Многоуровневая защита	Описание угрозы	Описание уязвимости	Уровень подверженности воздействию (1–5)	Уровень подверженности воздействию (1–10)	Описания текущих элементов контроля	Уровень вероятности с контролем (1–10)	Уровень риска с контролем (0–100)
Информация о финансовых инвестициях заказчиков	10 (ВВБ)	Узлы	Несанкционированный доступ к информации о заказчиках путем хищения учетных данных консультантов по финансовым вопросам	Хищение учетных данных с управляемых компьютеров локальной сети вследствие несвоевременного обновления баз данных антивирусных средств или конфигураций узлов либо несвоевременной установки обновлений системы безопасности	4 (80%)	8	**1.** Каждый консультант имеет доступ только к информации о своих клиентах. Таким образом, подверженность воздействию составляет менее 100%. **2.** Уведомления об обновлениях и исправлениях, отправляемые по электронной почте. **3.** В локальной сети каждые несколько часов выполняется установка требуемых обновлений, что уменьшает временной интервал, в течение которого узлы локальной сети уязвимы перед взломом.	Уязвимость: Контроль: 1 Всего = 6	48
Информация о финансовых инвестициях заказчиков	10 (ВВБ)	Узлы	Несанкционированный доступ к информации о заказчиках путем хищения учетных данных консультантов по финансовым вопросам	Хищение учетных данных с управляемых удаленных компьютеров вследствие несвоевременного обновления баз данных антивирусных средств или конфигураций узлов либо несвоевременной установки обновлений системы безопасности	4 (80%)	8	**1.** Каждый консультант имеет доступ только к информации о своих клиентах. Таким образом, подверженность воздействию составляет менее 100%. **2.** Уведомления об обновлениях и исправлениях, отправляемые по электронной почте. — Отсутствует решение, позволяющее обеспечить соответствие требованиям за пределами локальной сети.	Контроль: 5 Всего = 10	80

Рис. 4.24. Перечень рисков на уровне детализации

В заключение процедуры оценки рисков проводится количественный анализ. Чтобы определить количественные характеристики, необходимо выполнить следующие задачи:

- сопоставить каждому классу активов в организации денежную стоимость;
- определить стоимость актива для каждого риска;
- определить величину ожидаемого разового ущерба (англ. «single loss expectancy» — SLE);
- определить ежегодную частоту возникновения (англ. «annual rate of occurrence» — ARO);
- определить ожидаемый годовой ущерб (англ. «annual loss expectancy» — ALE).

Количественную оценку предлагается начать с активов, соответствующих описанию класса ВВБ. Для каждого актива определяется денежная стоимость с точки зрения его материальной и нематериальной ценности для организации. Также учитываются:

- стоимость замены;
- затраты на обслуживание и поддержание работоспособности;
- затраты на обеспечение избыточности и доступности;
- влияние на репутацию организации;
- влияние на эффективность работы организации;
- годовой доход;
- конкурентное преимущество;
- внутренняя эффективность эксплуатации;
- правовая и регулятивная ответственность.

Процесс повторяется для каждого актива в классах СВБ и НВБ.

Каждому классу активов сопоставляется одно денежное значение, которое будет представлять ценность класса активов. Например, наименьшее среди активов данного класса. Данный подход уменьшает затраты времени на обсуждение стоимости конкретных активов.

После определения стоимостей классов активов необходимо определить и выбрать стоимость каждого риска.

Следующей задачей является определение степени ущерба, который может быть причинен активу. Для расчетов предлагается использовать ранее определенный уровень подверженности воздействию, на основе которого определяется фактор подверженности воздействию (рекомендуемая формула пересчета — умножение значения уровня в баллах на 20 %, см. рис. 4.26).

Величина высокого влияния на деятельность = $ M			Уровень подверженности воздействию	Фактор подверженности воздействию, %
			5	100
Класс актива			4	80
Значение ВВБ	**$ M**		3	60
Значение СВБ	**$ M/2**		2	40
Значение НВБ	**$ M/4**		1	20
Оценочное значение риска =	Значение класса актива × Фактор подверженности воздействию (%) = Ожидаемый разовый ущерб			

Рис. 4.26. Количественная оценка ожидаемого разового ущерба

Последний шаг состоит в получении количественной оценки влияния путем умножения стоимости актива на фактор подверженности воздействию. В классической количественной модели оценки рисков это значение называется величиной ожидаемого разового ущерба (SLE). На рис. 4.27 приведен пример реализации такого подхода.

Описание риска	Значение класса актива	Уровень подверженности воздействию	Величина подверженности воздействию	Ожидаемый разовый ущерб
Риск для узла локальной сети	$ 10	4	80%	$ 8
Риск для удаленного узла	$ 10	4	80%	$ 8

Рис. 4.27. Пример определения ожидаемого разового ущерба (в примере суммы указаны в миллионах долларов)

Далее делается оценка ежегодной частоты возникновения (ARO). В процессе оценки ARO используются ранее полученные качественные оценки рис. 4.28.

Качественный уровень	Описание	Диапазон ежегодной частоты возникновения	Примеры описаний
Высокий	Очень вероятно	>= 1	Влияние раз в год или чаще
Средний	Вероятно	От 0,99 до 0,33	Не менее одного раза каждые 1–3 года
Низкий	Маловероятно	< 0,33	Реже, чем один раз в 3 года

Рис. 4.28. Количественная оценка ежегодной частоты возникновения

Для определения ожидаемого годового ущерба (ALE) значения SLE и ARO перемножаются:

$$ALE = SLE \times ARO. \quad (4.5)$$

Величина ALE характеризует потенциальные годовые убытки от риска. Хотя данный показатель может помочь в оценке ущерба заинтересованным лицам, имеющим финансовую подготовку, группа управления рисками безопасности должна напомнить, что влияние на организацию не ограничивается величиной годовых издержек — возникновение риска может повлечь за собой причинение ущерба в полном объеме.

Подводя итог, можно еще раз отметить, что процесс управления рисками безопасности, предлагаемый корпорацией Майкрософт, использует комбинированный подход, включающий оценку рисков на качественном уровне на начальном этапе и количественную оценку — на заключительном.

4.5.6. Анализ существующих подходов

Подводя итог обзору методик, перечислим те преимущества, которые дает проведение анализа рисков в сфере ИБ:

- выявление проблем в сфере безопасности (не только уязвимостей компонент системы, но и недостатков политик безопасности и т. д.);
- анализ рисков позволяет нетехническим специалистам (в частности, руководству организации) оценить выгоды от внедрения средств и механизмов защиты и принять участие в процессе определения требуемого уровня защищенности КС;

- проведение оценки рисков добавляет обоснованность рекомендациям по безопасности;
- ранжирование рисков по приоритетам позволяет выделить наиболее приоритетные направления для внедрения новых СЗИ, мер и процедур обеспечения ИБ;
- подробно описанные методики анализа рисков позволяет людям, не являющимся экспертами в данной области, воспользоваться аккумулированными в методике знаниями, чтобы получить заслуживающие доверия результаты анализа.

В то же время необходимо отметить, что оценка рисков на качественном уровне не позволяет однозначно сравнить затраты на обеспечение ИБ и получаемую от них отдачу (в виде снижения суммарного риска). Поэтому более предпочтительными представляются количественные методики. Но они требуют наличия оценок вероятности возникновения для каждой из рассматриваемых угроз безопасности. Кроме того, использование интегральных показателей, таких как ALE, опасно тем, что неправильная оценка вероятности угрозы в отношении очень дорогостоящего актива может кардинально изменить оцениваемое значение суммарной стоимости рисков.

Представляется более обоснованным формирование не единой, скалярной оценки стоимости риска, а векторной. Каждый элемент вектора оценки соответствует обобщенной угрозе безопасности (подмножеству угроз, сходных по способу реализации и оказываемому на систему воздействию).

Также предлагается проводить оценку риска, исходя из анализа модели конфликта интересов владельца системы и нарушителя ее безопасности. Данная модель должна строиться с использованием математического аппарата, допускающего отсутствие статистики относительно частоты возникновения угроз безопасности. Подобная модель описывается в следующем разделе пособия.

4.6. ВЫБОР ПРОЕКТА СИСТЕМЫ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ. ИГРОВАЯ МОДЕЛЬ КОНФЛИКТА «ЗАЩИТНИК-НАРУШИТЕЛЬ»

Будем предполагать, что в ходе предварительного исследования было сформировано множество проектов подсистемы защиты информации, отвечающих предъявляемым функциональным требованиям и стоимостным ограничениям. И теперь необходимо выбрать из этих проектов наилучший по соотношению затрат и получаемого эффекта. Рассмотрим один из возможных подходов к решению данной задачи.

Выбор предлагается проводить с помощью математического аппарата теории игр, а если точнее — используя одношаговые конечные игры двух игроков (матричные игры) [31]. Использование данного класса моделей определяется следующими свойствами решаемой задачи и допущениями относительно ее характера:

- имеется конфликт двух участников: в качестве первого игрока будем рассматривать владельца или защитника системы, в качестве второго — условный источник всех угроз, как преднамеренных, так и непреднамеренного характера;
- присутствует фактор неопределенности и отсутствует достоверная статистика (заранее неизвестно, какие угрозы будут реализованы, и как часто будут происходить нежелательные события);
- у каждого участника имеется конечное множество альтернатив (чтобы выполнить это требование, множество угроз безопасности предлагается разбить на подмножества, именуемые обобщенными угрозами);
- можно количественно оценить каждый исход игры;
- действия сторон являются однократными или могут быть сведены к суммарному однократному воздействию (выбор проекта производится только один раз).

Рассмотрим задание игры. Стратегии игрока I («защитника») заключаются во внедрении в ИС одного из проектов защиты или отказе

от каких-либо действий. Обозначим множество проектов подсистемы защиты ИС через **C**, а текущее состояние системы как $\hat{C}$. Тогда «защитник» будет выбирать стратегии, соответствующие элементам множества **C**∪{ }.

Обозначим через **U** конечное множество обобщенных угроз информационной безопасности защищаемой ИС. Обобщенная угроза — это подмножество угроз ИБ, сходных по оказываемому на ИС воздействию и причиняемому ущербу. Подобное разбиение множества угроз ИБ формируется на базе экспертных оценок. Тогда игровая стратегия «нарушителя» (игрока II) — выбор элемента из **U**∪{$\hat{U}$}, где — отказ от реализации угроз ИБ. В данной модели «нарушитель» рассматривается как источник всех угроз безопасности: и преднамеренных, и случайных. Конечная одношаговая антагонистическая игра (матричная игра) задается в виде:

$$\Gamma = <\mathbf{X}, \mathbf{Y}, H>, \qquad (4.6)$$

где **X** — множество стратегий «защитника», **X = C∪{ }**; **Y** — множество стратегий «нарушителя», **Y = U ∪ { }**; **H** — матрица выигрышей. При |**X**| = m, |**Y**| = n предлагается использовать матрицу выигрышей следующего вида (перед началом строк и над столбцами указаны соответствующие элементы множеств **X** и **Y**):

$$H = \begin{array}{c} \\ C_1 \\ \dots \\ C_{(m-1)} \\ \hat{C} \end{array} \begin{array}{c} \begin{array}{cccc} U_1 & \dots & U_{(n-1)} & \hat{U} \end{array} \\ \begin{bmatrix} -h_1 - \bar{h}_{11} & \dots & -h_1 - \bar{h}_{1(n-1)} & -h_1 \\ \dots & \dots & \dots & \dots \\ -h_{(m-1)} - \bar{h}_{(m-1)1} & \dots & -h_{(m-1)} - \bar{h}_{(m-1)(n-1)} & -h_{(m-1)} \\ -\bar{h}_{m1} & \dots & -\bar{h}_{m(n-1)} & 0 \end{bmatrix} \end{array}, \qquad (4.7)$$

где $\bar{h}_{ij}$ — оценки потерь от реализации «нарушителем» *j*-й обобщенной угрозы в отношении ИС, где реализован *i*-й проект подсистемы защиты; h_i — затраты на реализацию *i*-го проекта. Обе составляющие берутся со знаком минус, т. к. для «защитника» это потери (отрицательный выигрыш).

Построенная антагонистическая игра отражает ситуацию наиболее пессимистичного прогноза: предполагается, что «нарушитель»

всемогущ и имеет цель нанести максимальный вред. Если можно достоверно оценить возможности «нарушителя» и ценность для него результатов атаки на ИС, то предлагается использовать биматричные игровые модели. Биматричная игра определяется следующим образом:

$$\Gamma = < \mathbf{X}, \mathbf{Y}, H, H_2 >, \qquad (4.8)$$

где **X** и **Y** — множества стратегий игроков I и II, **H** — матрица выигрышей «защитника», $\mathbf{H_2}$ — матрица выигрышей «нарушителя»:

$$H_2 = \begin{array}{c} \\ C_1 \\ \dots \\ C_{(m-1)} \\ \widehat{C} \end{array} \begin{array}{c} \begin{array}{cccc} U_1 & \dots & U_{(n-1)} & \widehat{U} \end{array} \\ \begin{bmatrix} \tilde{h}_{11} - \hat{h}_{11} & \dots & \tilde{h}_{1(n-1)} - \hat{h}_{1(n-1)} & 0 \\ \dots & \dots & \dots & \dots \\ \tilde{h}_{(m-1)1} - \hat{h}_{(m-1)1} & \dots & \tilde{h}_{(m-1)(n-1)} - \hat{h}_{(m-1)(n-1)} & 0 \\ \tilde{h}_{m1} - \hat{h}_{m1} & \dots & \tilde{h}_{m(n-1)} - \hat{h}_{m(n-1)} & 0 \end{bmatrix} \end{array}, \qquad (4.9)$$

где $\tilde{h}_{ij}$ — оценка выигрыша «нарушителя» от реализации *j*-й угрозы в отношении ИС, где реализован *i*-й проект; $\hat{h}_{ij}$ — оценка затрат «нарушителя» на реализацию этой угрозы. Для угроз, источниками которых являются случайные события (например, сбой оборудования), принимаем = 0, а равным по модулю соответствующему элементу матрицы выигрышей игрока I. Нули в последнем столбце матрицы $\mathbf{H_2}$ соответствуют отказу от атаки на ИС. Данная модель отражает менее пессимистичный прогноз, основанный на наличии некоторых дополнительных знаний о «нарушителе».

Если для построенной игры существует решение в чистых стратегиях, то это указывает наиболее предпочтительный проект (или проекты) подсистемы защиты. Значение игры покажет максимальные ожидаемые потери при реализации наилучшего проекта. Если решение существует только в смешанных стратегиях, оно нуждается в дополнительной интерпретации: в реальной ИС невозможно поочередно использовать различные проекты защиты, как это предполагается определением смешанной стратегии. В этом случае можно отобрать проекты, попавшие в спектр оптимальной стратегии, и попытаться

сформировать компромиссный вариант, объединяющий их сильные стороны.

Достоинством подобной игровой модели «защитник-нарушитель» является то, что она позволяет учесть не только стоимость, но и особенности внедряемого проекта (через изменение оценок ожидаемых потерь).

5. ПРАКТИКУМ ПО ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

5.1. УПРАВЛЕНИЕ ДОСТУПОМ К ФАЙЛАМ НА NTFS

Цель работы.

Приобретение практических навыков настройки разрешений на доступ к файлам в операционных системах семейства Windows.

Используемые программные средства.

Компьютер или виртуальная машина с установленной ОС Windows 7 или Windows Server 2008. Для выполнения работы необходимы права администратора.

Теоретические сведения.

Операционные системы семейства Windows NT, в которое входят и Windows 7 и 8, обладают достаточно развитыми встроенными механизмами защиты информации. При соответствующей настройке они могут быть эффективно использованы для достижения различных целей безопасности.

Задачи аутентификации пользователей и разграничения доступа к данным взаимосвязаны: эффективного разграничения доступа не может быть без надежной аутентификации. ОС Windows позволяет создавать учетные записи пользователей и группы в локальной базе безопасности, а если компьютер включен в домен, то появляется возможность использовать и доменные учетные записи и группы, которые хранятся в базе Active Directory.

Допустим, компьютер с именем `COMP1` включен в домен `KAFEDRA`. Тогда полное имя учетной записи `User`, если она является локальной, будет `COMP1\User`, если доменной — `KAFEDRA\User`. Начиная с Windows 2000, стал поддерживаться и формат, разделяющий имя пользователя и полное имя домена символом «@». Пусть полное имя домена будет `KAFEDRA.mydomain.ru`, тогда имя пользователя может быть записано как `User@KAFEDRA.mydomain.ru`. В ОС Windows имена пользователей и доменов не чувствительны к регистру: имена учетной записи `user`, `User` и `USER` будут равнозначны. Значение пароля чувствительно к регистру.

Пользователи, группы пользователей и компьютеры в домене Windows имеют уникальные идентификаторы безопасности — SID (Security ID). SID имеет уникальное значение в пределах домена и формируется во время создания пользователя или группы, либо когда компьютер регистрируется в домене. SID сохранится и в том случае, если учетную запись переименовать. Вместе с ним сохранятся и все назначенные разрешения (см. далее). А вот удаление учетной записи и создание новой с тем же именем приведет к изменению SID.

Когда пользователь при входе в систему вводит имя и пароль, ОС выполняет проверку правильности пароля и, если она проходит, создает маркер доступа для пользователя. Маркер включает в себя SID пользователя и все SID'ы групп, в которые данный пользователь входит.

Для объектов, подлежащих защите, таких как файлы и папки, создается дескриптор безопасности. С ним связывается список управления доступом (*англ.* Access Control List — ACL), который содержит информацию о том, каким субъектам даны те или иные права на доступ к данному объекту. Чтобы определить, можно ли предоставить запрашиваемый субъектом тип доступа к объекту, ОС сравнивает SID в маркере доступа субъекта с идентификаторами, содержащимися в ACL.

Для каждого из предусмотренных типов доступа к файлу или папке (чтение, запись и т. д.) пользователю или группе может быть установлено «разрешить» (*англ.* allow), «запретить» (*англ.* deny) или разрешение может быть не установлено. Разрешения суммируются. Например, если у пользователя есть право на чтение, а у группы, куда он входит — на запись, то действующее (или эффективное) разрешение будет включать чтение и запись. Явное запрещение (deny) более приоритетно, чем аналогичное разрешение: например, если у пользователя есть разрешение на операцию, а одной из групп, в которые он входит, эта операция явно запрещена, пользователю эту операцию выполнить будет нельзя. Выполнить действие можно только в том случае, если оно разрешено, то есть отсутствие разрешения в итоговом («эффективном») наборе разрешений пользователя приведет к тому, что он не сможет выполнить соответствующее действие.

Если говорить о файлах и папках, то механизмы защиты на уровне файловой системы поддерживаются только на дисках с файловой системой NTFS. Файловая система FAT (и ее разновидность — FAT32) не предполагает наличия атрибутов безопасности файла или связанного с файлом ACL. Соответственно, нельзя установить разрешения на файл или папку, находящуюся на диске с FAT.

План лабораторной работы.

Лабораторная работа выполняется под учетной записью, обладающей правами локального администратора на компьютерах в учебном классе. Упражнения выполняются на диске с файловой системой NTFS.

1. В указанном преподавателем каталоге («папке») на локальном диске или диске виртуальной машины создайте новый каталог для выполнения данной лабораторной работы. Там создайте текстовый файл с произвольным содержанием. Просмотрите его разрешения на вкладке «Безопасность» (*англ.* Security) в свойствах файла (щелчок правой клавишей мыши на файле, в выпадающем меню — «Свой-

ства», *англ.* Properties). Вы увидите картинку, аналогичную представленной на рис. 5.1.

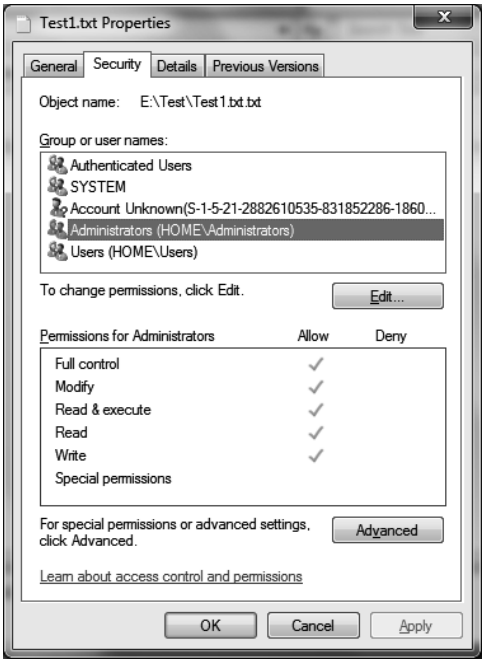


Рис. 5.1. Закладка «Безопасность» в свойствах файла

Обратите внимание, что сначала в списке могут появляться номера, потом они заменяются именами пользователей и групп. Эти номера и есть идентификаторы SID, которые затем, если это возможно, разрешаются в имена. На рис. 5.1 в ACL находится идентификатор удаленной учетной записи, который не может быть разрешен в имя пользователя (*англ.* Account Unknown — неизвестная учетная запись).

Проанализируйте текущие разрешения на созданный вами файл, имеющиеся у различных групп и пользователей. Эти разрешения унаследованы вашим файлом от объекта более высокого уровня, то есть от каталога, в котором он находится. Убедитесь в этом.

Для редактирования ACL нажмите кнопку Edit. Попробуйте выполнить следующие действия:

- добавить в ACL новую учетную запись, дав ей разрешения на изменение файла (*англ.* Modify);
- убрать группу `Users` из списка доступа на свой файл.

В связи с тем, что используется наследуемый ACL, вторую часть задания выполнить сразу не удастся: сначала нужно создать для объекта собственный ACL. Для этого нажмите кнопку Advanced (рис. 5.1) и в появившемся окне Advanced Security Settings (Дополнительные параметры безопасности) — кнопку Change Permissions (Изменить разрешения). Отказаться от использования наследуемых разрешений можно, убрав отметку Include inheritable permissions from this object's parent (переносить наследуемые от родительского объекта разрешения на этот объект), см. рис. 5.2. При этом будет предложено добавить в собственный ACL объекта разрешения из родительского списка (*англ.* Add) или полностью убрать их (*англ.* Remove). Лучше выбрать первый вариант, а потом уже отредактировать список так, как нужно.

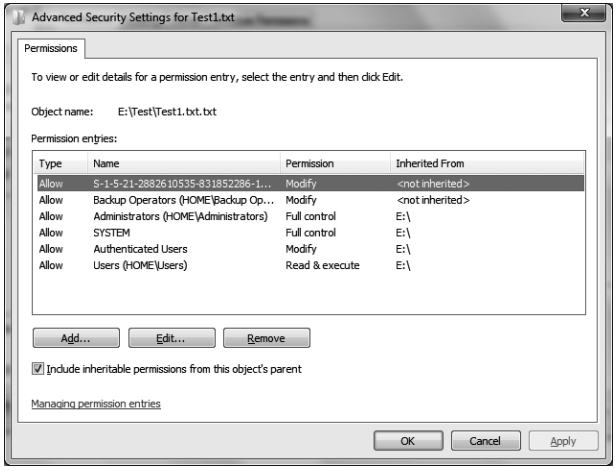


Рис. 5.2. Окно редактирования разрешений

Убрав группу Users (Пользователи) из ACL вашего файла, попробуйте открыть его от имени другой учетной записи, не имеющей явных разрешений и не входящей ни в одну из оставшихся в ACL групп (при необходимости создайте такую учетную запись). Если все сделано правильно, зайдя под подобной учетной записью Вы не получите доступа к файлу.

По умолчанию, локальная группа Users (Пользователи) включает всех локальных и доменных пользователей (если компьютер входит в домен Windows). Ознакомьтесь по справке или указанной ниже статье TechNet с другими стандартными локальными группами.

http://technet.microsoft.com/ru-ru/library/cc771990.aspx.

2. По справке или статье http://technet.microsoft.com/ru-ru/library/cc732880.aspx ознакомьтесь с разрешениями, которые можно давать на папку или файл в Windows. Подготовьте ответы на приведенные ниже вопросы.

Как можно просмотреть набор элементарных разрешений (из которых составлено, например, разрешение «изменить»), имеющихся у пользователя на файл или папку?

Чем составное разрешение «Изменить» (Modify) отличается от разрешения «Полный доступ» (Full Control)?

Дает ли разрешение на запись в папку право просматривать ее содержимое?

3. Эффективные или действующие разрешения (Effective Permissions) — это итоговые разрешения, складывающиеся из разрешений, данных пользователю и группам, которые и определяют, можно или нельзя данному субъекту получить доступ к данному объекту. Для выбранной учетной записи их можно узнать на вкладке Effective Permissions окна Advanced Security Settings, которое мы уже использовали при выполнении задания 1. Посмотрите действующие разрешения на ваш файл для одной из существующих учетных записей.

Чтобы лучше разобраться правилами назначения разрешений, выполните приведенные ниже задания.

При необходимости создайте локальную группу (назовем ее Students) и включенную в нее учетную запись (Student). Можно воспользоваться уже существующей записью. Группе Students дайте разрешение на чтение созданного вами файла. Проверьте, чтобы пользователь Student отсутствовал в перечне разрешений на файл. Зайдите под этой учетной записью. Может ли данный пользователь прочитать файл?

Повторите эксперимент, дополнительно указав на вкладке «Безопасность», что пользователю Student запрещено чтение файла. Сможет ли пользователь прочитать файл или нет? Ведь, с одной стороны, он в группе, которой разрешено чтение, с другой стороны — ему самому чтение запрещено.

4. Посмотрите разрешения на папку Program Files. Опишите в отчете, каким группам какие разрешения даются на эту папку, куда по умолчанию устанавливаются программы.

5. У объектов файловой системы, таких как файлы, папки и тома, есть владельцы. По умолчанию владельцем объекта становится его создатель. Владелец всегда может изменять разрешения для объекта, даже при отсутствии доступа к нему.

Более подробную информацию по этому поводу можно получить из справки или приведенной ниже статьи TechNet.

http://technet.microsoft.com/ru-ru/library/cc732983.aspx.

Ознакомьтесь с этими материалами. Подготовьте ответы на вопросы.

Кто может стать новым владельцем объекта (какие разрешения для этого нужны)?

Как сменить владельца объекта?

Предложите и выполните эксперимент, иллюстрирующий возможности владельца файла, отличающие его от других пользователей.

6. Управлять разрешениями на файлы и папки можно не только из графического интерфейса Windows, но и из командной строки. Для этого рекомендуется использовать утилиту icacls.exe (также для сов-

местимости поддерживается утилита cacls.exe). С возможностями этой утилиты можно ознакомиться по справке или из статьи TechNet.

http://technet.microsoft.com/ru-ru/library/cc753525.aspx.

С помощью утилиты выполните перечисленные ниже действия.

- сохраните текущий ACL выбранного вами файла в текстовый файл;
- предоставьте пользователю Student разрешение на изменение данного файла (modify);
- проверьте работу сделанных настроек;
- восстановите исходный ACL из копии, сделанной в начале выполнения задания.

7. При копировании и перемещении файлов на NTFS надо учитывать следующее:

- ACL файла или папки (и соответственно, все разрешения) сохраняется при перемещении объекта в пределах одного тома NTFS;
- при прочих операциях (перемещение между томами или копирование в любых вариантах) ACL заменяется на унаследованный от нового родительского контейнера.

Предложите и реализуйте эксперимент, позволяющий проверить выполнение этих правил.

5.2. УПРАВЛЕНИЕ ДОСТУПОМ В СУБД SQL SERVER

Цель работы.

Приобретение практических навыков настройки разрешений на доступ к объектам баз данных в среде СУБД Microsoft SQL Server 2008/2012.

Используемые программные средства.

Компьютер или виртуальная машина с установленной ОС семейства Windows и СУБД SQL Server. При работе в сети возможно совместное использование одного экземпляра SQL Server. В этом случае на остальных компьютерах должны быть установлены только клиентские компоненты и SQL Server Management Studio. Для выпол-

нения работы необходимы права, позволяющие создавать новую базу данных на экземпляре SQL Server.

Теоретические сведения.

Управление разрешениями на объекты реляционной базы данных несколько отличается от аналогичных операций в отношении объектов файловой системы. В данной лабораторной работе на примере СУБД SQL Server эти отличия будут показаны.

При работе с разрешениями в SQL Server используется понятие участников (principals), которые могут запрашивать ресурсы SQL Server, и которым могут предоставляться разрешения на использование таких ресурсов. Выделяются следующие группы участников:

- участники уровня Windows, к которым относятся локальные и доменные учетные записи пользователей и группы;

- участники уровня SQL Server, к которым относятся учетные записи SQL Server и роли уровня сервера;

- участники уровня базы данных — пользователи базы данных и роли уровня базы данных и приложения.

Необходимо отметить, что SQL Server разделяет понятие учетной записи (login) и пользователя (user). Сервер может быть сконфигурирован на использование только аутентификации Windows (*англ.* Windows Authentication Mode, используется по умолчанию) или на использование смешанного режима аутентификации (*англ.* SQL Server and Windows Authentication mode), см. рис. 5.3. В первом случае login можно создать только для пользователя или группы Windows. Во втором случае также возможно использовать собственные учетные записи SQL Server — логин и пароль хранятся самой СУБД, и ее же средствами выполняется проверка подлинности. При использовании смешанной аутентификации становится доступной административная учетная запись `sa`, которую рекомендуется переименовать и назначить ей надежный пароль. Учетная запись авторизуется на выполнение одной из серверных ролей (табл. 5.1).

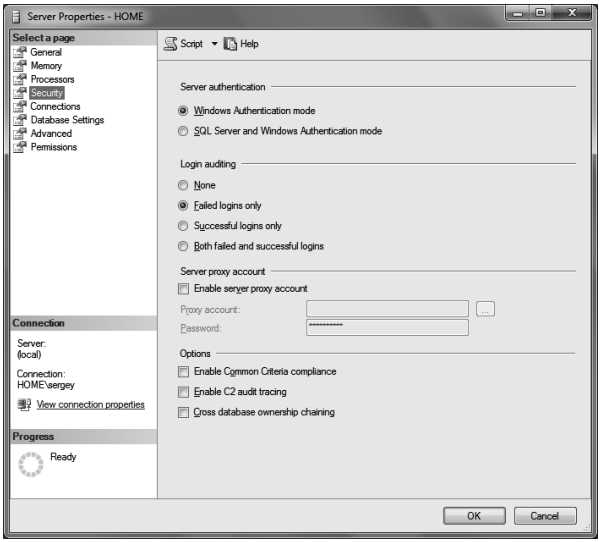


Рис. 5.3. Окно редактирования настроек безопасности экземпляра SQL Server

Таблица 5.1

Роли уровня сервера

Роль	Описание возможностей
sysadmin	Разрешено выполнять любые действия на сервере.
dbcreator	Разрешено создавать базы данных.
bulkadmin	Могут выполнять инструкцию BULK INSERT.
diskadmin	Позволяет управлять файлами на диске.
processadmin	Позволяет управлять подключениями, запускать и приостанавливать экземпляр SQL Server.
securityadmin	Создание и управление учетными записями, право «сбросить» пароль учетной записи. Управление разрешениями на уровне сервера и на уровне базы данных (при наличии доступа к базе данных).
serveradmin	Включает возможности ролей diskadmin и processadmin, позволяет изменять параметры конфигурации на уровне сервера и выключать сервер.
setupadmin	Добавление и удаление связанных серверов.
public	Каждая учетная запись принадлежит этой роли, членство в роли public изменить нельзя.

На уровне базы данных учетной записи сопоставляется пользователь (user). Для одной и той же учетной записи в различных базах данных сервера могут создаваться пользователи с разными именами.

Пользователи получают разрешения на работу с объектами базы данных или напрямую, или путем авторизации пользователя на выполнение одной из ролей уровня базы данных (рис. 5.4). Последний способ является более предпочтительным и позволяет организовать управление доступом в соответствии с ролевой моделью, описанной в первой главе пособия.

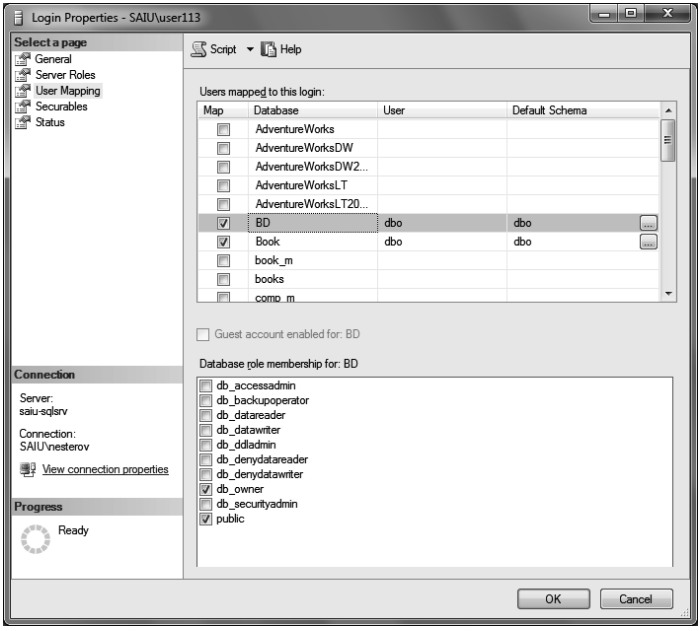


Рис. 5.4. Учетные записи и соответствующие им пользователи и роли в базах данных

Список ролей уровня сервера предопределен, и новые создавать нельзя (табл. 5.1). Также есть предопределенный набор ролей уровня базы данных (табл. 5.2), но в этом случае имеется возможность создавать новые роли.

Таблица 5.2

Роли уровня базы данных

Роль	Описание возможностей
db_owner	Владелец базы данных, можно выполнять все действия по настройке и обслуживанию базы данных, а также удалять базу данных.
db_securityadmin	Управление составом ролей (кроме роли db_owner) и связанными с ними разрешениями.
db_accessadmin	Добавление и удаление пользователей базы данных.
db_backupoperator	Возможность создавать резервные копии базы данных.
db_ddladmin	Выполнение DDL-инструкций (создание, изменение, удаление объектов базы данных, таких как таблицы, представления и т. д.).
db_datareader	Чтение данных (SELECT) из всех пользовательских таблиц, представлений и функций.
db_denydatareader	Запрет на чтение данных (SELECT) из всех пользовательских таблиц, представлений и функций.
db_datawriter	Право добавлять, удалять или изменять данные во всех пользовательских таблицах.
db_denydatawriter	Запрет добавлять, изменять или удалять данные в пользовательских таблицах.
public	Роль по умолчанию, имеющаяся в каждой базе данных. Каждый пользователь БД авторизован на эту роль.

Более подробную информацию об организации системы безопасности СУБД SQL Server можно получить из справочной системы TechNet: http://technet.microsoft.com/ru-ru/library/bb510589.aspx.

План работы.

1. Используя указанную преподавателем доменную или локальную учетную запись Windows, с помощью SQL Server Management Studio подключитесь к используемому экземпляру SQL Server. Проверьте установленный на сервере режим аутентификации.

2. В окне Object Explorer (по умолчанию — левая часть окна Management Studio) откройте список учетных записей (logins). На вы-

полнение каких серверных ролей авторизована используемая вами учетная запись?

3. В каких базах данных сервера вашей учетной записи сопоставлены пользователи? На выполнение каких ролей они авторизованы?

4. В среде Management Studio создайте новую базу данных. Откройте список пользователей и ролей. Убедитесь, что учетная запись, под которой вы работаете, сопоставлена пользователю dbo, авторизованному на роль db_owner.

5. Используя приведенный ниже скрипт, создайте в базе данных таблицы. Перед тем как запустить скрипт, уберите символы комментария («--») из первой строки и после ключевого слова use укажите имя вашей базы данных.

```
--use MyTest1
GO
CREATE TABLE dbo.Book (
    book_id int IDENTITY (1, 1) primary key,
    Title varchar(50) NOT NULL, --название книги
    Author varchar(50), -- автор
    Publisher varchar(50), -- издательство
    [Year] smallint) — год издания
GO
CREATE TABLE dbo.Status (
Status_id int IDENTITY (1, 1) primary key,
Status_name varchar(50) NOT NULL
) -- статус: выдана, в библиотеке и т.д.
GO
CREATE SCHEMA libr
GO
CREATE TABLE libr.Book_in_lib (
    lib_id int primary key , --номер экземпляра
    book_id int references dbo.Book ,
    status_id int references dbo.[Status])
GO
```

Обратите внимание, что приведенный скрипт создает не только три таблицы, но и схему libr. В SQL Server схема является контейнером логического уровня, к которому относятся объекты базы данных. Во вновь созданной БД уже будет несколько схем: dbo, sys, information_schema и т. д. Схема dbo — это схема по умолчанию для новых пользовательских объектов, sys и information_schema используются системными объектами. Оператором CREATE SCHEMA в БД можно создавать новые схемы.

Защищаемым объектом, на действия с которым пользователю предоставляются разрешения, может быть база данных, схема или объект базы данных. Определенное для схемы разрешение неявным образом распространяется на все объекты схемы, разрешение для базы данных — на все схемы и объекты этих схем.

6. Для указанной преподавателем учетной записи SQL Server (при самостоятельном выполнении работы создайте учётную запись Windows и учётную запись SQL Server для нее) создайте пользователя в вашей базе данных, в качестве схемы по умолчанию выберите dbo. В Management Studio это можно сделать из графического интерфейса (контекстное меню узла Security для выбранной БД, там New…-> User) или выполнив оператор CREATE USER. Например (если схема не указана, подразумевается dbo):

```
USE MyTest1
go
CREATE USER ns FOR LOGIN [HOME\ns]
```

Добавьте этого пользователя в роль db_datareader. Это можно сделать или через графический интерфейс или с помощью системной хранимой процедуры sp_addrolemember, первым параметром которой будет имя роли, а вторым — имя пользователя.

```
EXEC sp_addrolemember 'db_datareader', 'ns'
```

Введите в таблицы тестовый набор данных.

Подключитесь к серверу с учетной записью другого пользователя. Убедитесь, что можно получить доступ к базе данных и читать записи из всех таблиц, а добавлять или изменять данные нельзя.

7. Создадим новую роль уровня базы данных и добавим ей разрешение на удаление (DELETE), изменение (UPDATE) и добавление данных (INSERT) в объектах схемы `libr`. Добавим нашего пользователя к этой роли. Указанные действия надо выполнять с правами администратора или владельца базы данных. Как и в предыдущем случае, все это можно сделать в графическом интерфейсе или запуском скрипта.

```
CREATE ROLE libr_writer
GO
GRANT INSERT, UPDATE, DELETE ON SCHEMA :: libr TO
libr_writer
Go
EXEC sp_addrolemember 'libr_writer', 'ns'
```

Используемый в приведенном скрипте оператор GRANT позволяет предоставить разрешения. Оператор DENY позволяет запретить выполнение каких-то действий, а оператор REVOKE отменяет установленные оператором GRANT или DENY настройки разрешений. Таким образом, у разрешения может быть три состояния: «разрешено», «запрещено», «не задано». Действие можно выполнить, только если оно разрешено непосредственно пользователю или одной из ролей, на которые он авторизован. Запрещение более приоритетно, чем разрешение: если пользователь авторизован на выполнение двух ролей, одной из них действие разрешено, а другой — запрещено, то пользователь это действие выполнить не сможет. В SQL Server Management Studio можно просмотреть эффективные разрешения для пользователя (рис. 5.5).

Конкретный набор возможных разрешений зависит от типа объекта, с полным списком разрешений рекомендуется ознакомиться по

справке или приведенной ниже статье TechNet:
http://technet.microsoft.com/ru-ru/library/ms191291.aspx.

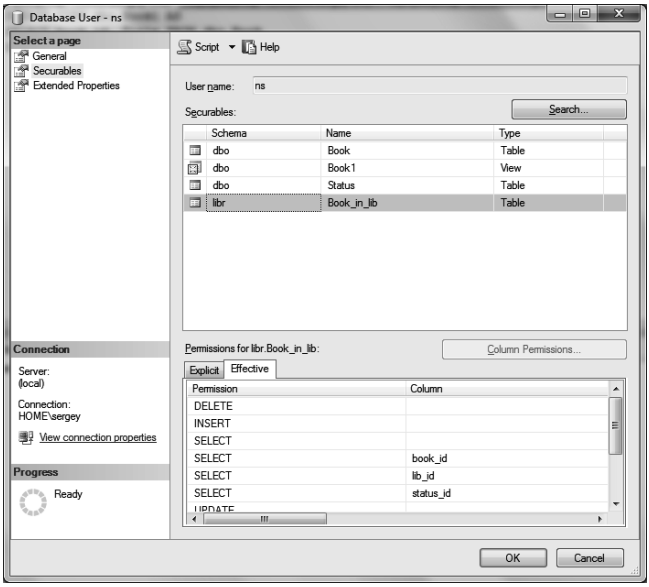


Рис. 5.5. Эффективные разрешения пользователя

Выполните описанные действия. Убедитесь, что пользователь с ограниченными правами может изменять данные в таблице `Book_in_lib`, относящейся к схеме `libr`.

8. Иногда нужно предоставить пользователю права на изменение отдельных столбцов. Как отмечается в документации SQL Server, на столбец могут быть предоставлены только разрешения SELECT, REFERENCES и UPDATE. Например:

```
GRANT UPDATE ON dbo.Book(Title) TO libr_writer
```

Выполните аналогичные действия в своей базе данных, проверьте, что пользователь получил указанные разрешения.

9. Самостоятельно по справке ознакомьтесь с форматом оператора CREATE VIEW, особое внимание обратите на задаваемые до-

полнительные параметры. Создайте представление, выбирающее из таблицы Book книги, изданные не ранее 2000 года. Предоставьте пользователю с ограниченными правами возможность изменять и добавлять подобные книги. Возможности изменять прочие записи таблицы и добавлять книги, изданные до 2000 года, он иметь не должен.

5.3. ВЫЯВЛЕНИЕ УЯЗВИМОСТЕЙ С ПОМОЩЬЮ MICROSOFT BASELINE SECURITY ANALYZER

Цель работы.

Приобретение практических навыков выявления уязвимостей в ПО производства компании Microsoft с помощью специализированного программного средства Baseline Security analyzer (BSA).

Используемые программные средства.

Компьютер и/или виртуальная машина с ОС Microsoft и установленной программой BSA. Для выполнения работы требуются права локального администратора.

Теоретические сведения.

Microsoft Baseline Security analyzer — программа, позволяющая проверить уровень безопасности установленной конфигурации операционной системы Windows, начиная с версии Windows 2000. Также проверяется и ряд других приложений разработки Microsoft. Данное средство можно отнести к разряду систем анализа защищенности. Оно безвозмездно распространяется через web-сервера Microsoft.

В процессе работы BSA проверяет наличие обновлений безопасности операционной системы, офисного пакета Microsoft Office (для версий XP и более поздних), серверных приложений, таких как SQL Server, Exchange Server, Internet Information Server и т. д. Кроме того, проверяется ряд настроек, касающихся безопасности, например, действующая политика паролей. Более подробную информацию можно получить из статей TechNet, посвященных продукту: http://technet.microsoft.com/ru-ru/security/cc184923.

Описание работы.

Перейдем к знакомству с программным продуктом. Надо отметить, что при подготовке описания данной лабораторной работы использовалась версия BSA 2.1, при использовании других версий пользовательский интерфейс может отличаться.

При запуске открывается окно, позволяющее выбрать объект проверки — один компьютер (выбирается по имени или ip-адресу), несколько (задаваемых диапазоном ip-адресов или доменным именем) или просмотреть ранее сделанные отчеты сканирования системы. При выборе сканирования отдельного компьютера по умолчанию подставляется имя локальной станции, но можно указать имя или ip-адрес другого компьютера.

Рис. 5.6. Настройка параметров проверки

Можно задать перечень проверяемых параметров. На рис. 5.6 представлен выбор вариантов проверки:

- проверка на наличие уязвимостей Windows, вызванных некорректным администрированием;

- проверка на «слабые» пароли (пустые пароли, отсутствие ограничений на срок действия паролей и т. д.);
- проверка на наличие уязвимостей web-сервера IIS, вызванных некорректным администрированием;
- аналогичная проверка в отношении СУБД MS SQL Server;
- проверка на наличие обновлений безопасности.

Перед началом работы программа обращается на сервер Microsoft для получения перечня обновлений для ОС и описаний известных уязвимостей.

Для успешной проверки локальной системы необходимо, чтобы программа выполнялась от имени учетной записи с правами локального администратора. Иначе проверка не может быть проведена, о чем будет выдано соответствующее сообщение (*на англ.* You do not have sufficient permissions to perform this command. Make sure that you are running as the local administrator or have opened the command prompt using the 'Run as administrator' option).

По результатам сканирования формируется отчет, вначале которого дается общая оценка уровня безопасности конфигурации проверяемого компьютера. В приведенном на рис. 5.7 примере уровень риска оценивается как «высокий» (*англ.* Severe risk).

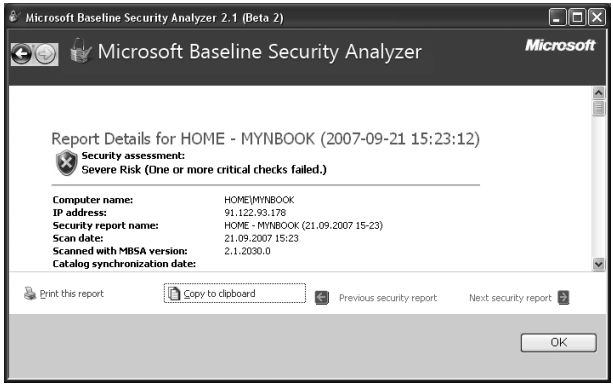


Рис. 5.7. Заголовок отчета с указанием уровня риска

Далее приводится перечень обнаруженных уязвимостей, разбитый на группы: результаты проверки установки обновлений, результаты проверки Windows и т. д. Надо отметить, что выпускаемые Microsoft обновления бывают различных типов:

- *Security updates* — собственно обновления безопасности, как правило, посвященные исправлению одной уязвимости программного продукта;
- *Update rollups* — набор исправлений безопасности, который позволяет одновременно исправить несколько уязвимостей. Это упрощает обслуживание процесса обновления программного обеспечения;
- *Service packs* — набор исправлений, как связанных, так и несвязанных с безопасностью. Установка Service pack, как правило, исправляет все уязвимости, обнаруженные с момента выхода предыдущего Service pack, таким образом устанавливать промежуточные обновления уже не надо.

В окне с описанием результата проверки можно выбрать ссылку Result details и получить более подробную информацию о найденных проблемах (рис. 5.8, 5.9).

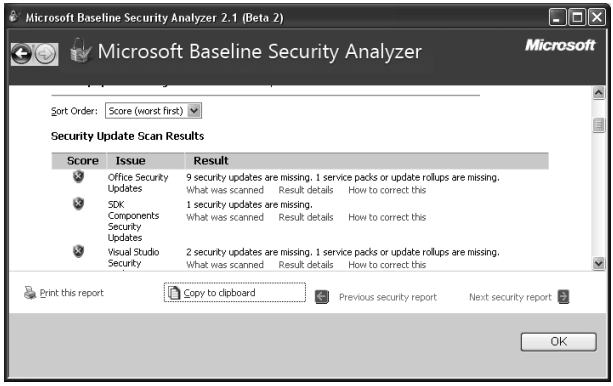


Рис. 5.8. Перечень отсутствующих обновлений (по группам)

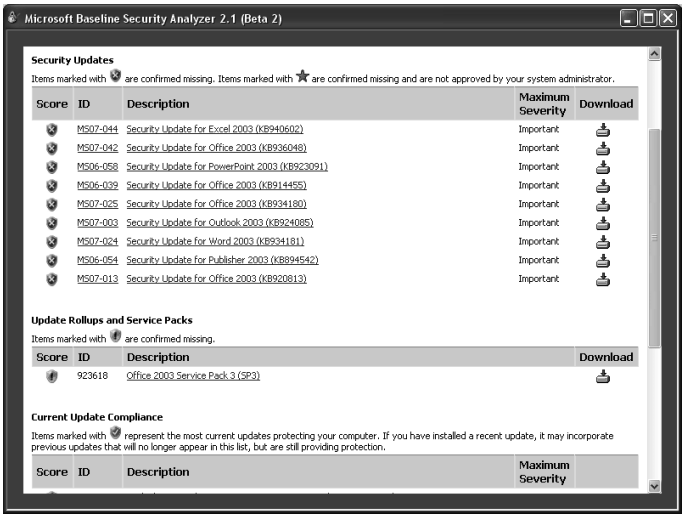


Рис. 5.9. Подробное описание отсутствующих обновлений Microsoft Office

При наличии подключения к сети Интернет, перейдя по приводимой в отчете ссылке, можно получить информацию об отсутствующем обновлении безопасности и скачать его.

Нужно отметить, что установка обновлений для систем с высокими требованиями в области непрерывности работы требует предварительной тщательной проверки совместимости обновлений с использующимися приложениями. Подобная проверка обычно производится на тестовых системах с близкой конфигурацией ПО. В то же время для небольших организаций и пользователей домашних компьютеров такая проверка зачастую неосуществима. Поэтому надо быть готовым к тому, чтобы восстановить систему после неудачного обновления. Для ОС семейства Windows, если после установки обновлений загрузка в нормальном режиме стала невозможна, можно использовать специальные режимы загрузки — безопасный режим или режим загрузки последней удачной конфигурации.

Также надо отметить еще одну особенность. На данный момент Baseline Security analyzer не существует в локализованной русско-

язычной версии. И содержащиеся там ссылки на пакеты обновлений могут указывать на иные языковые версии, что может создать проблемы при обновлении локализованных продуктов.

Для удобства работы с отчетом его материалы можно распечатать или скопировать в буфер обмена и через него поместить отчет, например, в документ Word.

Кроме версии программы с графическим интерфейсом, существует также утилита с интерфейсом командной строки. Называется она `mbsacli.exe` и находится в том же каталоге, куда устанавливался Baseline Security analyzer, например, «C:\Program Files\Microsoft Baseline Security Analyzer 2». У утилиты есть достаточно много параметров, получить информацию о них можно при запуске с ключом "`/?`".

Запуск без ключей приведет к сканированию локального компьютера с выводом результатов на консоль. Чтобы сохранить результаты сканирования, можно перенаправить вывод в какой-либо файл. Например: `mbsacli > mylog.txt`.

Ключ `/xmlout` приводит к запуску утилиты в режиме проверки обновлений: проверка на уязвимости, явившиеся результатом неудачного администрирования, проводиться не будет. Отчет формируется в формате xml. Например:

```
mbsacli /xmlout > c:\myxmlog.xml.
```

Задания.

Выполните проверку Вашего компьютера с помощью Microsoft Baseline Security analyzer. В отчете о выполнении лабораторной работы укажите:

- как был оценен уровень уязвимости компьютера;
- какие проверки проводились, в какой области обнаружено наибольшее количество уязвимостей;
- опишите наиболее серьезные уязвимости каждого типа, выявленные на компьютере.

Проведите анализ результатов: какие уязвимости можно устранить, какие — нельзя из-за особенностей конфигурации ПО или использования компьютера.

Выполните удаленную проверку соседнего компьютера из сети лаборатории (или другой виртуальной машины). Опишите наиболее серьезные уязвимости.

Выполните проверку нескольких компьютеров с помощью утилиты `mbsacli`. Для этого, предварительно создайте текстовый файл с перечнем имен компьютеров или ip-адресов и запускайте `mbsacli` с ключом `/listfile`, после которого указывается имя файла с перечнем компьютеров. В результате должно быть получено сообщение примерно следующего содержания:

```
Computer Name, IP Address, Assessment, Report Name
-----------------------------------------------
HOME\MYNBOOK, 127.0.0.1, Severe Risk, HOME - MYNBOOK
(06.10.2013 13-51)
```

Для того чтобы увидеть подробные результаты проверки, надо повторно запустить `mbsacli` с ключом `/ld`, после которого указывается имя отчета. Вывод можно перенаправить в текстовый файл для дальнейшей обработки. Например:

```
mbsacli /ld "HOME - MYNBOOK (06.10.2013 13-51)" >
c:\test\report1.txt
```

После выполнения задания проанализируйте результаты, кратко опишите их в отчете по лабораторной работе.

5.4. ИСПОЛЬЗОВАНИЕ СКАНЕРОВ БЕЗОПАСНОСТИ ДЛЯ ПОЛУЧЕНИЯ ИНФОРМАЦИИ О ХОСТАХ В СЕТИ

Цель работы.

Приобретение практических навыков определения работающих сетевых приложений с помощью сетевого сканера безопасности.

Используемые программные средства.

Компьютерная сеть учебного класса и виртуальные машины с ОС Windows Server. Сетевой сканер nmap.

Теоретические сведения.

Сетевые сканеры безопасности — программные средства, позволяющие проверить уровень уязвимости сетей. Они имитируют различные обращения к сетевым узлам, выявляя операционные системы компьютеров, запущенные сервисы, имеющиеся уязвимости. В некоторых случаях сканеры безопасности также имитируют реализацию различных атак, чтобы проверить уровень подверженности им компьютеров защищаемой сети.

При проведении анализа рисков сканеры безопасности могут использоваться как в процессе инвентаризации ресурсов, так и для оценки уровня уязвимости узлов сети.

В данной лабораторной работе используется ПО NMAP, которое свободно доступно на сайте http://nmap.org/.

Сканирование можно проводить при помощи утилиты командной строки nmap или с помощью графической оболочки для нее — Zenmap GUI.

По руководству пользователя, доступному по ссылке http://nmap.org/man/ru/, изучите вопросы, связанные с определением цели сканирования, способами сканирования портов, определения версий операционной системы и используемых служб. Кратко опишите это в отчете.

План работы.

Проведите исследование для виртуальных машин с Windows Server, работающих на вашем и соседних компьютерах. Сначала выполните сканирование при включенном межсетевом экране виртуальной машины, потом — при отключенном (настройка делается в Панель управления ->Система и безопасность -> Брандмауэр Windows).

Опишите и проанализируйте полученные результаты. Какие сетевые службы запущены на исследуемых виртуальных машинах? При описании можно использовать материалы технической статьи «Службы и сетевые порты в Microsoft Windows» доступной по ссылке http://support.microsoft.com/?kbid=832017.

Установите виртуальную машину роль «Web-сервер» и проверьте запуск web-сервера IIS с сайтом по умолчанию.

Повторно проведите сканирование данной виртуальной машины. Что показал сканер?

5.5. ВСТРОЕННЫЙ МЕЖСЕТЕВОЙ ЭКРАН (FIREWALL) WINDOWS SERVER 2008

Цель работы.

Приобретение практических навыков настройки межсетевого экрана.

Используемые программные средства.

Компьютерная сеть учебного класса и виртуальные машины с ОС Windows Server.

Теоретические сведения.

Персональный межсетевой экран появился в операционных системах семейства Windows, начиная с Windows XP / Windows Server 2003. В Windows Server 2008 возможности этого компонента существенно расширены, что позволяет более гибко производить настройки.

Описание работы.

Текущие настройки можно посмотреть, запустив из Панели управления (Control Panel) Windows Firewall и выбрав в открывшемся окне ссылку Change Settings.

Появившееся окно управления параметрами межсетевого экрана содержит 3 вкладки (рис. 5.10, *а*, *б*, *в*). Первая из них позволяет включить или отключить межсетевой экран. Во включенном состоянии он может разрешать определенные входящие подключения или запрещать все входящие подключения (флажок Block all incoming connections).

Исключения из общего «блокирующего» правила определяются на вкладке Exceptions. Там есть ряд предопределенных правил, также пользователь может добавлять свои. Если нужно, чтобы какое-то приложение при включенном межсетевом экране обслуживало вхо-

дящие подключения, для него должно быть описано правило. Сделать это можно либо указав программу (кнопка Add program), либо описав разрешаемый порт и протокол (кнопка Add Port). Пример формирования подобного правила представлен на рис. 5.10, *г*. Там дается разрешение для подключения на TCP-порт 8080. Если надо ограничить перечень ip-адресов, с которых производится подключение, это можно сделать, нажав кнопку Change Scope (по умолчанию, разрешены подключения с любого адреса).

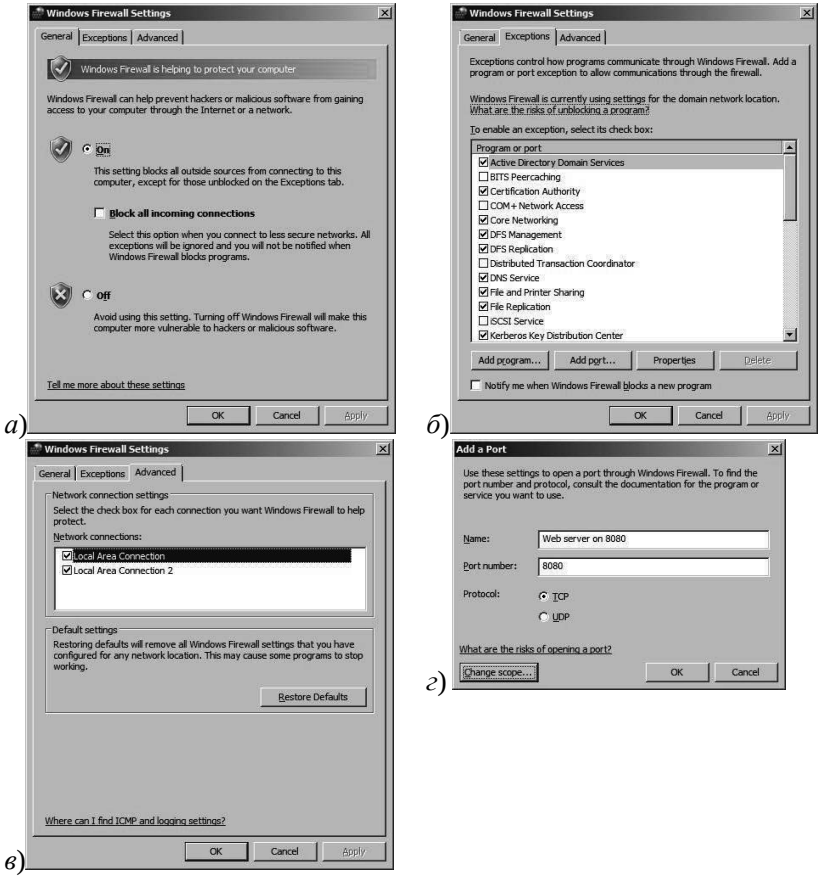


Рис. 5.10. Окно управления параметрами межсетевого экрана

Установка флажка «Notify me when Windows Firewall blocks a new program» приводит к тому, что при попытке нового приложения принимать входящие подключения, пользователю будет выдано сообщение. Если пользователь разрешит такой программе работать, для нее будет сформировано разрешающее правило.

Вкладка Advanced (рис. 5.10, *в*) позволяет включить или отключить межсетевой экран для отдельных сетевых интерфейсов.

Задание 1.

1. Откройте окно управления межсетевым экраном.
2. Опишите действующие настройки.
3. Создайте новое разрешающее правило.

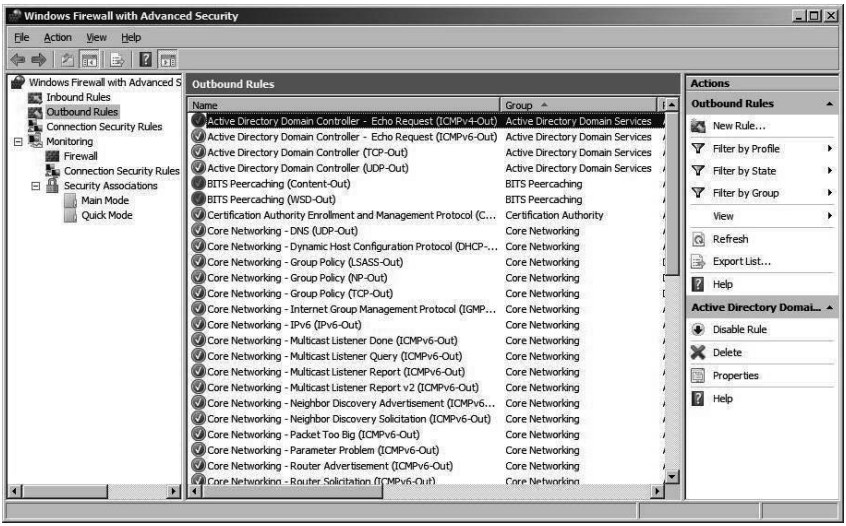


Рис. 5.11. Окно оснастки Windows Firewall with Advanced Security

Пока что работа с межсетевым экраном практически не отличалась от того, что было в Windows Server 2003. Новые возможности мы увидим, если из меню Administrative Tools запустить оснастку Windows Firewall with Advanced Security (рис. 5.11). В окне оснастки

можно увидеть настройки для разных профилей и выполнить более тонкую настройку правил фильтрации.

Правила фильтрации разделены на две группы — входящие правила и исходящие правила. В приведенном на рис. 5.11 примере настройки выполняются на контроллере домена. И для контроллеров определено правило, разрешающее отправку ICMP-пакетов echo request (они, в частности, отправляются, если надо проверить доступность удаленного узла с помощью команды ping).

Задание 2.

1. Найдите правило, разрешающее отсылку ICMP-пакетов echo request. Проверьте его работу для какого-нибудь узла из локальной или внешней сети, используя его ip-адрес (например, командой `ping 192.168.1.10` можно проверить доступность компьютера с указным адресом). Если ответ пришел, можно переходить ко второй части задания. Если ответа нет, попробуйте найти такой узел, который пришлет ответ.

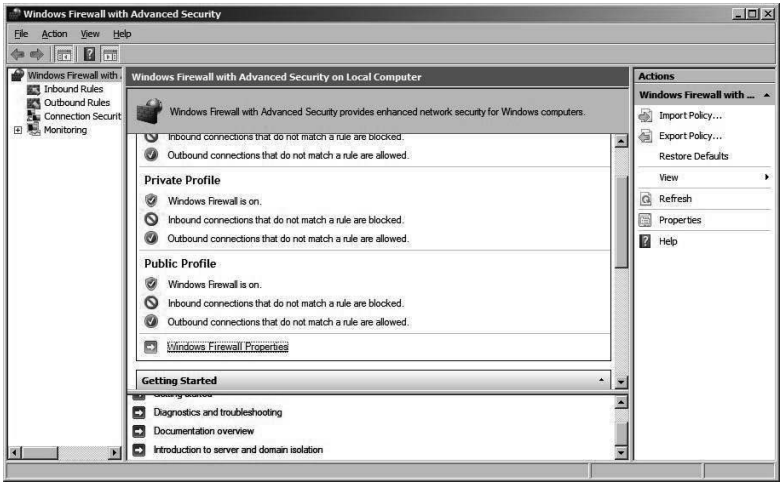


Рис. 5.12. Главное окно оснастки Windows Firewall with Advanced Security

2. Выбрав кнопку New Rule, создайте правило, запрещающее отправку ICMP-пакетов на данный узел. Проверьте его работу.

Теперь рассмотрим настройку, связанную с ведением журналов межсетевого экрана. По умолчанию журналирование отключено. Но если возникает подозрение, что межсетевой экран мешает установлению какого-то типа сетевых соединений, можно включить эту опцию и проанализировать журнал.

На рис. 5.12 представлено главное окно оснастки. Выберем пункт Firewall Properties и активируем ведение журнала отброшенных пакетов (рис. 5.13). Для этого в группе Logging в окне рис. 5.13, *а* надо нажать кнопку Customize и выполнить настройку, представленную на рис. 5.13, *б*.

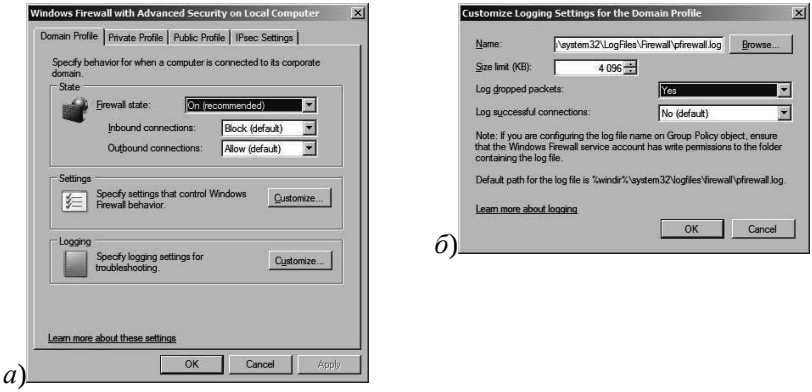


Рис. 5.13. Настройка параметров журнала событий

Задание 3.

1. Активируйте ведение журнала.

2. Выполните команду `ping` для проверки доступности узла, для которого создавалось блокирующее правило.

3. Проверьте содержимое файла журнала (путь к нему описан в окне, аналогичном представленному на рис. 5.13, *б*). Найдите записи, соответствующие сброшенным (*англ.* drop) ICMP-пакетам.

5.6. ИСПОЛЬЗОВАНИЕ ЦИФРОВЫХ СЕРТИФИКАТОВ

Цель работы.

Ознакомление с порядком использования цифровых сертификатов X.509 в протоколах защиты данных SSL/TLS и S/MIME.

Используемые программные средства.

Компьютер с подключением к сети Интернет, браузер, Microsoft Outlook или другой почтовый клиент, поддерживающий S/MIME.

Описание работы.

В ходе данной лабораторной работы будут рассмотрены некоторые вопросы использования цифровых сертификатов.

Начнем с их использования протоколом SSL/TLS (на самом деле это два разных протокола, но так как TLS разработан на базе SSL, принцип использования сертификатов один и тот же). Этот протокол широко применяется в сети Интернет для защиты данных передаваемых между Web-сервером и браузером клиента. Для аутентификации сервера в нем используется сертификат X.509.

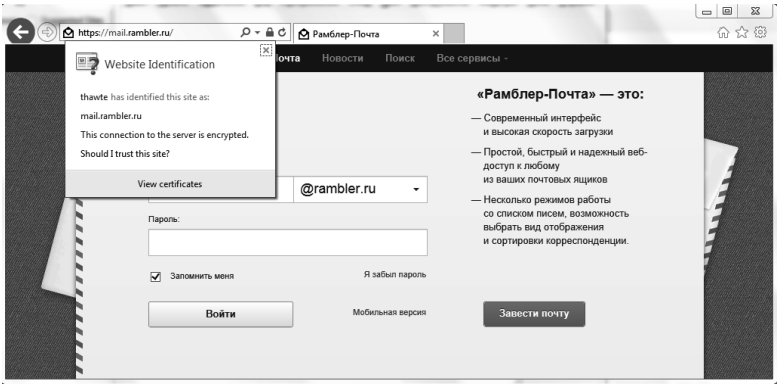


Рис. 5.14. Защищенное соединение с сайтом

Для примера обратимся на сайт Рамблер-почты (mail.rambler.ru). Для обеспечения безопасности передаваемых данных устанавливается защищенное соединение (рис. 5.14), на что указывает префикс https в строке адреса и изображение закрытого замка (в случае браузера In-

ternet Explorer). Если щелкнуть мышью по изображению замка, то увидим представленное на рис. 5.14 сообщение о том, что подлинность узла с помощью сертификата подтверждает удостоверяющий центр Thawte. Это значит, что мы на самом деле обратились на подлинный сайт Рамблер-почты (а не подделанный нарушителями) и можем безопасно вводить логин и пароль.

Нажав на ссылку «Просмотр сертификата» (*англ.* View certificates) можно узнать подробности о получателе и издателе, другие параметры сертификата (рис. 5.15).

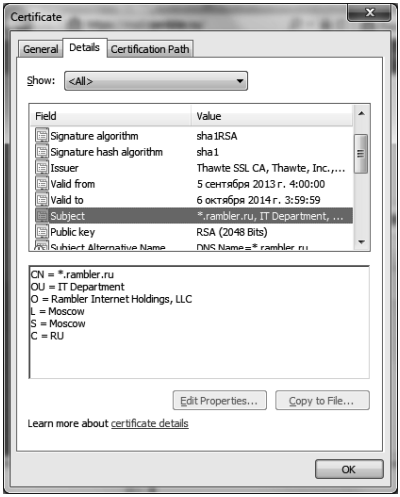


Рис. 5.15. Параметры сертификата

Задание.

Просмотрите параметры сертификата какого-либо защищенного сайта, например «Личного кабинета сотрудника СПбГПУ» https://staff.spbstu.ru или Сбербанк Онлайн https://online.sberbank.ru. Опишите, кем на какой срок, для какого субъекта сертификат был выдан.

Теперь рассмотрим другой вариант — мы подключаемся по SSL к Web-серверу, а браузер не может проверить его подлинность или

сообщает, что используется сертификат, выданный для другого Web-сервера. Например, подобная ситуация произойдет при подключении в раздел Интернет-обслуживания оператора мобильной связи Tele2 по ссылке https://www.selfcare.tele2.ru/work.html (на рис. 5.16).

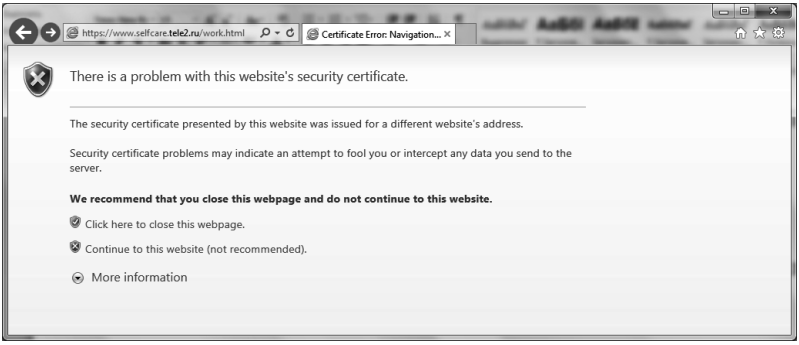


Рис. 5.16. Браузер сообщает о проблеме с сертификатом

Если нажать ссылку «Продолжить открытие этого Web-узла» (*англ.* Continue to this website), можно будет просмотреть сертификат.

Задание.

Разберитесь, в чем проблема с указанным сертификатом (на всякий случай в конце описания лабораторной работы приведен ответ).

Теперь рассмотрим, как хранятся сертификаты. Операционная система Windows поддерживает защищенные хранилища ключей и сертификатов. Работать с хранилищем можно, используя настройку консоль управления MMC «Сертификаты».

Из меню `Пуск-> Выполнить` запустите консоль командой `mmc`. В меню `Консоль` выберите `Добавить или удалить оснастку`, а в списке оснасток выберите `Сертификаты` (*англ.* Certificates). Если будет предложен выбор (а это произойдет, если вы работаете с правами администратора), выберите пункт «Моей учетной записи».

С помощью оснастки можно просматривать сертификаты текущего пользователя. Если ранее сертификаты не запрашивались, то в разделе «Личные сертификаты» элементов не будет.

В разделе «Доверенные корневые центры сертификации» (*англ.* Trusted Root Certification Authorities) представлен достаточно обширный список центров, чьи сертификаты поставляются вместе с операционной системой. Найдите в нем сертификаты удостоверяющего центра Thawte. Благодаря тому, что они уже были установлены, в рассмотренном в начале работы примере с подключением к почтовому серверу Рамблер браузер мог подтвердить подлинность узла.

Перейдем к раздел «Сертификаты, к которым нет доверия» (*англ.*Untrusted Certificates). Там находятся отозванные сертификаты, в частности, два сертификата, которые неизвестные получили от имени корпорации Microsoft в центре сертификации VeriSign в 2001 году. Когда это выяснилось, сертификаты отозвали (рис. 5.17).

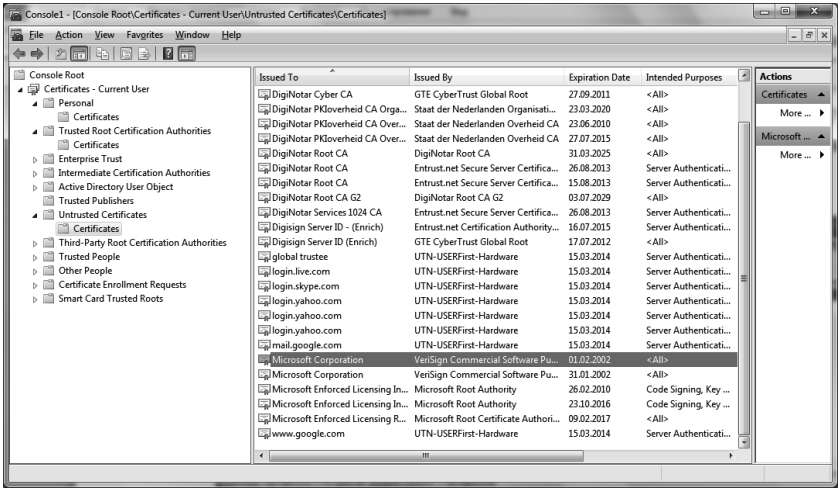


Рис. 5.17. Отозванные сертификаты

Теперь рассмотрим процесс запроса сертификата. Некоторые удостоверяющие центры позволяют получить «персональный» сертификат для защиты электронной почты с помощью протокола S/MIME (как разбиралось в разделе 3.1 данного пособия, для распре-

деления ключей в S/MIME нужен сертификат формата X.509). Например, их предоставляет удостоверяющий центр COMODO http://www.comodo.com/home/email-security/free-email-certificate.php.

Чтобы получить сертификат, понадобится заполнить небольшую анкету, указав имя, фамилию, адрес электронной почты (должен быть действующим), пароль для восстановления. Когда все заполнено, на указанный адрес почты будет отправлено письмо со ссылкой, по которой надо пройти для завершения процедуры получения сертификата. Полученный цифровой сертификат будет помещен в хранилище, в чем можно убедиться с помощью оснастки «Сертификаты».

Если использовать сертификат для защиты почты, дальнейшая настройка зависит от почтового клиента. Если это Microsoft Outlook 2010, требуемая настройка делается в меню `Файл-> Параметры -> Центр управления безопасностью -> Защита электронной почты`. Там можно выбрать используемый сертификат и алгоритмы шифрования (рис. 5.18).

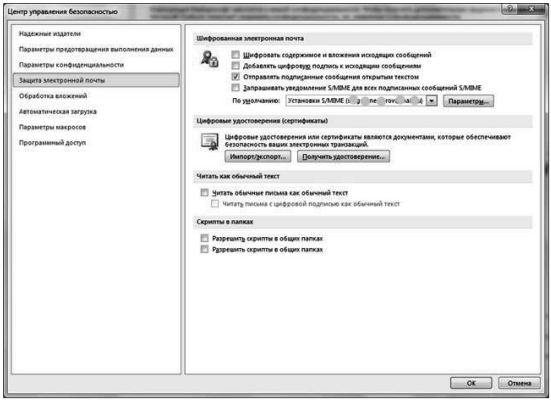


Рис. 5.18. Настройка S/MIME для Outlook

Чтобы использовать возможности S/MIME в Outlook 2010, в окне сообщения перейдите на вкладку «Параметры» и выберите «Подписать» и/или «Шифровать».

Настройка других почтовых клиентов выполняется во многом аналогично.

Задание.

Запросите цифровой сертификат и настройте почтовый клиент для использования S/MIME. Отправьте письмо с электронной подписью.

Ответ на задание про сертификат на сайте Tele2.

Проблема была в том, что сертификат был выдан для узла https://my.tele2.ru. Мы же воспользовались ссылкой с именем https://www.selfcare.tele2.ru и получили сертификат, формально не соответствующий адресу узла.

Доверять или нет такому сертификату — зависит от конкретного случая.

5.7. СОЗДАНИЕ ЦЕНТРА СЕРТИФИКАЦИИ (УДОСТОВЕРЯЮЩЕГО ЦЕНТРА) В WINDOWS SERVER 2008

Цель работы.

Приобретение практических навыков развертывания и настройки центра сертификации встроенными средствами Windows Server 2008.

Используемые программные средства.

Компьютер или виртуальная машина с ОС Windows Server 2008 и установленной и настроенной ролью Active Directory Domain Services (контроллер домена).

Описание работы.

Предыдущая лабораторная работа была посвящена вопросам использования цифровых сертификатов X.509 конечными пользователями. В данной лабораторной работе рассматриваются возможности, которые предоставляет Windows Server 2008 по созданию собственно центра сертификации (*англ.* Certification Authority, CA) в организации. Соответствующие службы присутствовали в серверных операционных системах семейства Windows, начиная с Windows 2000 Server.

В Windows Server 2008 для того, чтобы сервер смог работать как центр сертификации, требуется сначала добавить серверу роль Active Directory Certificate Services. Делается это с помощью оснастки Server Manager, которую можно запустить из раздела Administrative Tools в стартовом меню.

В Server Manager раскроем список ролей и выберем добавление роли (Add Roles), см. рис. 5.19.

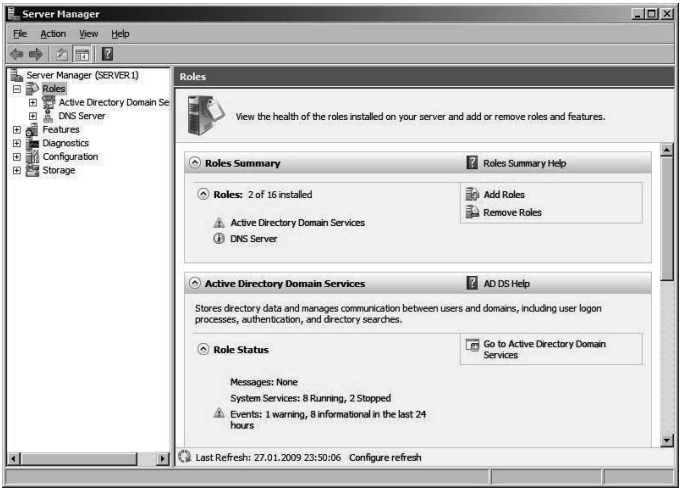


Рис. 5.19. Добавление роли

В нашем примере роль добавляется серверу, являющемуся членом домена Windows. Так как это первый CA в домене, он в нашей сети будет играть роль корневого (*англ.* Root). Рассмотрим по шагам процедуру установки.

В списке доступных ролей выбираем требующуюся нам Active Directory Certificate Services и нажимаем Next (рис. 5.20). После этого запускается мастер, который сопровождает процесс установки.

В дополнение к обязательной службе «Certification Authority» могут быть установлены дополнительные средства, предоставляющие Web-интерфейс для работы пользователей с CA (рис. 5.21). Это может понадобиться, например, для выдачи сертификатов удаленным или

внешним пользователям, не зарегистрированным в домене. Для выполнения данной лабораторной работы эта служба не понадобится.

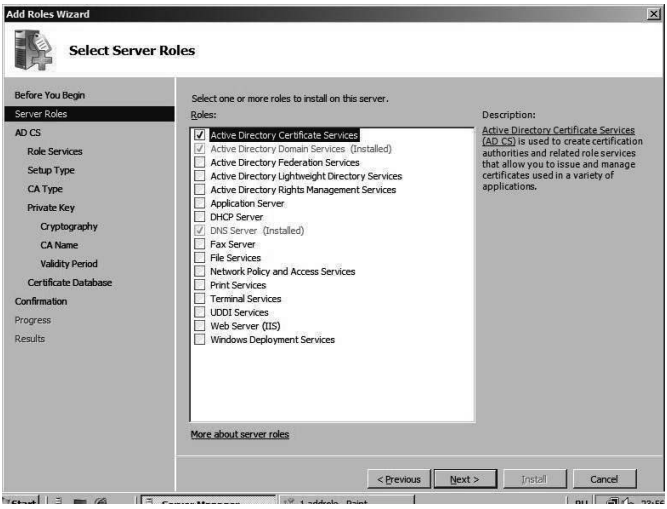


Рис. 5.20. Выбор добавляемой роли

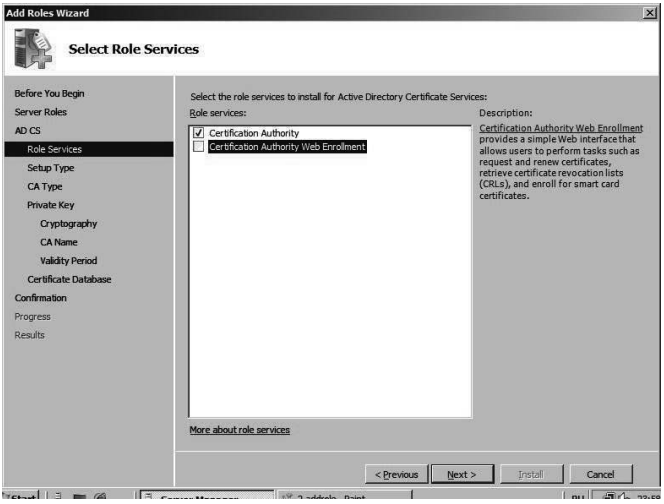


Рис. 5.21. Выбор устанавливаемых компонент

Следующий шаг — определения типа центра сертификации. Он может быть корпоративным (*англ.* Enterprise) или отдельно стоящим (*англ.* Standalone), см. рис. 5.22. Разница заключается в том, что Enterprise CA может быть установлен только на сервер, являющийся членом домена, так как для его работы требуется служба каталога Active Directory. Standalone CA может работать вне домена, например, обрабатывая запросы пользователей, полученные через Web-интерфейс. Для выполнения лабораторной работы нужно выбрать версию Enterprise.

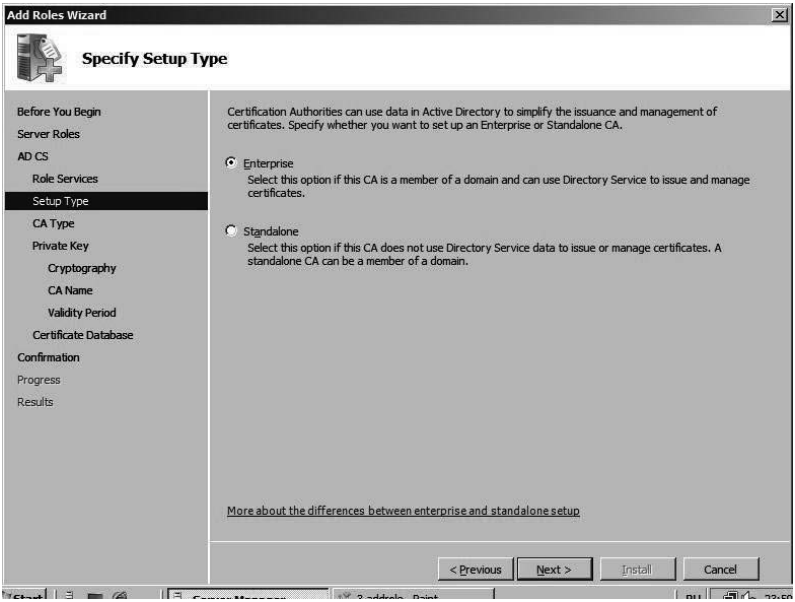


Рис. 5.22. Выбор типа центра сертификации

Следующее окно мастера позволяет определить, создается корневой (*англ.* Root) или подчиненный (*англ.* Subordinate) CA, см. рис. 5.23. В нашем примере развёртываемый CA является первым и единственным, поэтому выбираем вариант Root.

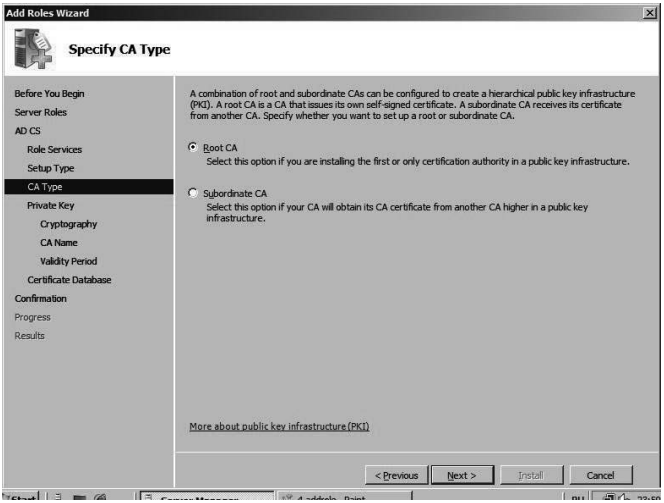


Рис. 5.23. Выбор типа центра сертификации (продолжение)

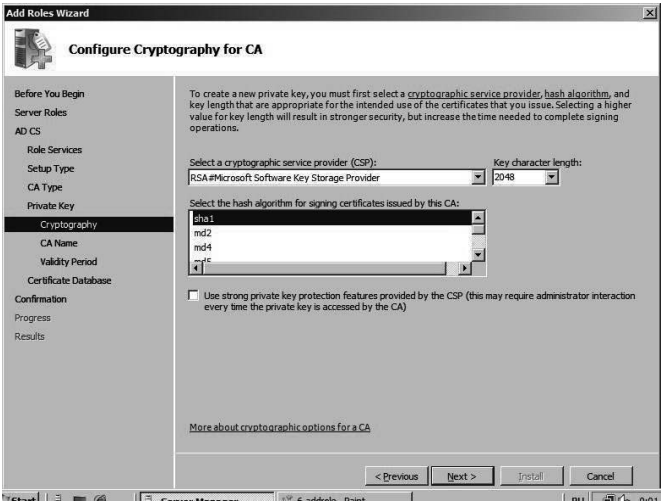


Рис. 5.24. Выбор криптографического провайдера
и алгоритма хеширования

Создаваемый центр сертификации должен будет использовать при работе как минимум одну ключевую пару — открытый и секрет-

ный ключ (иначе он не сможет подписывать выпускаемые сертификаты). Поэтому для продолжения установки мастер запрашивает, нужно ли создать новый секретный ключ, или будет использоваться уже существующий (тогда надо будет указать, какой ключ использовать). В нашей лабораторной работе надо создать новый ключ. При этом потребуется выбрать «криптографический провайдер» (программный модуль, реализующий криптоалгоритмы) и алгоритм хеширования. Согласимся с настройками по умолчанию (рис. 5.24).

Далее потребуется указать имя CA, размещение базы сертификатов и файлов журнала, подтвердить сделанные настройки. После этого, роль будет установлена.

Задание 1.

На учебном сервере или виртуальной машине установите роль Active Directory Certificate Services с настройками, аналогичными рассмотренным выше.

Управлять работой CA можно из оснастки Certification Authority, которая должна появиться в разделе Administrative Tools (рис. 5.25).

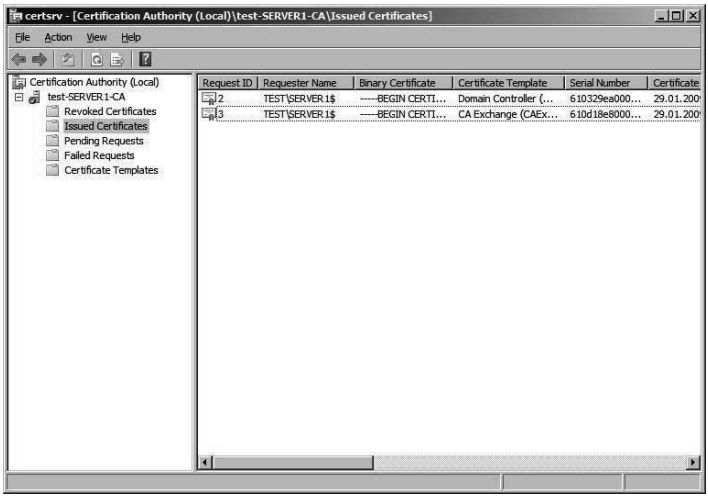


Рис. 5.25. Управление центром сертификации

Как видно на рис. 5.25, только что установленный Enterprise CA уже выпустил некоторое количество сертификатов для служебных целей, в частности, сертификаты для контроллеров домена. В свойствах данного сервера (пункт Properties контекстного меню) можно посмотреть сделанные настройки. Если выбрать закладку Policy Module и там нажать кнопку Properties, можно увидеть текущую настройку, определяющую порядок выдачи сертификатов (рис. 5.26).

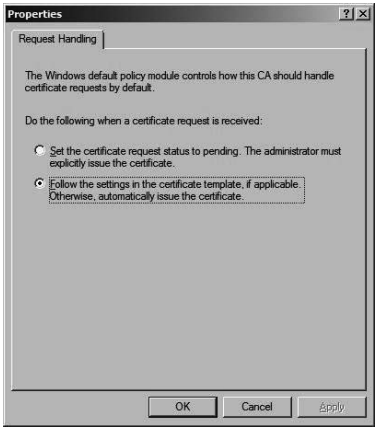


Рис. 5.26. Настройки, определяющие порядок выпуска сертификатов

В выбранном на рис. 5.26 случае, после запроса сертификат выдается в соответствии с настройками шаблона сертификата (или автоматически, если настроек нет). Возможен вариант, когда запрос помещается в очередь ожидающих, и сертификат выпускается только после утверждения администратором.

Задание 2.

Ознакомьтесь с текущими настройками центра сертификации.

Опишите, какие шаблоны сертификатов (*англ.* Certificate Templates) определены, и для каких целей служит каждый тип сертификатов.

Посмотрите, какие сертификаты выпущены (*англ.* Issued Certificates), есть ли отозванные сертификаты (*англ.* Revoked Certificates).

Теперь рассмотрим процесс получения цифрового сертификата. Сделать это можно с помощью оснастки Certificates, с которой мы познакомились в предыдущей работе. Если она не установлена, запустите консоль mmc и добавьте эту оснастку для текущей учетной записи.

Запустим оснастку, откроем раздел, посвященный сертификатам пользователя (*англ.* Personal) и запросим сертификат (рис. 5.27). Из перечня предложенных шаблонов сертификатов выберем User. Данный тип сертификатов может использоваться для шифрования файлов с помощью EFS (Encrypted File System — шифрующая файловая система), защиты электронной почты и аутентификации пользователей.

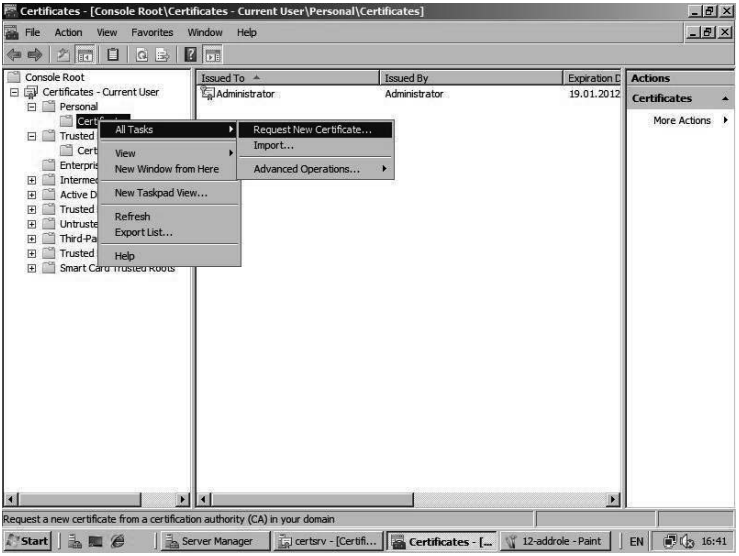


Рис. 5.27. Запрос сертификата

Для пользователя будет сгенерирована ключевая пара, и на основе данных, взятых из базы службы Active Directory и шаблона, будет выпущен сертификат, удостоверяющий открытый ключ. Этот сер-

тификат будет виден и в оснастке Certification Authority в списке выпущенных данным сервером.

Задание 3.

1. Запросите сертификат для одного из пользователей.

2. После получения изучите состав сертификата, его назначение.

3. Выполните экспорт сертификата (в оснастке Certificates выделите сертификат и в контекстном меню выберите All Tasks -> Export). Обратите внимание, что можно экспортировать только сертификат или сертификат вместе с секретным ключом (private key). Второй вариант надо использовать аккуратно, чтобы кто-нибудь не узнал ваш секретный ключ шифрования. Такой тип экспорта нужен, если вы хотите сохранить резервную копию ключевой пары и сертификата.

5.8. ШИФРОВАНИЕ ДАННЫХ ПРИ ХРАНЕНИИ — ФАЙЛОВАЯ СИСТЕМА EFS

Цель работы.

Приобретение практических навыков защиты данных при хранении с помощью шифрования встроенными средствами ОС Windows.

Используемые программные средства.

Компьютер или виртуальная машина с ОС Windows Server 2008 и установленными ролями Active Directory Domain Services (контроллер домена) и Active Directory Certificate Services (центр сертификации). Желательно наличие другого компьютера или виртуальной машины, входящей в тот же домен, на которой будет выполняться работа.

Теоретические сведения.

Шифрующая файловая система (*англ.* Encrypting File System, EFS) появилась в операционных системах семейства Windows, начиная с Windows 2000. Она позволяет шифровать отдельные папки и файлы на томах с файловой системой NTFS. Рассмотрим этот механизм подробнее.

Сначала несколько слов о рисках, которые можно снизить, внедрив данный механизм. Повышение мобильности пользователей приводит к тому, что большое количество конфиденциальных данных оказывается на дисках ноутбуков и съемных носителях. Вероятность того, что подобное устройство будет украдено или временно попадет в чужие руки, существенно выше чем, например, для жесткого диска корпоративного персонального компьютера (хотя и в этом случае, возможны кражи или копирование содержимого накопителей). Если данные хранить в зашифрованном виде, то даже если носитель украден, конфиденциальность данных нарушена не будет. В этом и заключается цель использования EFS.

Следует учитывать, что для передачи по сети, зашифрованный EFS файл будет расшифрован, и для защиты данных в этих случаях надо использовать дополнительные механизмы.

Описание работы.

Рассмотрим работу EFS. Пусть имеется сервер с ОС Windows Server 2008, входящий в домен, и три учетные записи, обладающие административными правами на сервере (одна из них — встроенная административная запись Administrator).

Пользователь User1 хочет защитить конфиденциальные файлы. Тут надо отметить, что хотя шифровать с помощью EFS можно и отдельные файлы, рекомендуется применять шифрование целиком к папке.

User1 с помощью оснастки Certificates запрашивает сертификат (можно выбрать шаблон User или Basic EFS). Теперь у него появляется ключевая пара и сертификат открытого ключа, и можно приступать к шифрованию.

Чтобы зашифровать папку, в ее свойствах на вкладке General нажимаем кнопку Advanced и получаем доступ к атрибуту, указывающему на шифрование файла.

Работа EFS организована так, что одновременно сжатие и шифрование файлов и папок осуществляться не может. Поэтому нельзя

разом установить атрибуты «Compress contents to save disk» и «Encrypt contents to secure data» (рис. 5.28).

Рис. 5.28. В свойствах папки устанавливаем шифрование

При настройках по умолчанию зашифрованная папка выделяется в проводнике зеленым цветом. Для зашифровавшего файл пользователя порядок работы с ним не изменится.

Теперь выполним «переключение» пользователей и зайдем в систему под другой учетной записью, обладающей административными правами, но не являющейся встроенной административной записью. Пусть это будет User2.

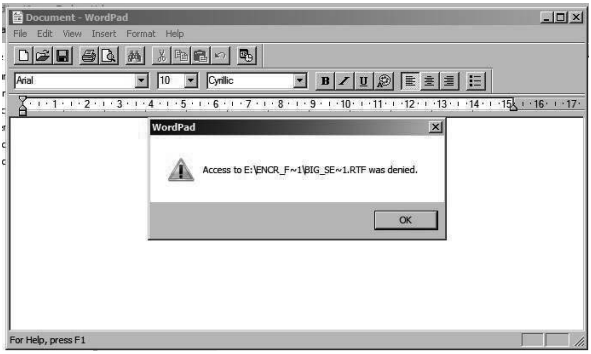


Рис. 5.29. Другой пользователь прочитать файл не сможет

Несмотря на то, что User2 имеет такие же разрешения на доступ к файлу, что и User1, прочитать он его не сможет (рис. 5.29).

Также User2 не сможет скопировать файл, так как для этого надо его расшифровать. При этом необходимо учитывать, что User2 может переименовать или вообще удалить файл или папку (у него есть на это разрешения).

Задание 1.

1. Работая под первой учетной записью, запросите сертификат (если он не был получен ранее), после чего зашифруйте папку с тестовым файлом, который не жалко потерять. Проверьте, что будет происходить при:

- добавлении файлов в папку;

- переименовании папки;

- копировании папки на другой диск с файловой системой NFTS на том же компьютере;

- копировании папки на сетевой диск или диск с FAT.

2. Убедитесь, что другой пользователь не сможет прочитать зашифрованный файл.

3. Снова зайдите под первой учетной записью. В оснастке Certificates удалите сертификат пользователя (несмотря на выдаваемые системой предупреждения). Завершите сессию пользователя в системе и войдите заново. Попробуйте открыть зашифрованный файл.

Как вы убедились, если сертификат и соответствующая ему ключевая пара удалены, пользователь не сможет прочитать зашифрованные им же данные. В частности поэтому в EFS введена роль агента восстановления. Он может расшифровать зашифрованные другими пользователями данные.

Реализуется это следующим образом. Файл шифруется с помощью симметричного криптоалгоритма на сгенерированном системой случайном ключе (назовем его K1). Ключ K1 шифруется на открытом ключе пользователя, взятом из сертификата, и хранится вместе с зашифрованным файлом. Также хранится K1, зашифрованный на от-

крытом ключе агента восстановления. Теперь либо пользователь, осуществлявший шифрование, либо агент восстановления могут файл расшифровать.

При настройке по умолчанию, роль агента восстановления играет встроенная учетная запись администратора домена.

Задание 2.

Зайдите в систему под встроенной учетной записью администратора и расшифруйте папку.

То, какой пользователь является агентом восстановления, задается с помощью групповых политик. Запустим оснастку Group Policy Management. В политике домена найдем группу Public Key Policies и там Encrypting File System, где указан сертификат агента восстановления (рис. 5.30). Редактируя политику (пункт Edit в контекстном меню, далее Policies->Windows Settings-> Security Settings -> Public Key Policies -> Encrypting File System), можно отказаться от присутствия агентов восстановления в системе или наоборот, указать более одного агента (рис. 5.31).

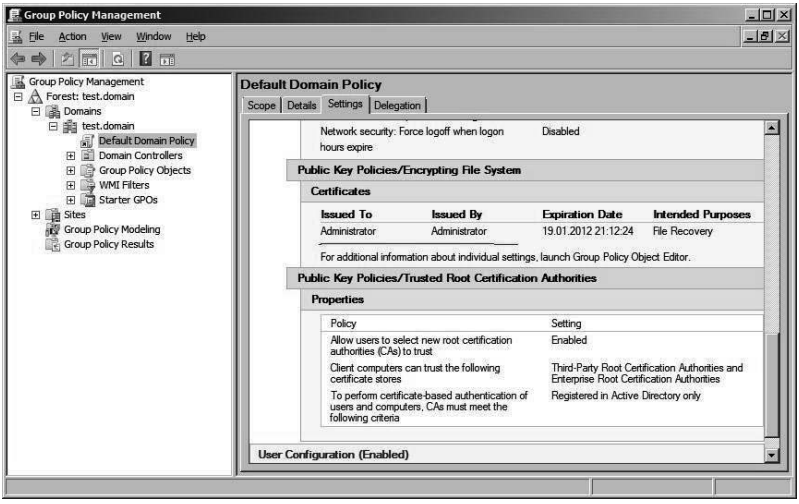


Рис. 5.30. Агент восстановления

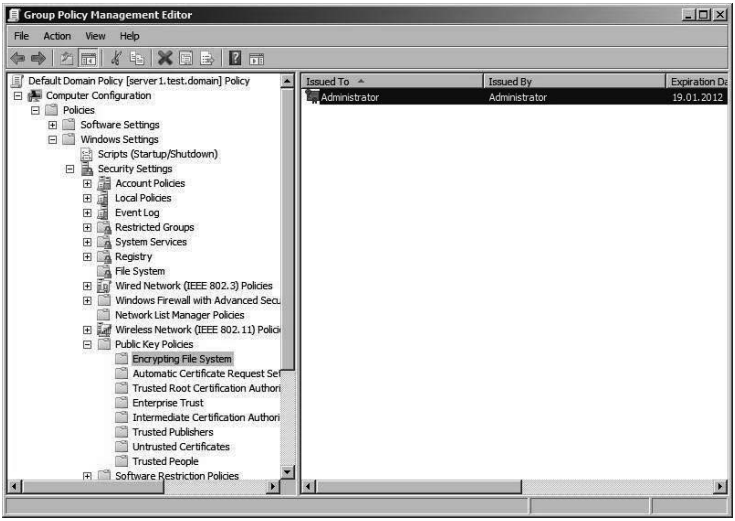


Рис. 5.31. Изменение агента восстановления

Задание 3.

1. Отредактируйте политику таким образом, чтобы убрать из системы агента восстановления (удалите в политике сертификат). Выполнив команду «gpupdate /force» (меню Start->run-> gpupdate /force) примените политику.

2. Повторив действия из предыдущих заданий, убедитесь, что теперь только тот пользователь, который зашифровал файл, может его расшифровать.

3. Теперь вернем в систему агента восстановления, но будем использовать новый сертификат. В редакторе политик находим политику Encrypting File System и в контекстном меню выбираем Create Data Recovery Agent. Это приведет к тому, что пользователь Administrator получит новый сертификат и с этого момента сможет восстанавливать шифруемые файлы.

Теперь рассмотрим, как можно предоставить доступ к зашифрованному файлу более чем одному пользователю. Такая настройка возможна, но делается она для каждого файла в отдельности.

В свойствах зашифрованного файла откроем окно с дополнительными параметрами, аналогичное представленному на рис. 5.28 для папки. Если нажать кнопку Details, будут выведены подробности относительно того, кто может получить доступ к файлу. На рис. 5.32 видно, что в данный момент это пользователь User1 и агент восстановления Administrator. Нажав кнопку Add можно указать сертификаты других пользователей, которым предоставляется доступ к файлу.

Рис. 5.32. Данные о пользователях, которые могут расшифровать файл

Задание 4.

Зашифруйте файл. Предоставьте другому пользователю, не являющемуся агентом восстановления, возможность расшифровать данный файл. Проверьте работу выполненных настроек.

5.9. ИСПОЛЬЗОВАНИЕ MICROSOFT SECURITY ASSESSMENT TOOL

Цель работы.

Приобретение практических навыков оценки рисков организации, связанных с информационной безопасностью, с использованием ПО Microsoft Security Assessment Tool (MSAT).

Используемые программные средства.

Компьютер или виртуальная машина с ОС Windows, программа MSAT, подключение к сети Интернет.

Описание работы.

В ходе данной лабораторной работы мы познакомимся с разработанной Microsoft программой для самостоятельной оценки рисков, связанных с безопасностью — Microsoft Security Assessment Tool (MSAT). Она доступна на сайте Microsoft по ссылке

http://www.microsoft.com/ru-ru/download/details.aspx?id=12273.

Как отмечают разработчики, приложение предназначается для организаций с числом сотрудников менее 1000 человек, чтобы содействовать лучшему пониманию потенциальных проблем в сфере безопасности. В ходе работы пользователь, выполняющий роль аналитика, ответственного за вопросы безопасности, отвечает на две группы вопросов.

Первая из них посвящена бизнес-модели компании, и призвана оценить риск для бизнеса, с которым компания сталкивается в данной отрасли и в условиях выбранной бизнес-модели. Создается так называемый профиль риска для бизнеса (ПРБ).

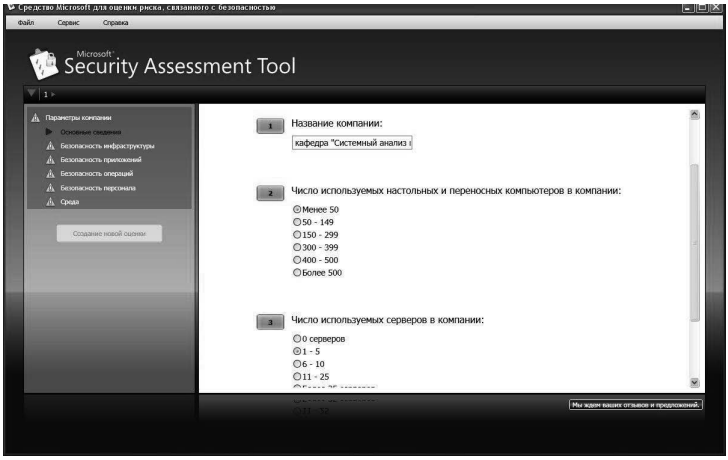


Рис. 5.33. Информация о компании

Вопросы этого этапа разбиты на 6 групп. Первая касается общих сведений о компании (рис. 5.33): название, число компьютеров, серверов и т. д. Вторая группа вопросов озаглавлена «Безопасность инфраструктуры». Примеры вопросов: «Использует ли компания подключение к Интернет?», «Размещаются ли службы, используемые как внешними, так и внутренними клиентами, в одном и том же сегменте?». Остальные группы — «Безопасность приложений», «Безопасность операций», «Безопасность персонала», «Среда».

Когда проведен первый этап оценки, полученная информация обрабатывается (для этого требуется подключение к Интернет), после чего начинается второй этап анализа. Для технических специалистов он будет более интересен, так как касается используемых в компании политик, средств и механизмов защиты (рис. 5.34).

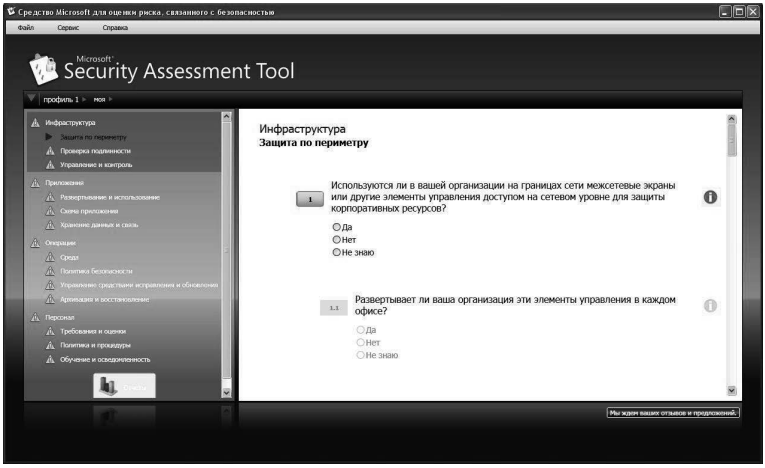


Рис. 5.34. Анализ используемых механизмов защиты

Вопросы организованы в соответствии с концепцией многоуровневой защиты. Сначала рассматривается защита инфраструктуры (защита периметра, аутентификация…), затем вопросы защиты на уровне приложений, далее проводится анализ безопасности операций (определена ли политика безопасности, политика резервного копиро-

вания…), последняя группа вопросов касается работы с персоналом (обучение, проверка при приеме на работу и т. д.).

Во многом тематика вопросов соответствует разделам стандартов ISO 17799 и 27001, рассмотренных в разделе 4.3 данного пособия.

После ответа на все вопросы программа вновь обращается к удаленному серверу и генерирует отчеты. Наибольший интерес для технических специалистов представляет «Полный отчет». В частности, он содержит предлагаемый список приоритетных действий. Фрагмент списка представлен в табл. 5.3.

Таблица 5.3

Список предлагаемых в отчете MSAT действий

Список приоритетных действий	
Предмет анализа	***Рекомендация***
Высокий приоритет	
Операции > Управление средствами исправления и обновления > Управление средствами исправления	Наличие политики исправлений и обновлений для операционных систем является полезным начальным шагом, однако необходимо разработать такую же политику и для приложений. Разработайте такую политику, пользуясь сведениями, доступными в разделе, посвященном передовым методикам. Сначала установите исправления для внешних приложений и приложений Интернета, затем для важных внутренних приложений и, наконец, для не особо важных приложений.

Задание.

Подробно опишите реально существующее или вымышленное малое предприятие: сферу деятельности, состав и структуру информационной системы, особенности организации процесса защиты информации, применяемые методы и средства.

С помощью программы MSAT проведите оценку рисков для предприятия.

5.10. ЛАБОРАТОРНЫЙ ПРАКТИКУМ «KASPERSKY SECURITY CENTER»

Целью представляемого лабораторного практикума является изучение средств управления антивирусной защитой уровня предприятия на примере программного продукта «Лаборатории Касперского» (далее — ЛК) Kaspersky Security Center. Данный продукт позволяет осуществить централизованное развертывание и управление средствами антивирусной защиты корпоративной сети.

Более подробно о Security Center можно узнать на домашней странице продукта по ссылке http://www.kaspersky.ru/security-center.

Для выполнения лабораторных работ необходимы предварительные знания по администрированию Windows Server 2008 и Windows 7, достаточные для организации лабораторного стенда и установки требуемого программного обеспечения.

Также желательно знакомство с антивирусными продуктами ЛК, так как в ходе лабораторного практикума основное внимание будет уделено вопросам их централизованного администрирования.

Используемые программные средства.

Security Center может быть установлен на компьютер под управлением 32 или 64-разрядной ОС семейства Windows. Это может быть как клиентская ОС, так и серверная. Минимальные требования к оборудованию — процессор с частотой 1 ГГц, 512 Мб оперативной памяти, 1 Гб свободного пространства на жестком диске.

В описаниях лабораторных будут задействоваться четыре виртуальные машины, характеристики которых представлены в табл. 5.4. Аналогичные работы можно провести и с использованием физических компьютеров. Кроме того, объединив некоторые роли (например, контроллер домена и сервер антивирусной защиты), можно выполнить работы на меньше числе виртуальных или физических машин.

При планировании выполнения лабораторных работ на виртуальных машинах лучше конфигурировать виртуальные машины на

использование 1024 Мб ОЗУ, иначе в процессе удаленной установки ПО Kaspersky Endpoint Security могут возникнуть ошибки из-за недостатка памяти.

Дистрибутив Security Center можно загрузить по ссылке http://www.kaspersky.ru/downloads-security-center. По этой же ссылке доступна и документация по продукту.

Таблица 5.4

Описание используемых виртуальных машин

Имя виртуальной машины	Описание
Serv.labs.local	Операционная система Windows Server 2008 R2. Сервер, выполняющий роли контроллера домена «labs.local», dns-сервера, а также dhcp-сервера во внутренней виртуальной сети (раздает ip-адреса из диапазона 192.168.15.10/24 — 192.168.15.254/254). Один сетевой интерфейс. Статический IP-адрес 192.168.15.1/24.
AVServ.labs.local	Операционная система Windows Server 2008 R2. Сервер — член домена labs.local — будет конфигурироваться как сервер управления антивирусной защитой. Два сетевых интерфейса. Внутренний: статический IP-адрес 192.168.15.2/24. Через «внешний» интерфейс сервер подключен к Интернет (требуется для получения обновлений).
Station1.labs.local	Операционная система Windows 7 Professional. Компьютер — член домена labs.local. Один сетевой интерфейс, настройки получаются динамически.

Окончание табл. 5.4

Имя виртуальной машины	Описание
NB1.labs.local	Операционная система Windows 7 Professional. Компьютер — член домена labs.local; считаем, что это ноутбук, используемый как внутри корпоративной сети, так и за ее пределами, что потребует особой настройки политик антивирусной защиты. Один сетевой интерфейс, настройки получаются динамически. Должна быть предусмотрена возможность конфигурирования интерфейса или на подключение к «внутренней» сети (компьютер в корпоративной сети), или на подключение к Интернет (компьютер вне корпоративной сети).

Для организации полнофункциональной антивирусной защиты понадобятся лицензионные ключи для антивирусов или можно воспользоваться пробными версиями продуктов.

Кроме того, для иллюстрации порядка удаленной установки продуктов ЛК при наличии на компьютере несовместимых приложений предлагается на виртуальную машину NB1.labs.local установить антивирусный продукт другого производителя. Например, это может быть продукт Microsoft Security Essentials.

В ходе лабораторных работ также понадобится программа-архиватор, например, свободно распространяемый 7zip (http://www.7-zip.org/). А для тестирования работоспособности антивирусной защиты будет использоваться тестовый файл eicar.com, доступный на сайте http://www.eicar.org в разделе «Download Anti-Malware Testfile». Этот исполняемый файл DOS не является вирусом (все, что он делает — выводит строчку "EICAR-STANDARD-ANTIVIRUS-TEST-FILE!"),

но для целей тестирования его сигнатура внесена в базы антивирусных программ.

5.10.1. Установка Kaspersky Security Center

Цель работы.

Данная лабораторная работа посвящена установке сервера управления антивирусной защитой Security Center.

Предварительные сведения.

До того, как приступить к установке, надо определиться с общим сценарием развертывания антивирусной защиты. Два основных сценария, предлагаемых разработчиками Security Center:

- развертывание антивирусной защиты внутри организации;
- развертывание антивирусной защиты сети организации-клиента (используется организациями, выступающими в роли сервис-провайдеров). Эта же схема может использоваться внутри организации, имеющей несколько удаленных подразделений, компьютерные сети которых администрируются независимо от сети головного офиса.

В данных лабораторных работах будет реализовываться первый сценарий. Если планируется использовать второй, то дополнительно потребуется установка и настройка компонента Web-Console. И здесь нужно сказать об архитектуре Security Center. Он включает в себя следующие компоненты:

1. *Сервер администрирования*, который осуществляет функции централизованного хранения информации об установленных в сети организации программах ЛК и управления ими.

2. *Агент администрирования* осуществляет взаимодействие между Сервером администрирования и программами ЛК, установленными на компьютере. Есть версии Агента для разных операционных систем — Windows, Novell и Unix.

3. *Консоль администрирования* предоставляет пользовательский интерфейс для управления Сервером. Консоль администрирования выполнена в виде компонента расширения к Microsoft Management

Console (MMC). Она позволяет подключаться к Серверу администрирования как локально, так и удаленно, по локальной сети или через Интернет.

4. *Kaspersky Security Center Web-Console* предназначена для контроля состояния антивирусной защиты сети организации-клиента, находящейся под управлением Kaspersky Security Center. Использование этого компонента в рамках данного лабораторного практикума изучаться не будет.

Рассмотрим теперь рекомендуемый порядок действий по развертыванию в сети организации системы антивирусной защиты, основанной на продуктах ЛК. Он включает в себя следующие этапы.

1. Установка и настройка Сервера и Консоли администрирования.

2. Создание групп администрирования и распределение по ним клиентских компьютеров.

3. Удаленная установка на клиентские компьютеры Агента администрирования и антивирусных программ ЛК.

4. Обновление сигнатурных баз программ ЛК на клиентских компьютерах.

5. Настройка уведомлений о событиях антивирусной защиты.

6. Запуск задачи проверки по требованию и проверка работы уведомлений о событиях на клиентских компьютерах.

7. Анализ отчетов.

8. Настройка автоматической установки антивирусных программ на новые компьютеры в сети.

В данной лабораторной работе будет рассмотрено выполнение первого этапа. На рис. 5.35 представлена схема лабораторного стенда, имитирующего защищаемую сеть (он также был описан ранее в табл. 5.4). Цель этой лабораторной работы — установить сервер и консоль администрирования Security Center на сервер AVServ.

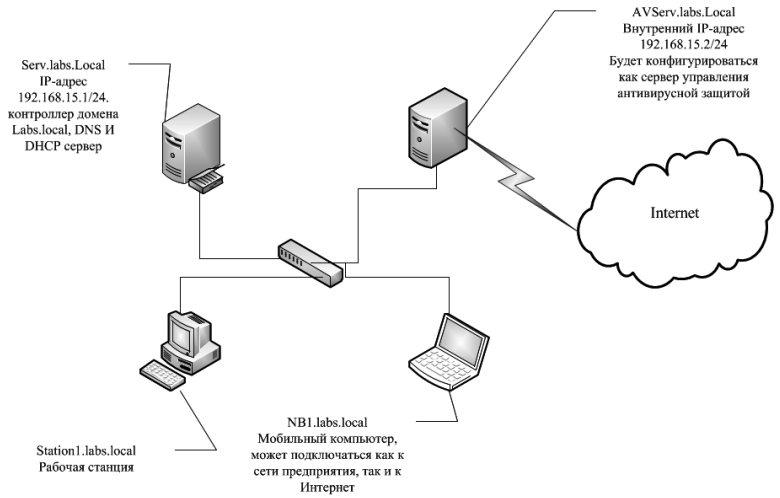


Рис. 5.35. Схема сети

Таблица 5.5

Различия в версиях дистрибутива Kaspersky Security Center 9.0

Компонент	**Полная версия**	**Lite-версия**
Дистрибутив Сервера администрирования	да	да
Дистрибутив Kaspersky Endpoint Security for Windows	да	нет
Дистрибутив Агента администрирования	да	да
Microsoft SQL 2005 Server Express Edition	да	нет
Microsoft .NET Framework 2.0 SP1	да	нет
Microsoft Data Access Component 2.8	да	нет
Microsoft Windows Installer 3.1	да	нет
Kaspersky Security Center System Health Validator	да	нет

Дистрибутив Security Center можно загрузить по ссылке http://www.kaspersky.ru/downloads-security-center. При этом можно выбрать версию скачиваемого дистрибутива — Lite или полную. В табл. 5.5 перечислены различия версий дистрибутивов для версии 9.0,

которая использовалась при подготовке описаний лабораторных работ. Для выполнения лабораторной нужна будет полная версия, так как вместе с установкой сервера администрирования будет устанавливаться СУБД MS SQL Server 2005 Express, которая используется для хранения данных о состоянии антивирусной защиты.

Описание работы.

После завершения подготовительных действий запускаем на сервере AVServ программу инсталляции Security Center. После окна приветствия будет запрошен путь для сохранения файлов, необходимых в процессе инсталляции, появится еще одно окно приветствия и окно с лицензионным соглашением, которое необходимо принять для продолжения процесса установки.

При выборе типа установки отметим пункт «Выборочная», что позволит подробно ознакомиться с перечнем устанавливаемых компонент и применяемых настроек.

Если выбрать вариант «Стандартная», то в результате работы мастера будут установлены Сервер администрирования вместе с серверной версией Агента администрирования, Консоль администрирования, доступные в дистрибутиве плагины управления программами и Microsoft SQL Server 2005 Express Edition (если он ранее не был установлен).

Следующим шагом будет выбор устанавливаемых компонент сервера (рис. 5.36). Нам нужно установить Сервер администрирования, и отметку на этом пункте оставляем.

Технология Cisco NAC, позволяющая проверять безопасность подключающегося к сети мобильного устройства или компьютера, у нас использоваться не будет.

Также в рамках лабораторного практикума не планируется развертывание антивирусной защиты на мобильных устройствах (таких как смартфоны), поэтому указанные компоненты сейчас не устанавливаем.

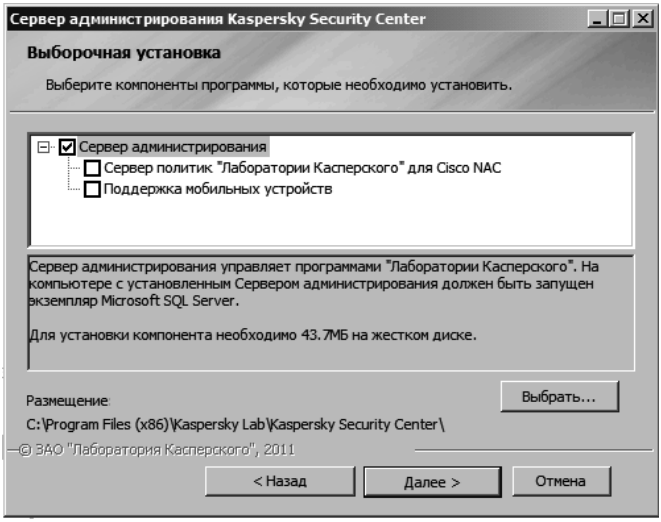


Рис. 5.36. Выбор устанавливаемых компонент сервера

Далее потребуется указать размер защищаемой сети. Для выполнения лабораторных работ нам более всего подходит вариант «менее 100 компьютеров».

Выбираемый размер сети влияет на установку значений ряда параметров, определяющих работу антивирусной защиты (они перечислены в табл. 5.6). Данные настройки можно при необходимости поменять и после установки сервера.

Также потребуется указать учетную запись, от имени которой будет запускаться сервер администрирования, или согласиться с созданием новой записи (рис. 5.37).

В предыдущих версиях ОС Windows (например, при установке на Windows Server 2003) в этом окне может присутствовать вариант «Учетная запись системы». В любом случае, данная запись должна обладать правами администратора, что потребуется как для создания базы данных, так и для последующей работы сервера.

Таблица 5.6

Параметры, устанавливаемые в зависимости от размера сети

Параметр / число компьютеров	**1–100**	**100–1000**	**1000–5000**	**Более 5000**
Отображение в дереве консоли узла подчиненных и виртуальных Серверов администрирования и всех параметров, связанных с подчиненными и виртуальными Серверами	отсутствует	отсутствует	присутствует	присутствует
Отображение разделов **Безопасность** в окнах свойств Сервера и групп администрирования	отсутствует	отсутствует	присутствует	присутствует
Создание политики Агента администрирования с помощью мастера первоначальной настройки	отсутствует	отсутствует	присутствует	присутствует
Распределение времени запуска задачи обновления на клиентских компьютерах случайным образом	отсутствует	в интервале 5 минут	в интервале 10 минут	в интервале 10 минут

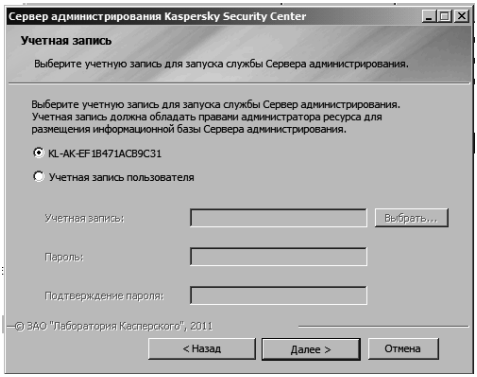
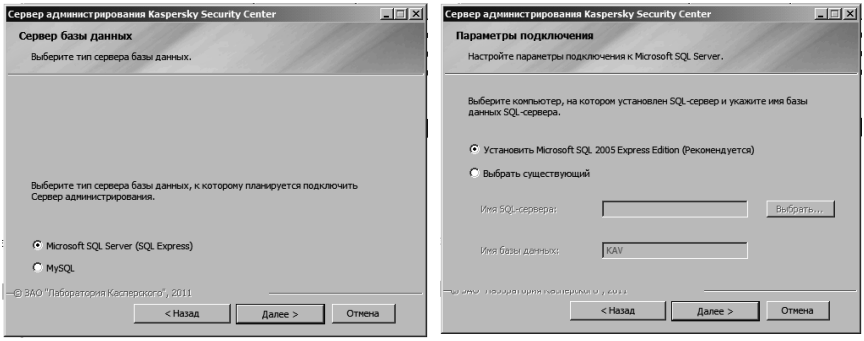


Рис. 5.37. Выбор учетной записи

Следующий шаг — выбор используемого сервера баз данных (рис. 5.38). Для хранения данных Security Center 9.0 может использовать Microsoft SQL Server (версии 2005, 2008, 2008 R2, в том числе и редакции Express 2005, 2008) или MySQL Enterprise. На рис. 5.38, *а* показано окно выбора типа СУБД. Если выбран сервер MySQL, то потребуется указать имя и номер порта для подключения.

Если использовать существующий экземпляр MS SQL Server, потребуется указать его имя и название базы данных (по умолчанию, она называется KAV). В наших лабораторных работах будет использоваться рекомендуемая конфигурация, подразумевающая установку MS SQL Server 2005 Express вместе с установкой Security Center (рис. 5.38, *б*).



а) *б*)

Рис. 5.38. Выбор сервера баз данных

После выбора SQL Server в качестве используемой СУБД надо указать режим аутентификации, который будет использоваться при работе с ним. Здесь оставляем настройку по умолчанию — режим аутентификации Microsoft Windows (рис. 5.39).

Для хранения инсталляционных пакетов и распространения обновлений сервер администрирования будет использовать папку, предоставляемую в общий доступ. Можно указать существующую

папку или создать новую. Имя общего ресурса по умолчанию — \\<имя_сервера>\KLSHARE.

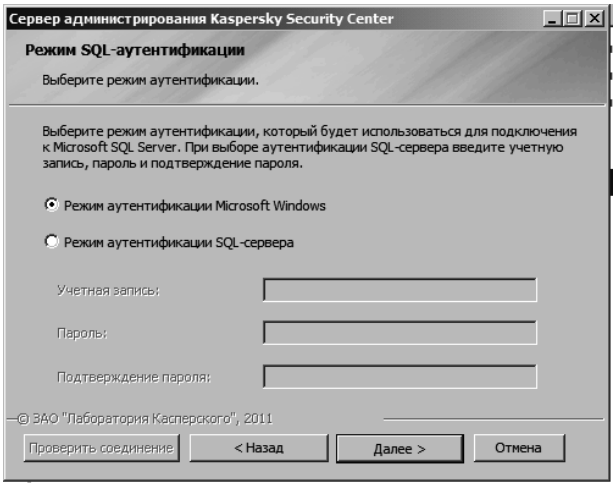


Рис. 5.39. Выбор режима аутентификации для MS SQL Server

Также предоставляется возможность указания номеров портов, используемых для подключения к серверу Security Center. По умолчанию используется TCP-порт 14000, а для защищенного с помощью протокола SSL соединения — TCP-порт 13000. Если после установки к серверу администрирования не удается подключиться, стоит проверить, не блокируются ли эти порты межсетевым экраном Windows. Кроме упомянутых выше, для передачи на сервер информации о выключении компьютеров используется UDP-порт 13000.

Далее потребуется указать способ идентификации сервера администрирования. Это может быть ip-адрес, имена DNS или NetBIOS. В используемой для лабораторного практикума виртуальной сети организован домен Windows и присутствует DNS-сервер, поэтому будем использовать доменные имена (рис. 5.40).

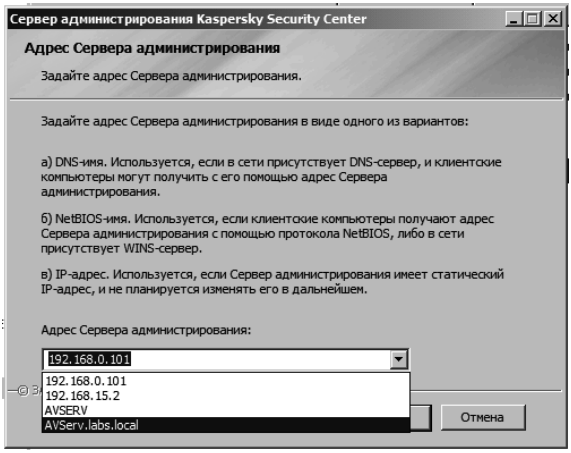


Рис. 5.40. Выбор способа идентификации сервера

Следующее окно позволяет выбрать устанавливаемые плагины для управления антивирусными программами ЛК. Забегая вперед, можно сказать, что развертываться будет продукт Kaspersky Endpoint Security 8 for Windows, плагин для которого нам и понадобится (рис. 5.41).

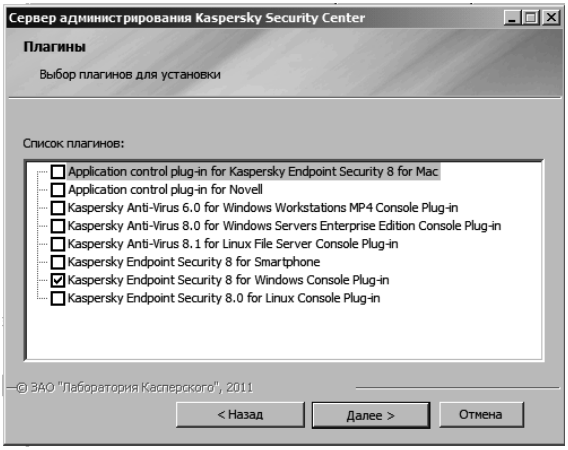


Рис. 5.41. Выбор устанавливаемых плагинов

После этого выбранные программы и компоненты будут установлены на сервер. По завершении установки будет запущена консоль администрирования или, если вы убрали «галочку» в последнем окне мастера установки, запустите ее из меню Пуск -> Программы-> Kaspersky Security Center.

Задание 1.

В соответствии с описанием выполните установку сервера администрирования на виртуальную машину AVServ.

При запуске консоли выполняется начальная настройка сервера. На первом шаге можно указать коды активации или файлы лицензионных ключей для антивирусных продуктов ЛК. Если у вас есть «корпоративный» ключ на несколько компьютеров, при настройках по умолчанию ключ будет автоматически распространяться сервером на клиентские компьютеры.

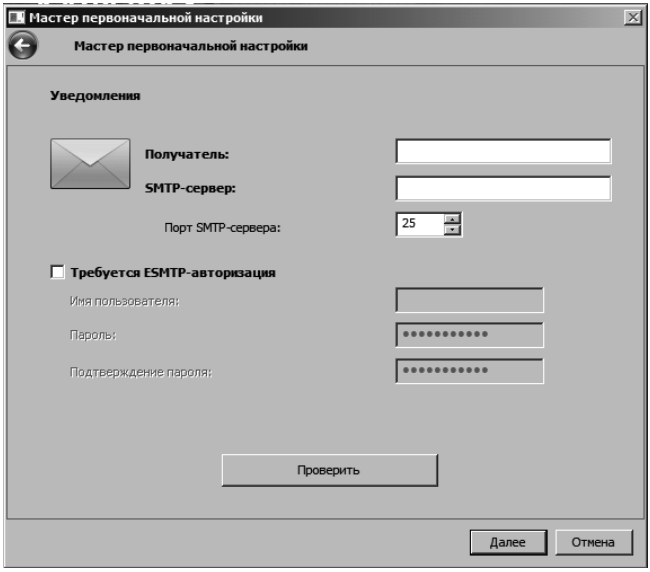


Рис. 5.42. Настройка параметров оповещения по электронной почте

Также можно согласиться или отказаться от использования Kaspersky Security Network (KSN), удаленного сервиса по предоставлению доступа к базе знаний Лаборатории Касперского о репутации файлов, Интернет-ресурсов и программного обеспечения.

Следующий шаг — настройка параметров для оповещения администратора антивирусной защиты по электронной почте. Надо указать почтовый адрес, smtp-сервер и, при необходимости, параметры для авторизации на сервере (рис. 5.42). Если в лаборатории нет подходящего почтового сервера, можно этот шаг пропустить и сделать настройки позже.

Если доступ в Интернет осуществляется через прокси-сервер, понадобится указать его параметры. После прохождения этого этапа будет выполнено автоматическое создание стандартных политик, групповых задач и задач администрирования. Они будут более подробно рассмотрены в следующих лабораторных работах.

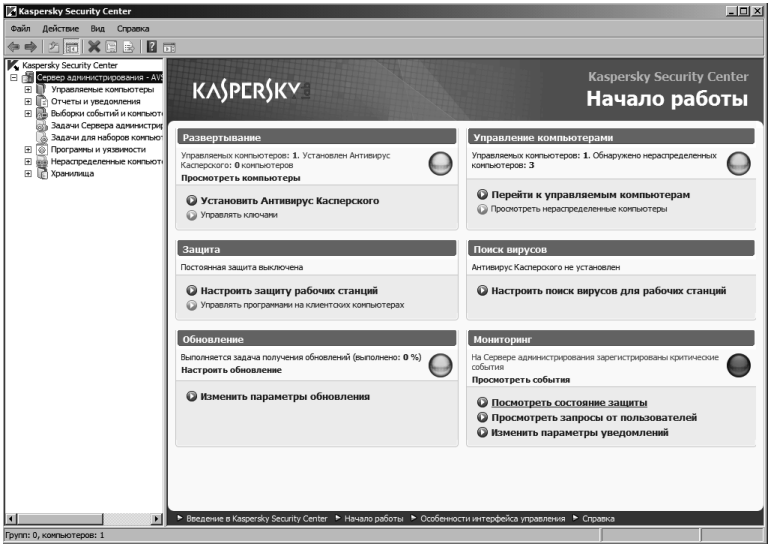


Рис. 5.43. Окно консоли администрирования — 1 критическое событие

Следующий шаг — автоматический запуск загрузки обновлений. Если загрузка началась успешно, можно, не дожидаясь ее окончания, нажать кнопку «Далее» и после окончания работы мастера начальной настройки перейти в основное окно Консоли администрирования (рис. 5.43). Там должно отобразиться, что в сети есть один управляемый компьютер (вместе с сервером администрирования на компьютер AVServ был установлен и агент администрирования), на котором отсутствует антивирусная защита. Это расценивается как критическое событие.

Задание 2.

Выполните начальную настройку сервера.

Отдельно консоль администрирования можно установить из папки Console дистрибутивного диска, запустив программу Setup. Если используется дистрибутив, скачанный из Интернет, то надо отрыть указанную в начале установки папку для сохранения дистрибутивных файлов. По умолчанию это папка C:\KSC9\russian\Console.

Рис. 5.44. Выбор удаленного сервера

Задание 3.

На виртуальную машину Station1.labs.local установите консоль администрирования Security Center. Проверьте возможность подключения к серверу AVServ.labs.local. Для этого в окне консоли надо указать его адрес или имя (рис. 5.44), а также согласиться на получение сертификата сервера (рис. 5.45).

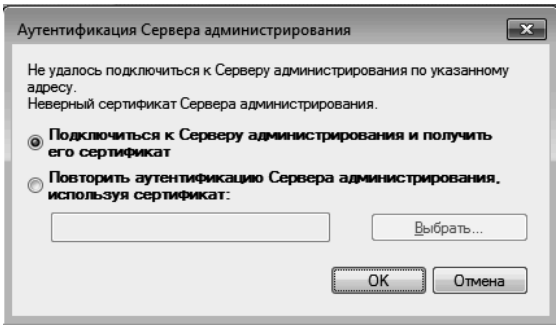


Рис. 5.45. Получение сертификата сервера

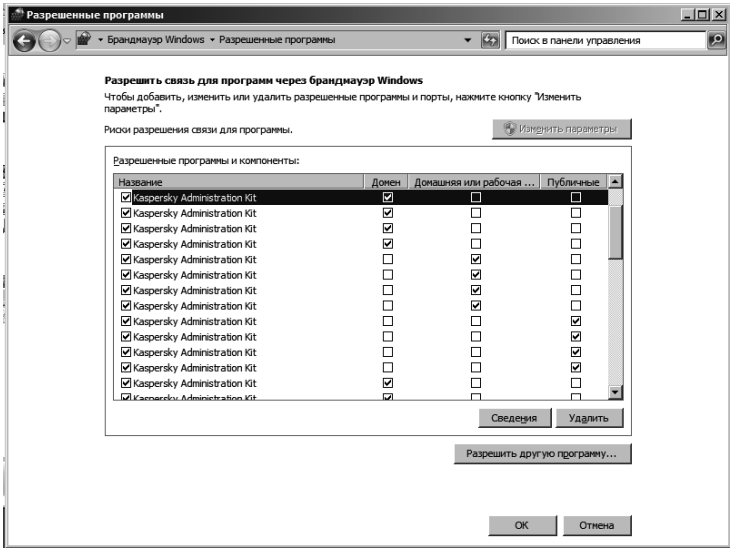


Рис. 5.46. Разрешающие настройки на брандмауэре Windows

Если подключиться не удалось, проверьте, не блокируются ли на сервере AVServ порты, используемые для подключения к серверу Security Center (см. выше). Настройку можно проверить через Панель управления: `Система и безопасность -> Брандмауэр Windows -> Разрешить запуск программы через брандмауэр Windows`. Соответствующие разрешающие настройки должны присутствовать, см. рис. 5.46 (названия правил остались как в предыдущей версии продукта — Kaspersky Administration Kit).

5.10.2. Развертывание антивирусной защиты: установка агентов администрирования, проверка совместимости

Цель работы.

Лабораторная работа посвящена вопросам автоматизированного развертывания средств антивирусной защиты в компьютерной сети. Рассматриваются задачи установки агента администрирования и проверки компьютеров на наличие ПО, несовместимого с продуктами «Лаборатории Касперского».

Описание работы.

В ходе выполнения предыдущей лабораторной работы были установлены сервер администрирования и консоль управления. Связь сервера администрирования с клиентскими компьютерами обеспечивает агент администрирования. Поэтому его необходимо установить на каждый клиентский компьютер, который будет подключен к системе централизованного управления антивирусной защитой. На тот компьютер, где установлен сервер администрирования, агент отдельно устанавливать не требуется, так как вместе с сервером была автоматически установлена серверная версия агента (ее особенность в том, что агент взаимодействует только с сервером администрирования, установленным на том же компьютере).

Запустим все виртуальные машины, которые составляют наш лабораторный стенд, и рассмотрим процесс автоматизированного развертывания агентов администрирования. На сервере AVServ запустим консоль администрирования Kaspersky Security Center (также

можно управлять сервером с рабочей станции Station1, на которую в ходе выполнения первой работы была установлена консоль администрирования). Для начала ознакомимся со списком управляемых компьютеров, где уже есть агент администрирования. Раскрыв узел *Управляемые компьютеры*, перейдем на вкладку *Компьютеры* (рис. 5.47). В выводимом списке отображается только один компьютер — AVServ, на котором установлена серверная версия агента администрирования. Статус «критический» установлен из-за того, что антивирусное ПО на данном компьютере пока что отсутствует.

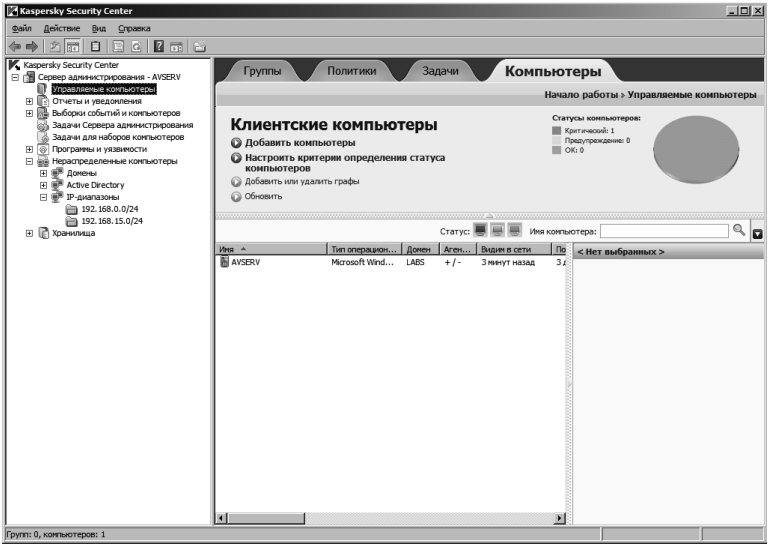


Рис. 5.47. Просмотр списка управляемых компьютеров

Первое, что необходимо сделать в процессе развертывания агента — выполнить обнаружение компьютеров. В консоли администрирования перейдем на узел *Нераспределенные компьютеры* (рис. 5.48). Поиск можно выполнить путем опроса компьютеров в сети Windows, используя информацию службы Active Directory или по диапазону ip-адресов. Конкретный способ зависит от организации защищаемой се-

ти. Например, если в сети есть компьютеры, не включённые в домен Windows, поиск в диапазоне ip-адресов может быть предпочтительнее, чем получение информации из Active Directory.

В нашем примере подойдут все три вышеназванных способа. Воспользуемся поиском в сети Windows — в соответствующем разделе в окне консоли администрирования нажмем ссылку *Опросить сейчас*.

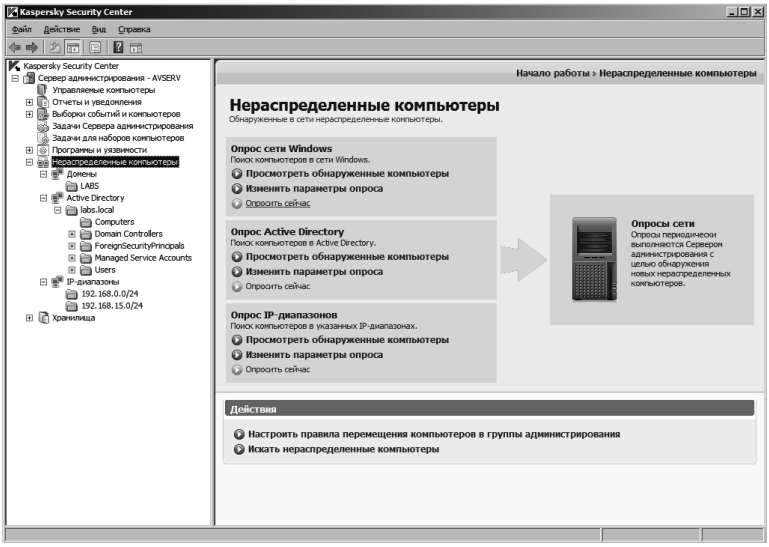


Рис. 5.48. Поиск компьютеров в сети

После того, как опрос закончится (процесс его прохождения иллюстрируется «линейкой состояния»), раскрываем узел, соответствующий найденному домену Labs, и видим все обнаруженные компьютеры. Выделим их и нажмем ссылку *Уставить программу* (рис. 5.49). Это приведет к запуску мастера удаленной установки.

Здесь необходимо пояснить некоторые особенности, связанные с удаленной установкой ПО средствами Security Center. Перед тем как начать развертывание программы, надо создать инсталляционный па-

кет. Делается это через узел *Хранилища->Инсталляционные пакеты* (рис. 5.49). Там надо запустить мастер, который запросит название установочного пакета и путь к дистрибутивным файлам.

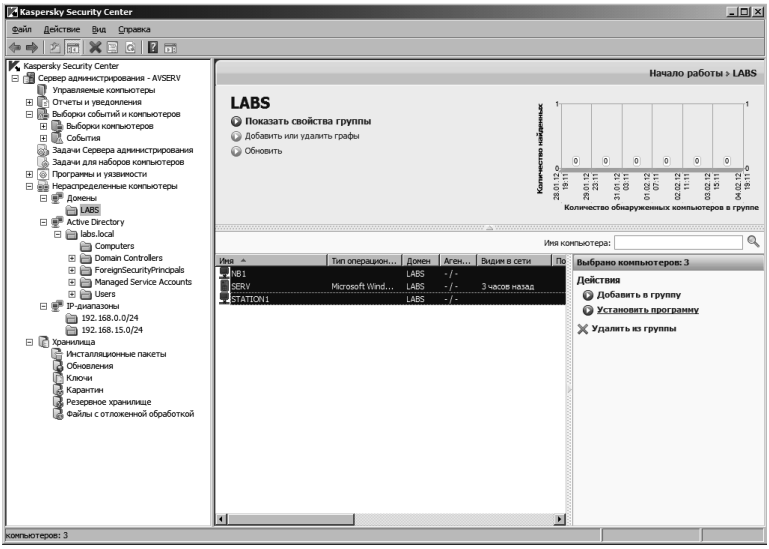


Рис. 5.49. Обнаруженные компьютеры

В нашем случае установка Kaspersky Security Center 9.0 производилась из полной версии дистрибутива и оба установочных пакета — для агента администрирования и для антивирусного продукта Kaspersky Endpoint Security 8 for Windows — были автоматически подготовлены. Так что в первом окне мастера установки понадобится только указать нужный пакет (рис. 5.50).

Следующий шаг — выбор способа, который будет использоваться для установки распространяемого пакета. В связи с тем, что агент администрирования на клиентские компьютеры еще не установлен, оставляем отметки *Средствами Windows из папки общего доступа* и *Не устанавливать программу, если она уже установлена* (рис. 5.51).

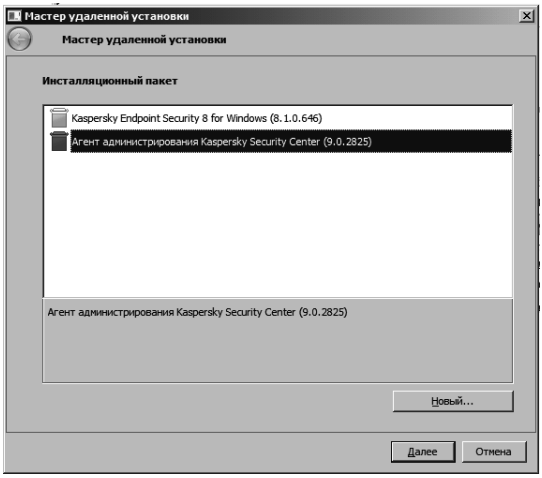


Рис. 5.50. Мастер удаленной установки:
выбор инсталляционного пакета

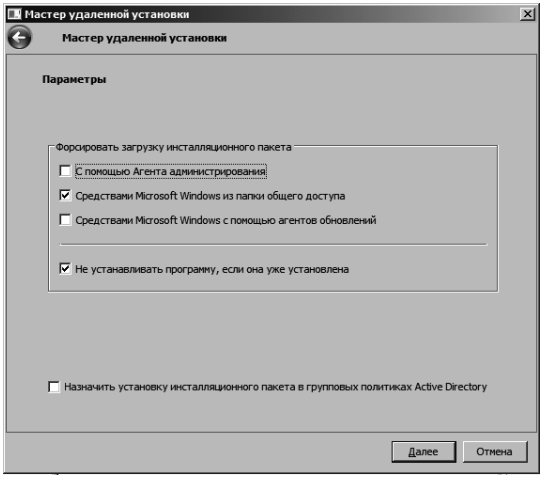


Рис. 5.51. Мастер удаленной установки:
выбор способа установки

Для того чтобы установка средствами Windows прошла правильно, на клиентском компьютере необходимо открыть порты TCP 139 и 445, UDP 137 и 138. По умолчанию эта настройка сделана на

всех клиентских компьютерах, включенных в домен Windows (в случае клиента с Windows 7 есть некоторые особенности, которые описаны ниже). Если в сети присутствуют компьютеры, не включенные в домен, попытка автоматической установки агента администрирования на них может закончиться неудачно. В этом случае можно воспользоваться утилитой подготовки компьютера к удаленной установке riprep.exe, которая сделает все необходимые настройки (см. далее).

После выбора инсталляционных пакетов и способа установки потребуется указать учетную запись, от имени которой будет проводиться установка. Этот этап можно пропустить, если на компьютерах уже установлен агент администрирования. В нашем случае надо нажать кнопку *Добавить* и указать запись, обладающую правами администратора (рис. 5.52). Если компьютеры не включены в домен, таких записей может понадобиться несколько.

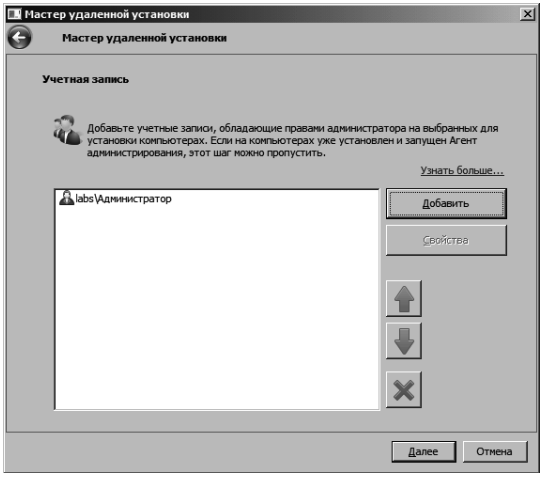


Рис. 5.52. Указание учетной записи

Следующее окно определяет, как будет производиться перезагрузка компьютера, если она понадобится в результате установки па-

кета. Настройка по умолчанию — запросить у пользователя разрешение на перезагрузку.

Для целей нашей лабораторной более предпочтительным будет вариант *Перезагрузить компьютер*. А при развертывании нового ПО в действующей сети предприятия, когда установка производится на серверы, стоит выбрать *Не перезагружать компьютер* и выполнить перезагрузку вручную позже.

Далее согласимся с перемещением компьютеров, на которые будет установлен агент администрирования, в группу *Управляемые компьютеры* (рис. 5.53).

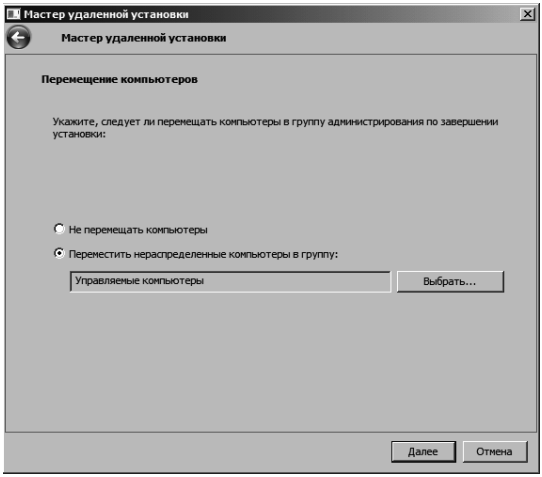


Рис. 5.53. Перемещение компьютеров
в группу Управляемые компьютеры

В следующих окнах мастера потребуется несколько раз нажать кнопку *Далее*, чтобы подтвердить сделанный выбор и запустить созданную задачу на выполнение. Перейдя на узел *Задачи для наборов компьютеров*, можно отследить процесс установки и оценить его результат. В примере, представленном на рис. 5.54, на одном компьютере установка прошла успешно, на двух — завершилась неудачей.

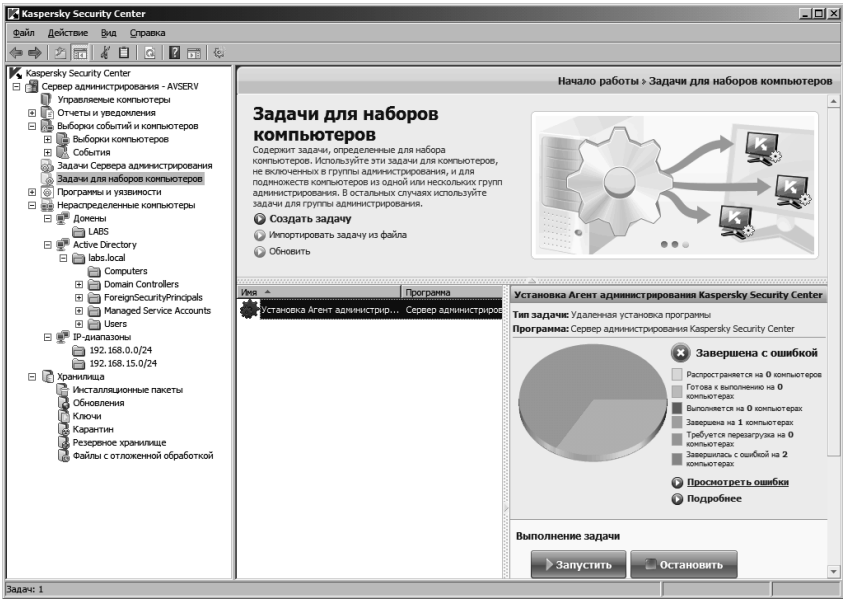


Рис. 5.54. Установка агента администрирования

Если в представленном на рис. 5.54 окне с отчетом перейти по ссылке *Подробнее*, то мы увидим, что удачно завершилась установка агента на сервер Serv, а на компьютеры NB1 и Station1 не удалось установить программу по причине того, что «Возможно компьютер отключен от сети». Это может произойти из-за того, что компьютер выключен, возникли неполадки в сетевой инфраструктуре или настройки межсетевого экрана на компьютере не позволяют сделать удаленную установку. В нашем примере причина неудачи — настойки межсетевого экрана.

В подобных случаях, при установке антивируса Касперского на компьютеры с Windows 7, рекомендуется предварительно настроить общий доступ к сетевым папкам.

В случае если компьютер с Windows 7 включен в домен, надо открыть *Панель управления -> Центр управления сетями и общим доступом*, далее выбрать *Изменить дополнительные параметры обще-*

го доступа и в блоке *Доступ к общим папкам* выбрать *Включить общий доступ, чтобы сетевые пользователи могли читать и записывать файлы в общих папках* (рис. 5.55). Такую настройку надо сделать на компьютерах NB1 и Station1.

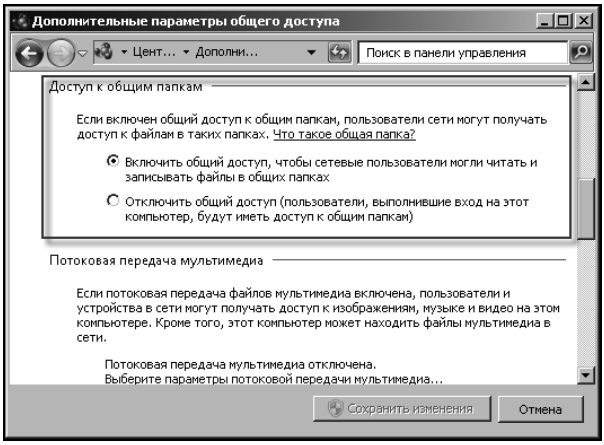


Рис. 5.55. Настройка общего доступа
к папкам в Windows 7

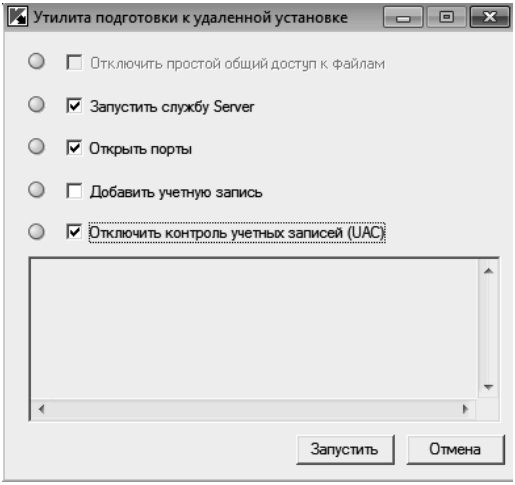


Рис. 5.56. Запуск утилиты ripгep

В более общем случае проблему с настройками Windows можно решить с помощью запуска на компьютерах утилиты riprep.exe. На сервере AVServ она находится в папке, куда устанавливался Security Center: C:\Program Files (x86)\Kaspersky Lab\Kaspersky Security Center. Можно скопировать утилиту в папку общего доступа и запустить оттуда на машинах, где установка не прошла. Запустив утилиту, надо отметить необходимые изменения (рис. 5.56), а после их внесения может понадобиться перезагрузка. Для большой сети подобную задачу можно автоматизировать с помощью административных скриптов.

После произведённых изменений повторно запустим задачу установки агентов администрирования (рис. 5.54, кнопка *Запустить* внизу справа). Теперь установка должна пройти успешно.

Задание 1.

Выполните развертывание агентов администрирования Security Center на виртуальных машинах лабораторного стенда.

После выполненной установки агентов администрирования в группе *Управляемые компьютеры* должны оказаться все 4 виртуальные машины лабораторного стенда. Если отображается только один сервер AVServ, возможно это просто старые данные, и надо выполнить команду *Обновить* из контекстного меню.

Следующая задача при подготовке к развертыванию антивирусного ПО — поиск установленных на компьютерах программ, несовместимых с антивирусами ЛК. Такие программы придется деинсталлировать.

Провести удаленную деинсталляцию программ с клиентских компьютеров можно, запуская задачи удаленной деинсталляции. Security Center позволяет создавать задачи следующих видов:

- групповые задачи, которые созданы для клиентских компьютеров выбранных групп администрирования;

- задачи для наборов компьютеров, которые созданы для выбранных клиентских компьютеров вне зависимости от того, принадлежат ли эти компьютеры к какой-либо группе администрирования.

Но перед тем как программу деинсталлировать, ее надо найти. В этом может помочь создание отчета о несовместимых программах.

Для этого надо перейти на узел *Отчеты и уведомления*, там выбрать *Отчет о несовместимых программах* и, при необходимости, в контекстном меню выбрать пункт *Обновить*. В отчет попадут выявленные агентами администрирования программы, которые известны Security Center как несовместимые. В первую очередь, это другие антивирусные продукты.

В нашем случае в отчете указывается на наличие программы Microsoft Security Essentials 2 x64 на компьютере NB1 (рис. 5.57).

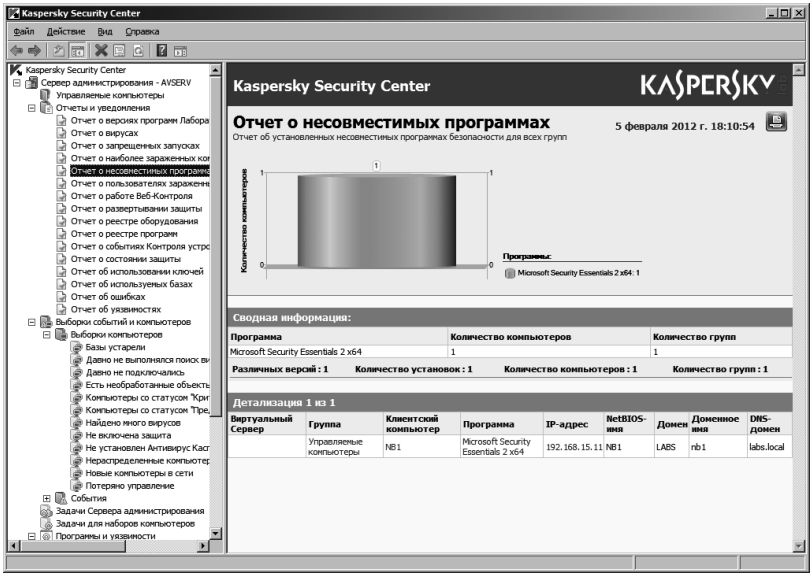


Рис. 5.57. Отчет о несовместимых программах

Найдем NB1 в группе управляемых компьютеров и создадим задачу деинсталляции антивируса Microsoft Security Essentials (рис. 5.58). Обратите внимание, что после запуска мастера активным стал узел *Задачи для наборов компьютеров*.

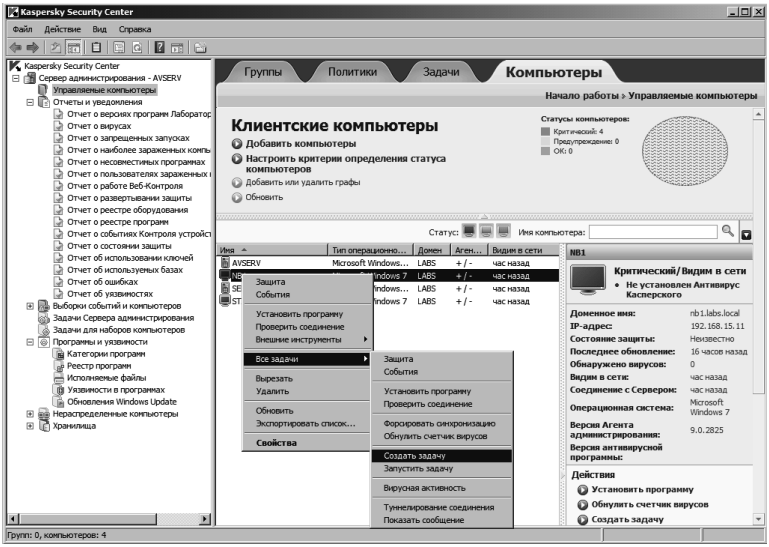


Рис. 5.58. Создание задачи для выбранного компьютера

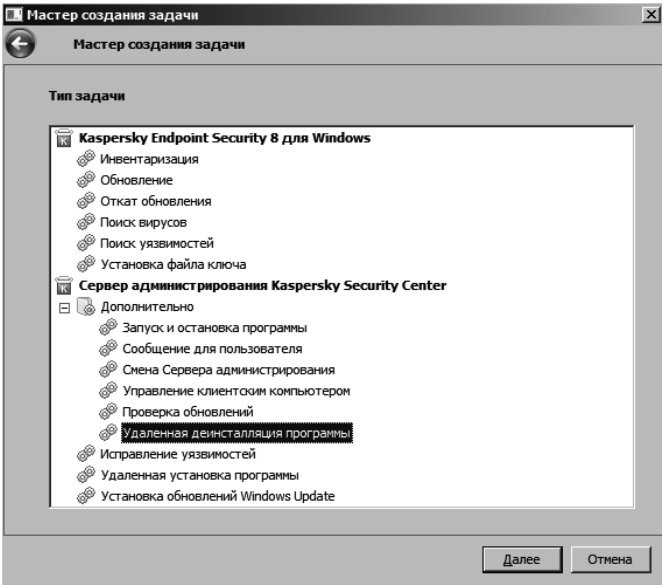


Рис. 5.59. Выбор типа задачи

В первом окне мастера надо указать название создаваемой задачи — «Удаление Security Essentials». Далее потребуется выбрать тип задачи: нас интересует удаленная деинсталляция программы, которая относится к задачам для приложения *Сервер администрирования Kaspersky Security Center*, подгруппа *Дополнительно* (рис. 5.59).

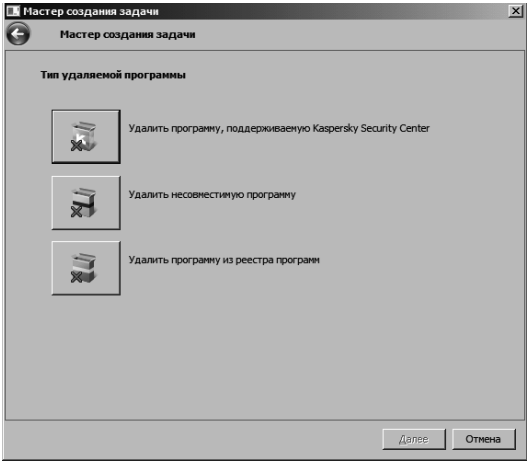


Рис. 5.60. Выбор типа удаляемой программы

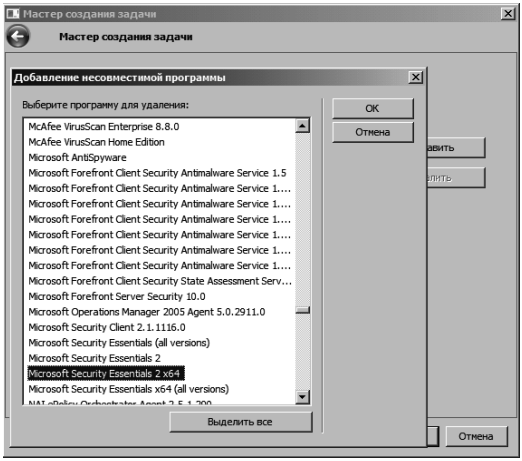


Рис. 5.61. Выбор удаляемой программы

Также нам известно, что данная программа находится в перечне несовместимых с антивирусами ЛК. Это указываем при выборе типа удаляемой программы (рис. 5.60).

В следующем окне нажмем кнопку *Добавить* и в перечне несовместимых программ найдем Microsoft Security Essentials 2 x64 (рис.5.61).

В следующем окне мастера создания задачи понадобится указать способ, которым будет загружаться на клиентский компьютер утилита деинсталляции. Агент администрирования на компьютер NB1 уже установлен, поэтому можно оставить настройки по умолчанию (рис. 5.62). Далее будет запрос на перезагрузку компьютера после деинсталляции, выберем вариант *Перезагрузить компьютер*.

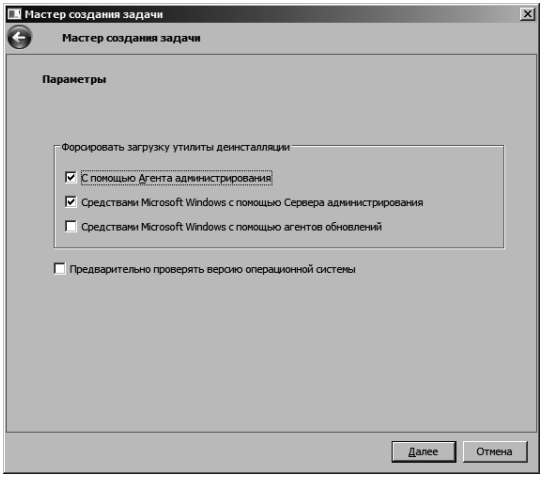


Рис. 5.62. Выбор способа загрузки утилиты деинсталляции

Следующее окно мастера позволяет указать используемую учетную запись. Это нужно в случае, если агент администрирования Security Center на клиентском компьютере недоступен. В нашем случае это окно можно пропустить, так как агент на компьютере NB1 присутствует.

В расписании запуска выберем вариант *Немедленно* и в последнем окне подтвердим создание задачи.

Теперь выделим созданную задачу в узле *Задачи для наборов компьютеров* и дождёмся окончания ее выполнения, на что может потребоваться определенное время. Отчет показал, что задача завершилась, и требуется перезагрузка компьютера (рис. 5.63). После автоматической перезагрузки, задача будет отмечена как успешно завершенная. Таким образом, мы выполнили необходимые действия, предшествующие развертыванию в сети средств антивирусной защиты.

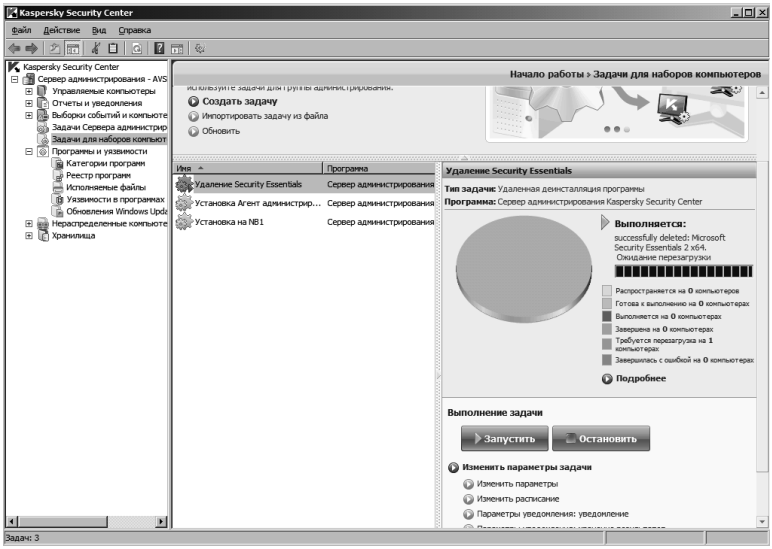


Рис. 5.63. Отчет о выполнении задачи

Задание 2.

Выполните проверку компьютеров на наличие несовместимых приложений и удалите обнаруженные антивирусные программы.

5.10.3. Развертывание антивирусной защиты и управление лицензионными ключами

Цель работы.

Работа посвящена вопросам автоматизации установки антивируса Kaspersky Endpoint Security и управлению лицензионными ключами.

Описание работы.

Итак, у нас имеется 4 компьютера, на одном из которых установлен сервер администрирования Security Center, проведена установка агентов администрирования, и решены проблемы с несовместимыми приложениями. Пора переходить к развертыванию средств антивирусной защиты.

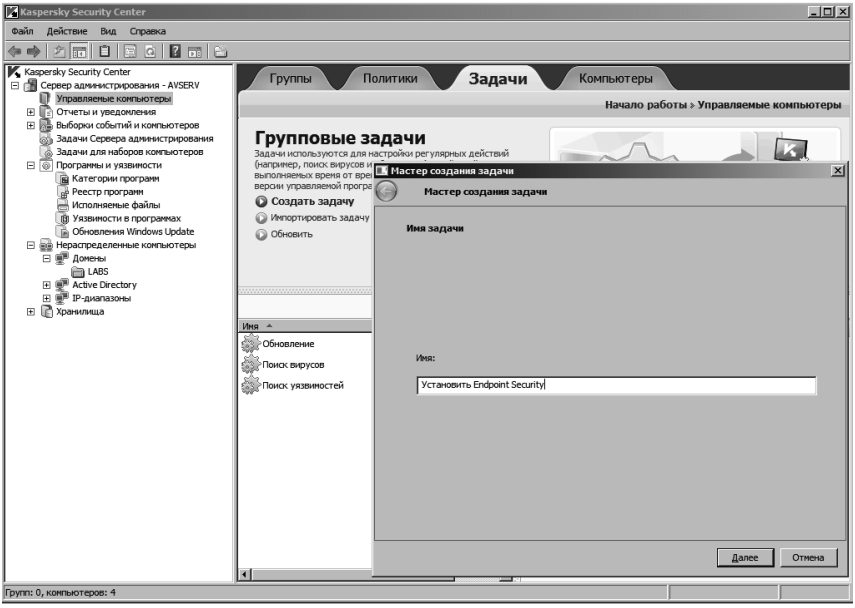


Рис. 5.64. Создание задачи для группы

Запустим три виртуальные машины Serv, AVServ и Station1. Будем считать, что пользователь с ноутбуком NB1 пока не подключился к нашей сети.

Нам требуется развернуть антивирусное решение на всех компьютерах из группы *Управляемые компьютеры*. Это можно сделать, создав задачу для группы. В консоли управления перейдем на соответствующий группе узел и там — на вкладку *Задачи*. Создадим новую задачу «Установить Endpoint Security» (рис. 5.64).

Тип задачи — удаленная установка программы (рис. 5.65). Развертываемый инсталляционный пакет для Kaspersky Endpoint Security уже подготовлен, так что его нужно только выделить в окне мастера создания задачи. Если бы требовалось установить другой продукт, для которого пакет не подготовлен, можно нажать кнопку *Новый* и создать необходимый пакет.

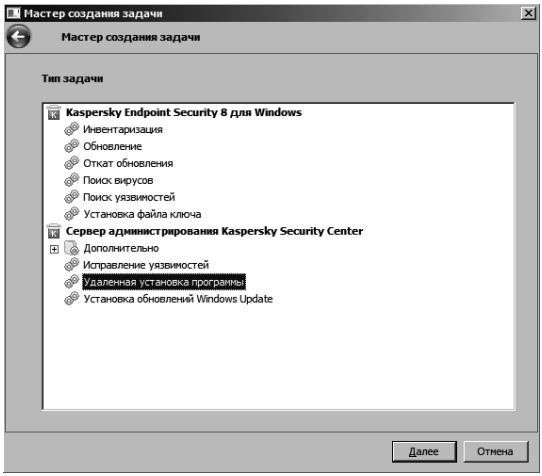


Рис. 5.65. Тип задачи: удаленная установка программы

В следующем окне мастера предлагается дополнительно выполнить установку агента администрирования. В нашем случае этого не требуется (рис. 5.66) — агент уже установлен на всех компьютерах.

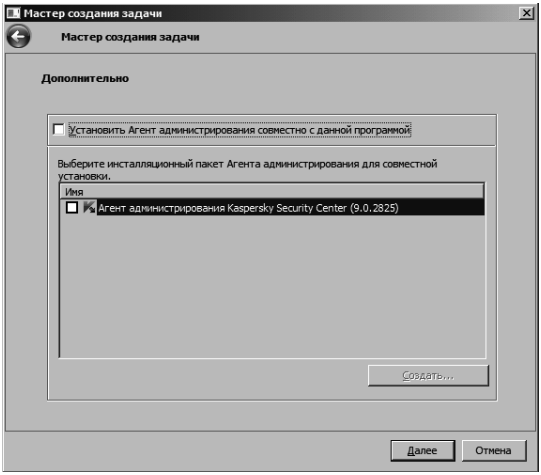


Рис. 5.66. Дополнительная настройка

Далее будет запрос относительно способа загрузки. Оставляем отметки *С помощью агента администрирования* и *Не устанавливать программу, если она уже установлена*. В следующем окне отметим, что после установки надо перезагрузить компьютер. Потом будет запрос учетной записи, от имени которой будут проводиться действия. Но так как установка будет произведена агентом администрирования, запись здесь можно не указывать. В расписании запуска выберем вариант *Немедленно* и подтвердим создание задачи.

Здесь потребуется время на передачу дистрибутива на клиентские компьютеры и установку. В зависимости от быстродействия используемого компьютера, выполнение задачи на виртуальных машинах может занять достаточно продолжительное время. Результат успешной установки представлен на рис. 5.67. Перезагрузка после установки не потребовалась. Обратите внимание, что четверть круговой диаграммы на рисунке окрашена серым — в отношении одного из 4-х компьютеров группы задача не выполнялась, так как этот компьютер отсутствует в сети.

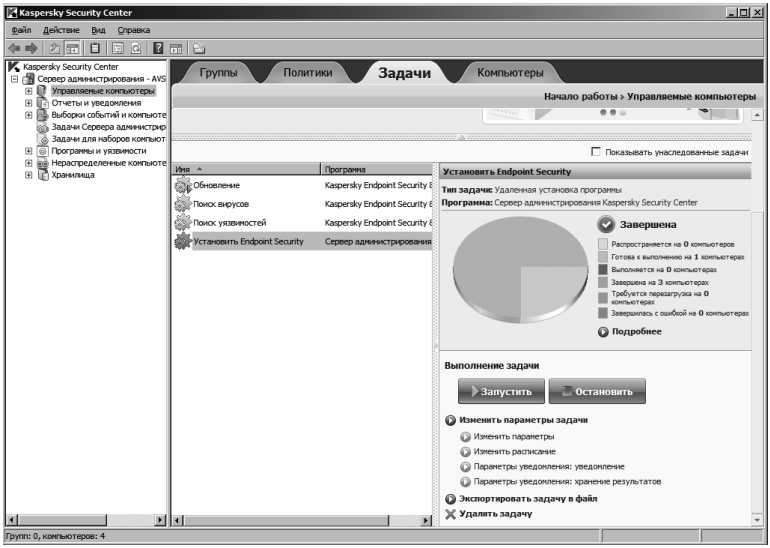


Рис. 5.67. Результат установки

Если в консоли управления перейти на узел *Отчеты и уведомления* и открыть *Отчет о версиях программ Лаборатории Касперского*, можно получить подробные сведения об установленных продуктах.

Задание 1.

Выполните автоматическое развертывание Endpoint Security на компьютерах Serv, AVServ, Station1. Создайте отчет об установленных программах.

Запустим теперь виртуальную машину NB1 (при недостатке памяти можно на время выключить Station1). Спустя несколько минут после загрузки NB1 там автоматически началась установка антивирусного ПО (рис. 5.68). Связано это с тем, что созданная групповая задача будет выполняться на всех компьютерах группы.

Задание 2.

Загрузите виртуальную машину NB1. Убедитесь, что на ней тоже запустится установка антивирусного ПО.

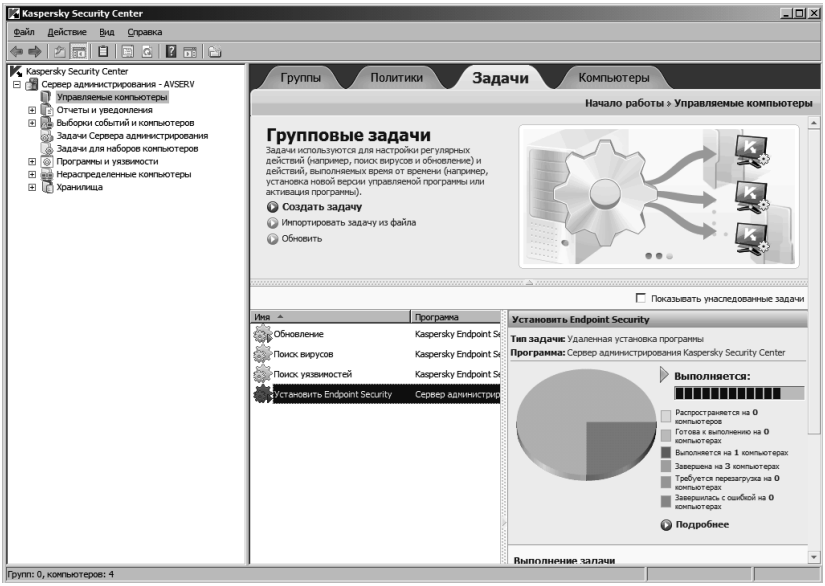


Рис. 5.68. После появления в сети отключенного компьютера установка продолжилась

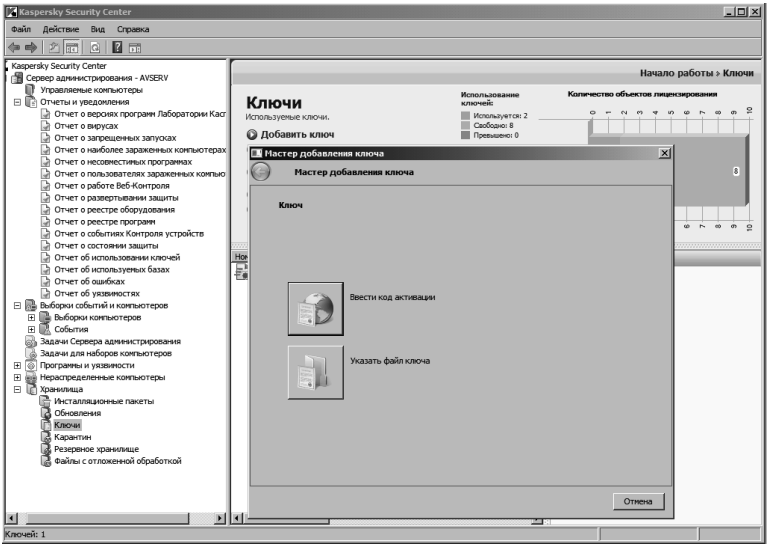


Рис. 5.69. Добавление ключа

Для защиты от нелегального распространения в антивирусных продуктах ЛК используются лицензионные ключи. При отсутствии ключа функциональность продукта ограничивается. Security Center позволяет централизованно управлять ключами в сети предприятия.

Лицензионные ключи можно ввести на стадии первоначальной настройки сервера администрирования или позже, через консоль администрирования (узел *Хранилища*, вложенный узел *Ключи*). Нажав ссылку *Добавить ключ*, можно добавить ключи в хранилище, введя код активации (понадобится подключение к Интернет) или указав файл лицензионного ключа (рис. 5.69). При добавлении в хранилище нового ключа по умолчанию он будет автоматически распространён на управляемые компьютеры.

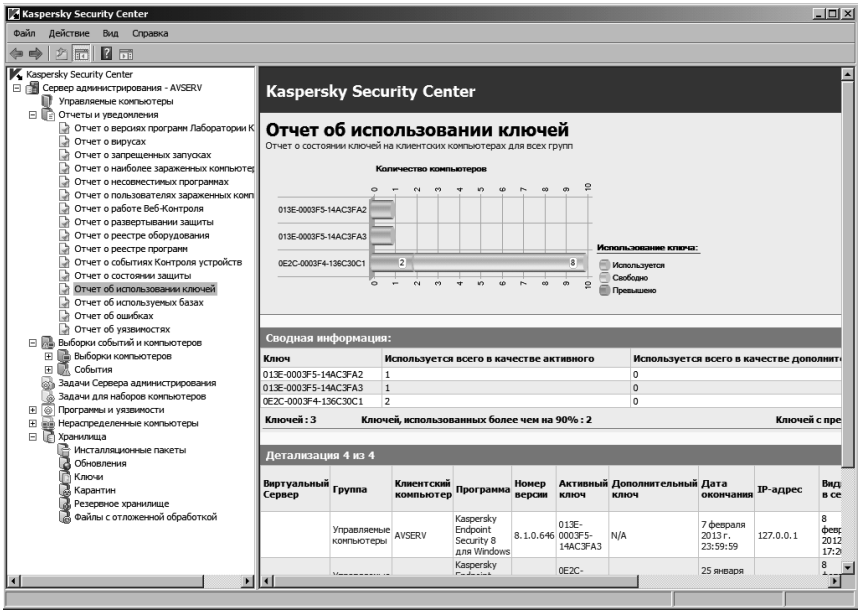


Рис. 5.70. Отчет об использовании ключей

Получить информацию об использовании ключей из хранилища можно из отчета об использовании ключей, который открывается или

по ссылке из окна управления ключами, или из раздела *Отчеты и уведомления -> Отчет об использовании ключей* (рис. 5.70). Там представлена информация о том, какие лицензии и на каких компьютерах используются, сколько имеется свободных лицензий, срок действия ключей.

Задание 3.

Откройте хранилище ключей. Сформируйте отчет об использовании ключей и ознакомьтесь с его содержанием. При необходимости, добавьте ключи в хранилище.

5.10.4. Конфигурирование сервера администрирования

Цель работы.

В работе рассматриваются вопросы, связанные с созданием на сервере администрирования политик и задач, а также их выполнением на клиентских компьютерах.

Описание работы.

В предыдущих лабораторных работах выполнялось развертывание антивирусного программного обеспечения на компьютерах защищаемой сети. Сейчас рассмотрим вопросы, связанные с настройкой сервера администрирования и подключенных к нему клиентов. Основные средства администрирования здесь — это задачи (рис. 5.71) и политики.

Задача — это именованное действие, производимое на сервере администрирования или инициируемое сервером на клиенте. Задачи могут определяться для сервера администрирования, набора компьютеров или группы компьютеров. Некоторые из задач создаются автоматически, другие создает администратор (например, задачи удаленной установки или деинсталляции программ, которые создавались в ходе предыдущей лабораторной). Начнем с предустановленных задач сервера, перечень которых представлен на рис. 5.71.

Если воспользоваться ссылкой *Изменить параметры задачи*, то из настроек расписания можно узнать, что загрузка обновлений про-

изводится каждый час. Перейдя на вкладку *Параметры* можно изучить перечень источников обновления (рис. 5.72).

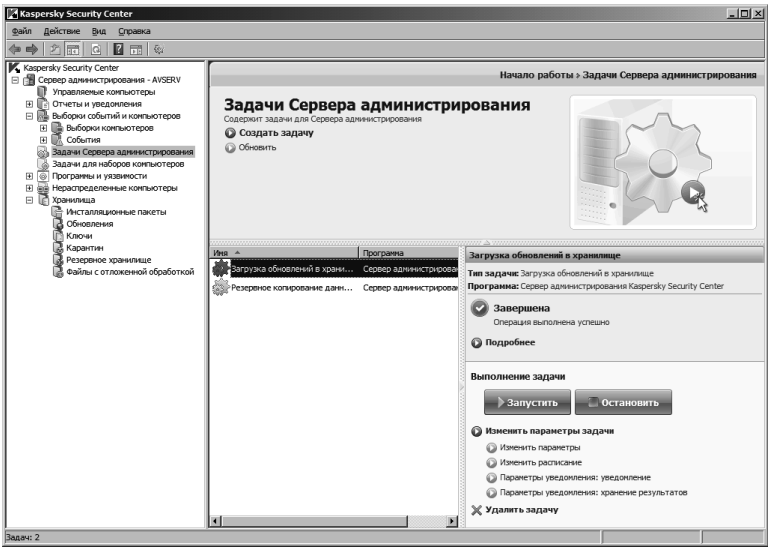


Рис. 5.71. Задачи сервера администрирования

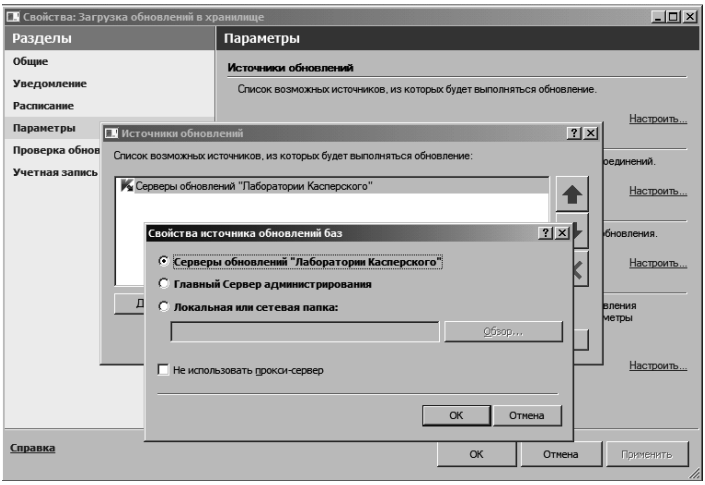


Рис. 5.72. Настройка источника обновлений

По умолчанию для одиночного сервера администрирования конфигурируется получение обновления с серверов ЛК. При иерархической организации структуры управления антивирусной защитой появятся и другие источники, например, главный сервер администрирования. Также можно настроить обновление из локальной или сетевой папки.

Задание 1.

1. Изучите параметры задачи «Загрузка обновлений в хранилище». Какие обновления загружаются, какой сетевой протокол и в каком режиме используется для загрузки?

2. Проанализируйте параметры задачи «Резервное копирование данных сервера администрирования». Как часто проводится резервное копирование, и где хранятся резервные копии?

Предустановленные задачи можно доопределить, поменяв параметры. Кроме того, их можно запустить и вручную (например, если немедленно требуется получить обновление антивирусных баз).

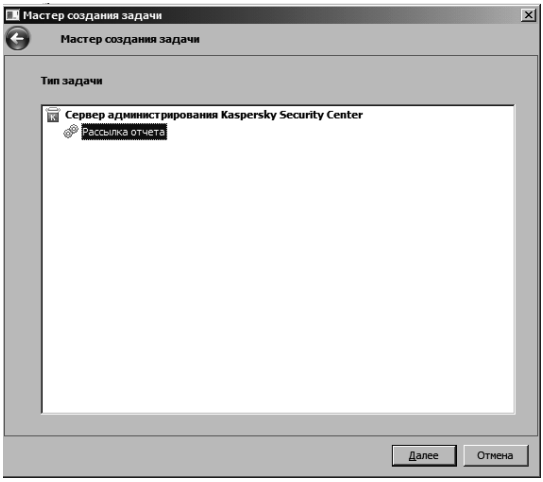


Рис. 5.73. Создание задачи
сервера администрирования

Еще один тип задач сервера администрирования — это рассылка отчетов. Если создать новую задачу сервера администрирования, только этот тип и будет доступен (рис. 5.73). Можно сконфигурировать отправку отчетов по электронной почте (надо указать сервер, через который будет производиться отправка, и адрес получателя) или сохранение отчета в указанной папке.

Задание 2.

Создайте задачу на ежедневное сохранение отчета о вирусах в папке d:\avreport. Отчет должен формироваться в 11:00 и быть в формате pdf. Проверьте работу созданной задачи.

Групповые задачи выполняются в отношении компьютеров, включенных в группу. С созданием задач для группы *Управляемые компьютеры* мы уже сталкивались в предыдущих лабораторных работах. Для этой группы есть три предустановленные задачи — *Обновление*, *Поиск вирусов* и *Поиск уязвимостей* (рис. 5.74).

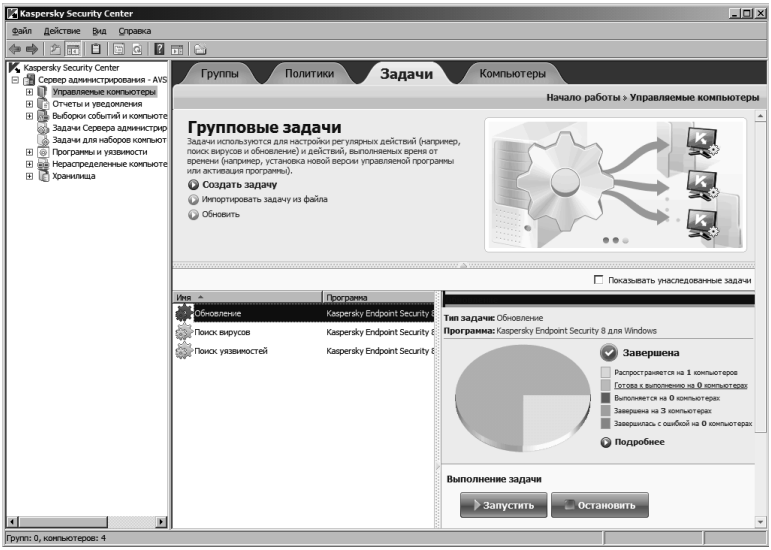


Рис. 5.74. Задачи для группы «Управляемые компьютеры»

При настройках по умолчанию обновление клиентских компьютеров производится при каждой загрузке обновлений в хранилище сервера администрирования. Источники обновления могут быть различны при работе в локальном или мобильном режиме (рис. 5.75). В локальном режиме при настройках по умолчанию обновление производится с сервера Security Center, в мобильном — с серверов Лаборатории Касперского. Мобильный режим включается, если агент администрирования на компьютере не может установить связь с сервером администрирования.

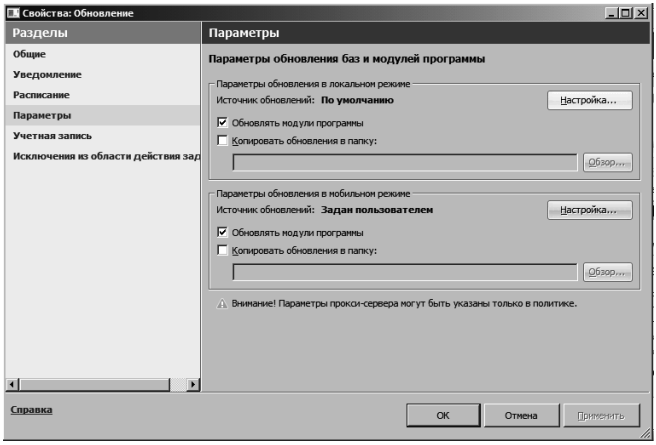


Рис. 5.75. Параметры обновления

Задание 3.

Ознакомьтесь с параметрами и расписанием стандартных задач для группы «Управляемые компьютеры». Как часто производится поиск вирусов «по расписанию»? Проверяются ли при этом файлы в архивах? Используется ли при таких проверках эвристический анализ? Как часто производится поиск уязвимостей, и ПО каких производителей проверяется?

В группе можно создавать подгруппы. Для этого нужно в окне, аналогичном представленному на рис. 5.74, перейти на вкладку *Грун-*

пы и выбрать *Создать подгруппу*. Далее можно переместить в группу относящиеся к ней компьютеры (рис. 5.76).

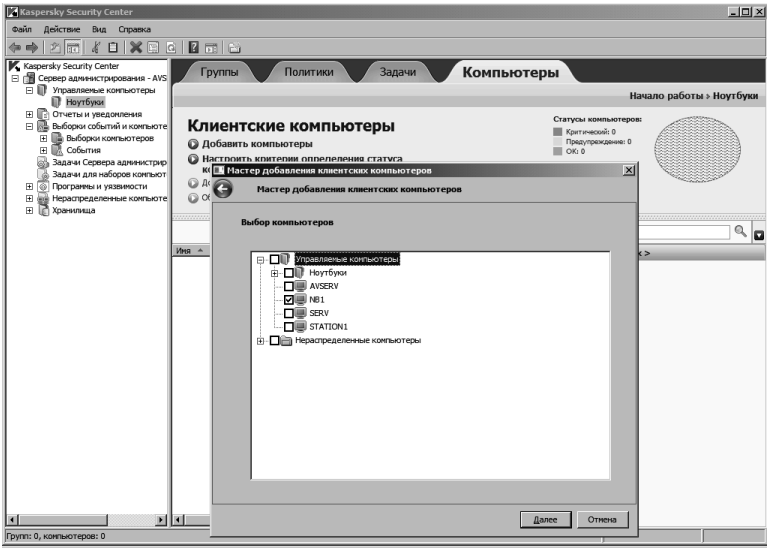


Рис. 5.76. Добавление компьютера в группу

Задание 4.

В группе «Управляемые компьютеры» создайте подгруппу «Ноутбуки». Переместите туда компьютер NB1.

Групповые задачи по умолчанию будут выполнены на всех компьютерах группы и включенных в нее подгрупп. Создайте групповую задачу для отправки сообщения (рис. 5.77) пользователям группы «Управляемые компьютеры». Также создайте аналогичную задачу для подгруппы «Ноутбуки». Запустите обе задачи. Сообщения каких задач появились на компьютере NB1? На компьютере Serv? Изучите перечень возможных задач для группы.

Редактируя созданную групповую задачу, в разделе *Исключения из области действия задачи* можно настроить подгруппы, для которых задача не будет выполняться (рис. 5.78).

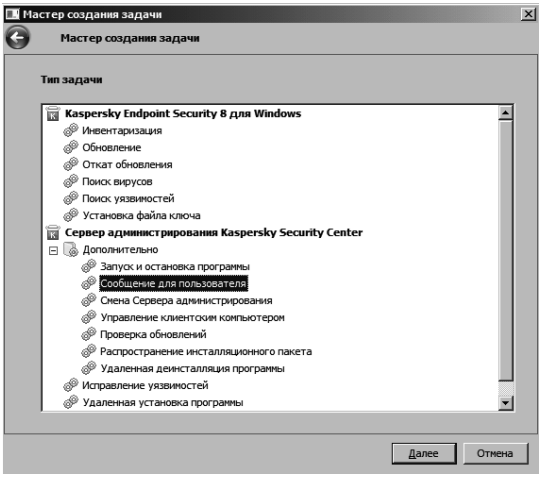


Рис. 5.77. Создание задачи на отправку сообщения пользователям

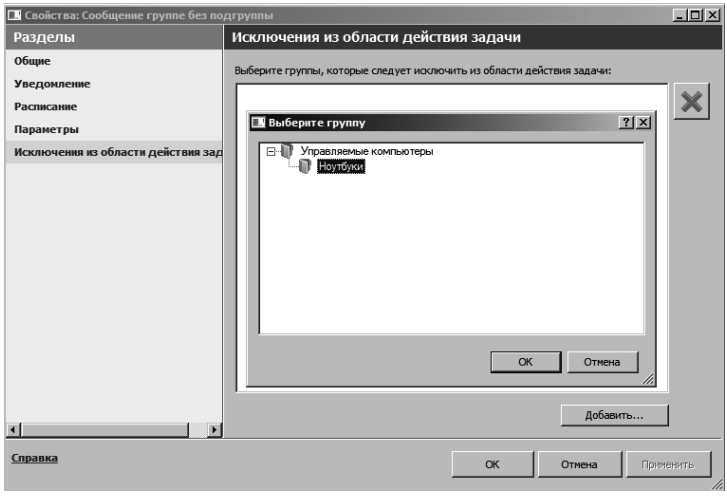


Рис. 5.78. Настройка исключений для задачи

В отличие от групповых задач, *задачи для набора компьютеров* при выполнении не учитывают принадлежность компьютера той или иной группе. Но компьютеры нужно указать каким-то другим обра-

зом. Подобного класса задачу мы уже создавали при развертывании агента администрирования. Тогда требуемый список был получен в результате поиска нераспределенных компьютеров. Когда задачу надо выполнить на клиентах, соответствующих определённому условию, можно воспользоваться «выборками» компьютеров (рис. 5.79).

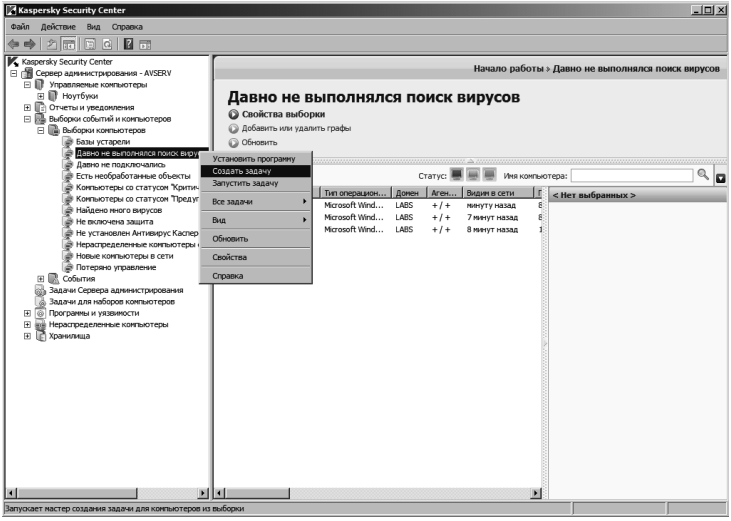


Рис. 5.79. Создание задачи для выборки компьютеров

Задание 5.

Раскройте узел *Выборки событий и компьютеров -> Выборки компьютеров* и ознакомьтесь с перечнем предустановленных выборок. Обратите внимание, что создать задачу для выборки, как это показано на рис. 5.79, можно только в случае если выборка не пуста. Создайте задачу поиска вирусов для выборки «Давно не выполнялся поиск вирусов», задача должна запускаться вручную (если указанная выборка не содержит компьютеров, используйте другую). Запустите задачу, проверьте ее выполнение.

Рассмотрим процесс создания новой выборки. Пусть требуется выбрать компьютеры с операционной системой Windows 7. Из контекстного меню узла *Выборки компьютеров* вызовите команду *Со-*

здать ->Новая выборка. Появится окно с запросом названия выборки (назовем ее «Windows 7»), и она будет создана. После этого нужно найти выборку в списке и из контекстного меню вызвать команду *Свойства*. В окне свойств нужно перейти в раздел *Условия* и там нажать кнопку *Свойства*. После этого в разделе *Программа* нужно указать требуемую версию операционной системы (рис. 5.80). Подобных условий отбора может быть несколько.

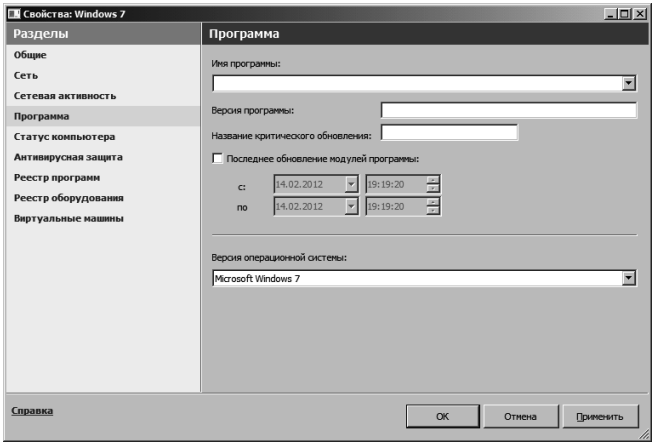


Рис. 5.80. Выбор параметров фильтрации

Задание 6.

Создайте выборку компьютеров с операционной системой Windows 7. Для этой выборки создайте и выполните задачу отправки пользователям сообщения, проверьте ее работу.

Ознакомьтесь с полным перечнем возможных условий отбора компьютеров.

Политики позволяют централизованно определить настройки для программ ЛК, как антивирусных, так и управляющих ими (имеются в виду сервер и агент администрирования). Политика определяется для группы администрирования и наследуется входящими в нее подгруппами.

Применение политики производится следующим образом: если на клиентском компьютере выполняются резидентные задачи, они продолжают свое выполнение с новыми значениями параметров, не прерываясь. Запущенные периодические задачи (проверка по требованию, обновление баз программ) продолжают выполнение с неизменными значениями. Новый запуск периодических задач производится с измененными значениями параметров.

В случае использования иерархической структуры серверов администрирования подчиненные серверы получают политики с главного сервера и распространяют их на клиентские компьютеры.

Если происходит разрыв соединения между сервером администрирования и клиентским компьютером, на клиентском компьютере вступает в силу политика для мобильного пользователя (если она определена), или политика продолжает действовать с прежними параметрами до восстановления соединения.

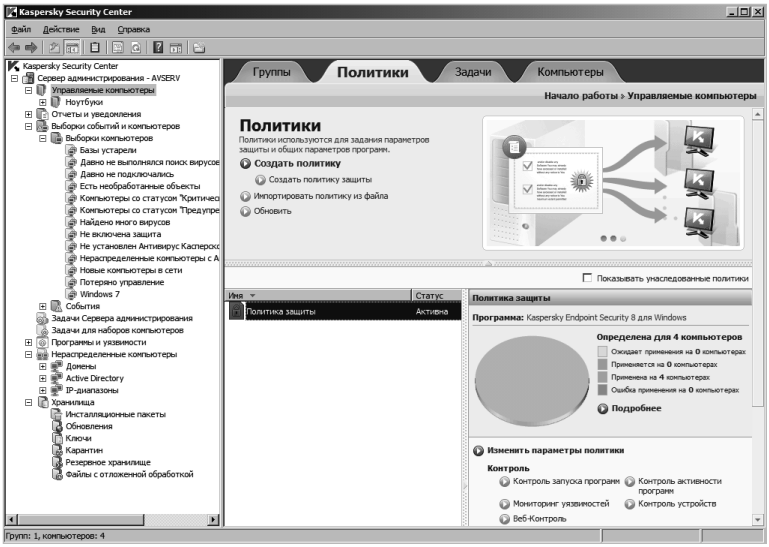


Рис. 5.81. Предустановленная политика для группы Управляемые компьютеры

Для группы *Управляемые компьютеры* существует предустановленная политика, которую можно просмотреть, выделив узел группы и открыв вкладку *Политики* (рис. 5.81). Круговая диаграмма позволяет увидеть результаты распространения политики.

Задание 7.

Перейдя по ссылке «Изменить параметры политики» (рис. 5.81), ознакомьтесь с установками политики защиты. С какими настройками работает файловый антивирус? Какие настройки сделаны для защиты клиентского компьютера в случае сетевой атаки?

При создании политики можно настроить минимальный набор параметров, без которых программа не будет работать. Остальные значения параметров устанавливаются по умолчанию и соответствуют значениям по умолчанию при локальной установке программы. После создания политики параметры, на изменение которых наложен запрет (установлен «замок»), начинают действовать на клиентских компьютерах независимо от того, какие параметры были определены для программы ранее.

Для одной программы в группе можно создать несколько политик, но активной может быть только одна из них. При создании новой активной политики предыдущая активная политика становится неактивной. Изменения можно сделать в свойствах политики в разделе *Дополнительно -> Активность и наследование* (рис. 5.82). Если на этой вкладке выбран вариант *Политика для мобильных пользователей*, политика начинает действовать на компьютерах в случае их отключения от сети организации.

Создадим новую политику, определяющую, что файловый антивирус должен использовать настройки, соответствующие *Высокому уровню безопасности* вместо *Рекомендуемого уровня*. Отличие заключается в том, что *Высокий уровень* требует от резидентного антивирусного модуля проводить более глубокую проверку файлов, проверять архивы и т. д. Плата за более качественную антивирусную

проверку — это снижение «полезной» производительности компьютера, поэтому применять ее будем в особых случаях, о чем далее.

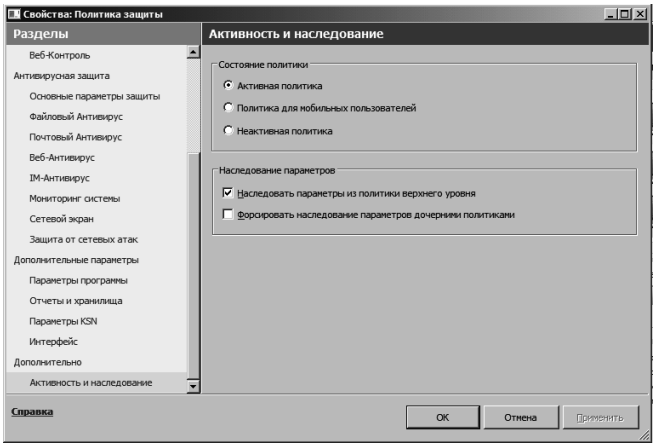


Рис. 5.82. Выбор активной политики

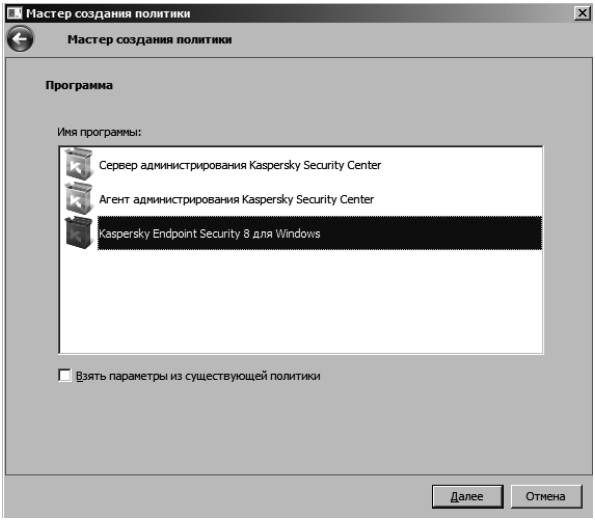


Рис. 5.83. Создание новой политики
для Kaspersky Endpoint Security

Пусть создаваемая политика называется «Сверхнадежная защита» и действует для программы Kaspersky Endpoint Security 8.0 (рис. 5.83).

В следующем окне будет запрошен конфигурационный файл для импорта параметров. Но его у нас нет, поэтому просто идем далее. В окне *Настройка параметров контроля* оставляем все по умолчанию.

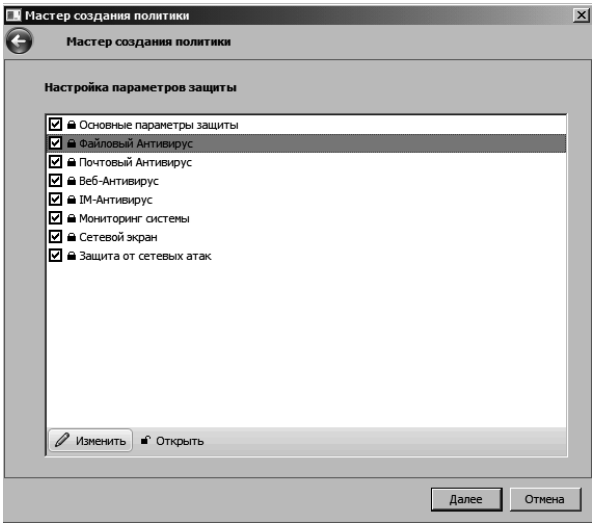


Рис. 5.84. Выбор изменяемых настроек

Изменения понадобится внести в следующем окне — *Настройка параметров защиты*. Там надо выделить группу настроек *Файловый Антивирус* и нажать кнопку *Изменить* (рис. 5.84). Уровень безопасности устанавливаем как *Высокий* (рис. 5.85).

Далее соглашаемся с настройками по умолчанию, с присоединением к сети KSN и т. д. Важно в последнем окне мастера создания политики, представленном на рис. 5.86, указать, что данная политика не является активной.

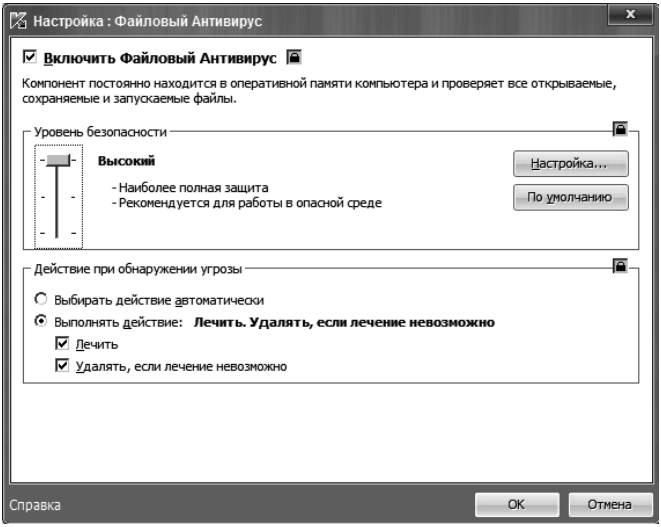


Рис. 5.85. Изменение настройки

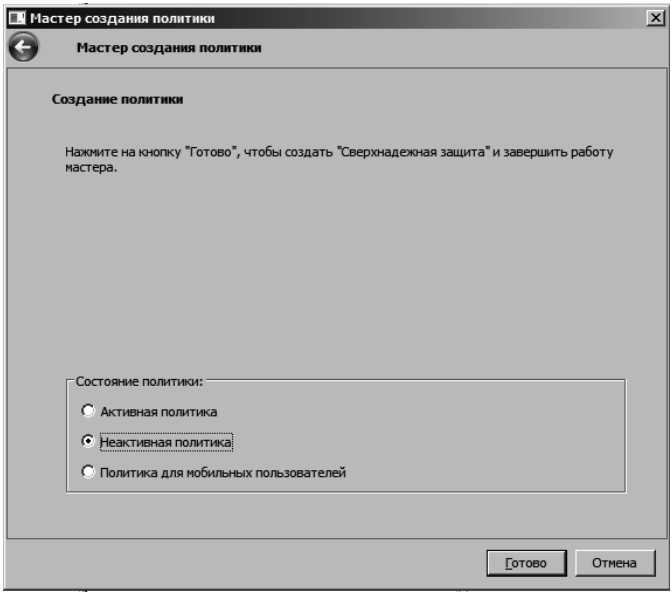


Рис. 5.86. Изменение состояния политики

Задание 8.

В соответствии с приведенным описанием создайте новую политику, но не делайте ее активной.

На сервере можно настроить специальную политику, автоматически активирующуюся при вирусной атаке. Делается это следующим образом. Из контекстного меню сервера выберем команду *Свойства* (рис. 5.87).

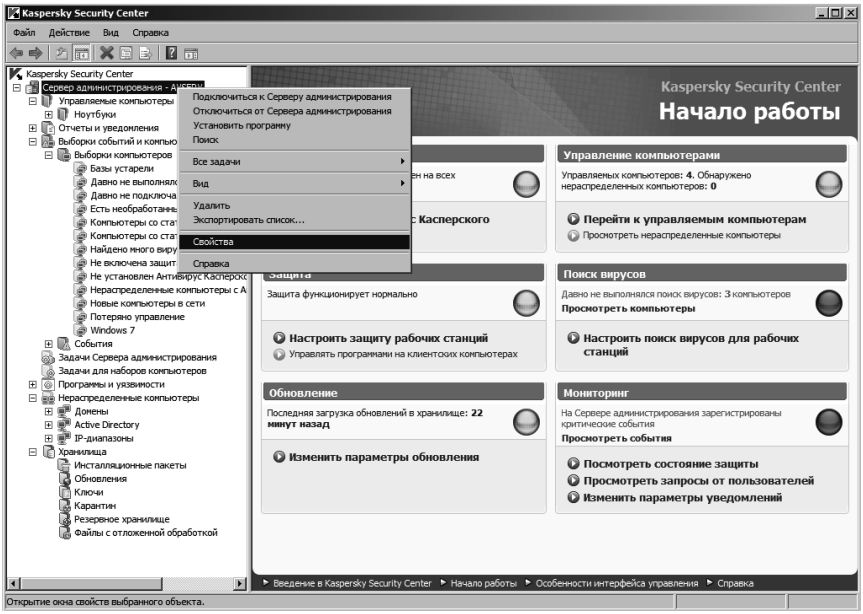


Рис. 5.87. Свойства сервера

Далее перейдем в раздел *Вирусная атака* и определим, что является критерием вирусной атаки. Пусть это будет обнаружение антивирусами для рабочих станций и файловых серверов в течение 10 минут не менее 5 вирусов (рис. 5.88). Ставим соответствующую галочку и меняем настройку по умолчанию (по умолчанию — 10 вирусов; к сожалению, из-за неудачного выбора шрифтов надпись на рисунке видна не полностью).

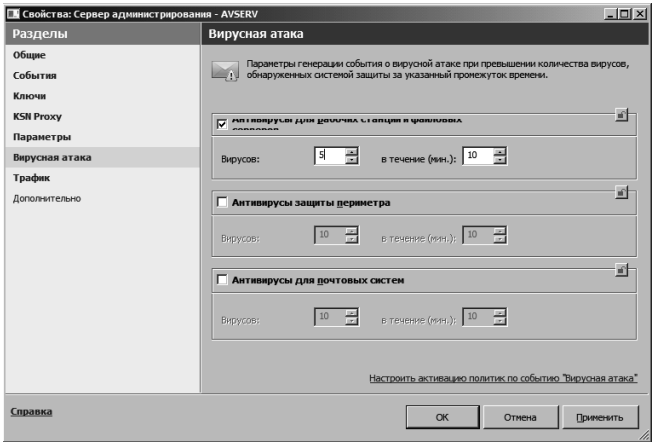


Рис. 5.88. Определяем критерий вирусной атаки

Далее переходим по ссылке *Настроить активацию политик по событию «Вирусная атака»* и добавляем подготовленную в предыдущем задании политику для антивирусов для рабочих станций и серверов.

Задание 9.

В соответствии с приведенным описанием, назначьте политику, применяемую в случае вирусной атаки.

5.10.5. Работа с вирусными инцидентами

Цель работы.

В работе рассматривается порядок действий администратора антивирусной защиты при обнаружении в сети вредоносного программного обеспечения.

Описание работы.

Для выполнения данной лабораторной работы нам понадобится тестовый файл EICAR, который доступен на сайте http://www.eicar.org/ по ссылке Download Anti Malware Test File. Его содержимое — это строка символов, приведенная ниже.

```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-
ANTIVIRUS-TEST-FILE!$H+H*
```

Если эту строку поместить, например, в редактор Notepad и сохранить как файл с расширением *.com, получим исполняемый файл для ОС MS-DOS, который при запуске выводит сообщение «EICAR-STANDARD-ANTIVIRUS-TEST-FILE!». Сигнатура этого файла для целей тестирования вносится в базы антивирусных продуктов.

При настройках по умолчанию резидентный антивирусный модуль Kaspersky Endpoint Security не проверяет содержимое архивов. Поэтому на виртуальной машине NB1 в папке d:\test_virus\ сохраним тестовый файл EICAR в заархивированном виде в формате *.zip, и этот же архив повторно заархивируем с помощью архиватора 7zip в формат *.7z.

Рассмотрим процесс настройки политики в части оповещения о выявленных вирусах. Через консоль администрирования откроем *Политику защиты* для группы *Управляемые компьютеры* (см. предыдущую лабораторную работу) и перейдем в раздел *Дополнительные параметры -> Интерфейс* (рис. 5.89).

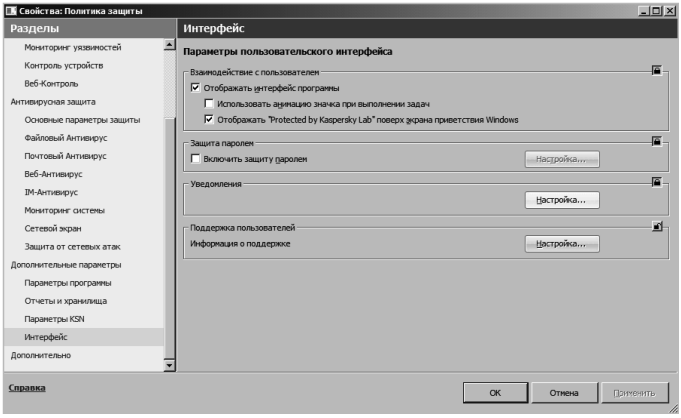


Рис. 5.89. Раздел политики, управляющий уведомлениями

В секции *Уведомления* надо нажать кнопку *Настройка*, и откроется окно настройки уведомлений. Перейдем в секцию *Файловый антивирус* (рис. 5.90).

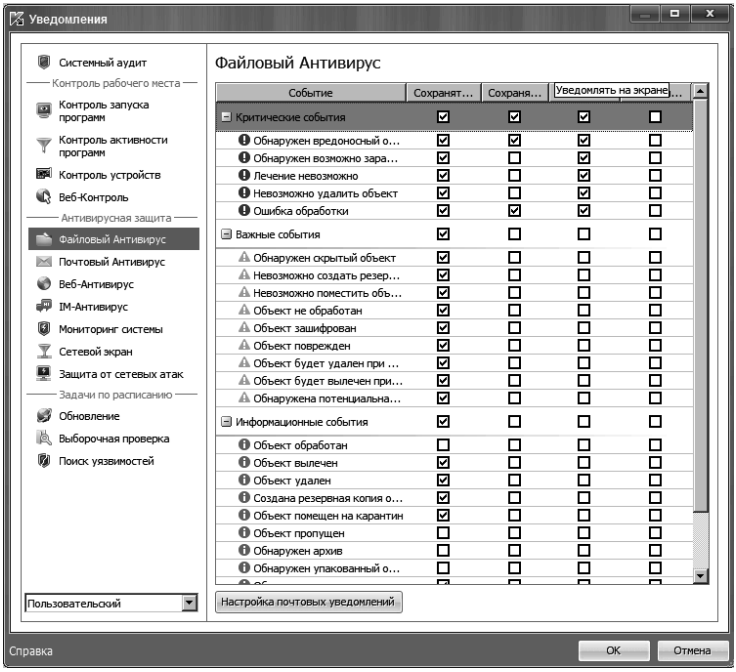


Рис. 5.90. Настройка уведомлений на экране

По умолчанию информация о событиях записывается в журнал антивирусной программы и, в некоторых случаях, в журнал Windows. Пользователь о событиях не информируется. Чтобы это поведение изменить, для группы критических событий файлового антивируса установим отметку *Уведомлять на экране*.

Задание 1.

Не меняя настроек политики, на компьютере NB1 разархивируйте файл eicar.com. Установленный на компьютере антивирус должен удалить файл. Для пользователя файл как будто просто пропадает, без всяких сообщений. После этого на сервере администрирования измените действующую политику для Endpoint Security так, как это показано на рис. 5.90. Подождите минуту–две, пока политика применится на NB1, и повторите эксперимент. Что изменилось?

Настройка уведомлений на экране нужна в первую очередь для пользователя, потому что он увидит сообщение на своем мониторе. Для администратора можно настроить отправку уведомлений по электронной почте. Здесь тоже имеет смысл ограничиться только критическими событиями, иначе почтовый ящик будет быстро забит сообщениями. В окне, представленном на рис. 5.90, отметим для группы критических событий отправку уведомлений по почте. А в нижней части окна можно нажать кнопку *Настройка почтовых уведомлений* и задать настройки для отправки сообщений (рис. 5.91).

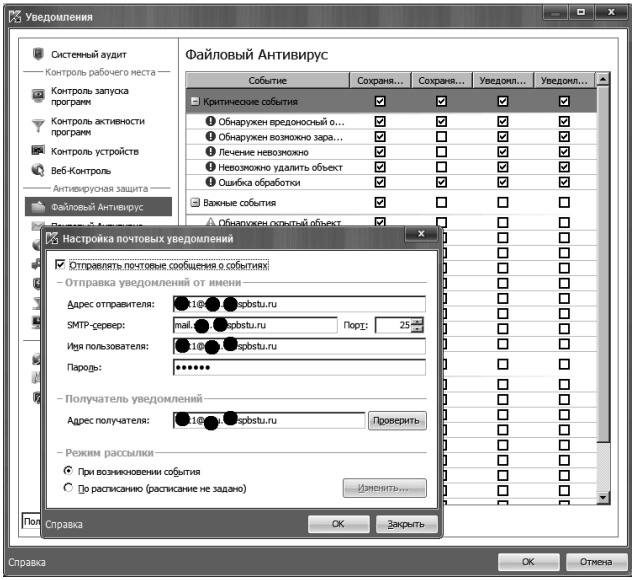


Рис. 5.91. Настройка уведомлений по электронной почте

Задание 2.

Ознакомьтесь с настройкой отправки уведомлений по электронной почте. При наличии технической возможности выполните настройку и проверьте ее работу.

Заданная нами политика будет определять поведение клиентских компьютеров при отправке уведомлений. Но если компьютер не

имеет доступа к почтовому серверу, то и уведомление на почту администратора он не отправит.

Отправка уведомлений сервером администрирования настраивается через узел *Отчеты и уведомления*, вкладка *Уведомления* (рис. 5.92).

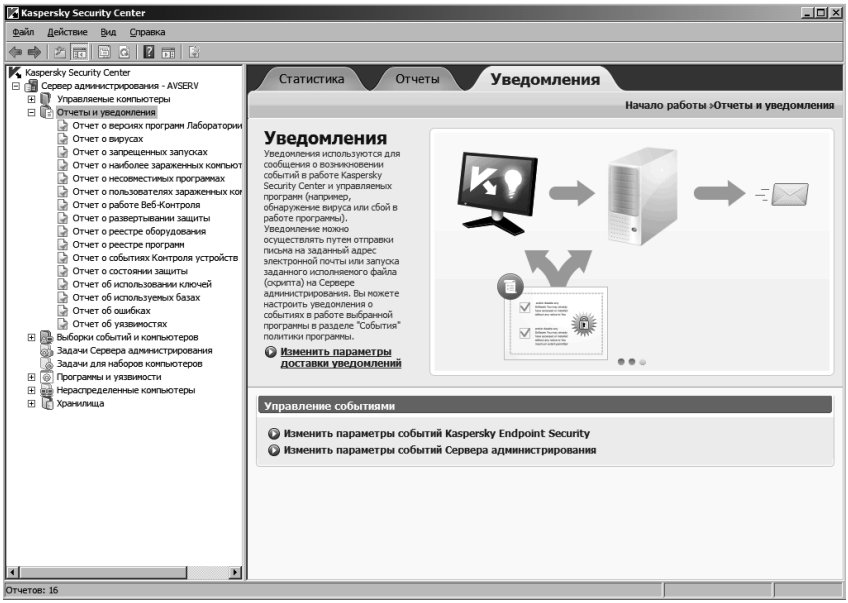


Рис. 5.92. Настройка доставки уведомлений

Если перейти по ссылке *Изменить параметры доставки уведомлений*, то отроется окно (рис. 5.93), где можно указать реквизиты электронной почты или сконфигурировать запуск исполняемого файла. Там же можно задать шаблон для автоматически формируемого текста сообщения.

Задание 3.

Настройте сервер для отправки сообщений по электронной почте. При наличии технической возможности проверьте доставку сообщений.

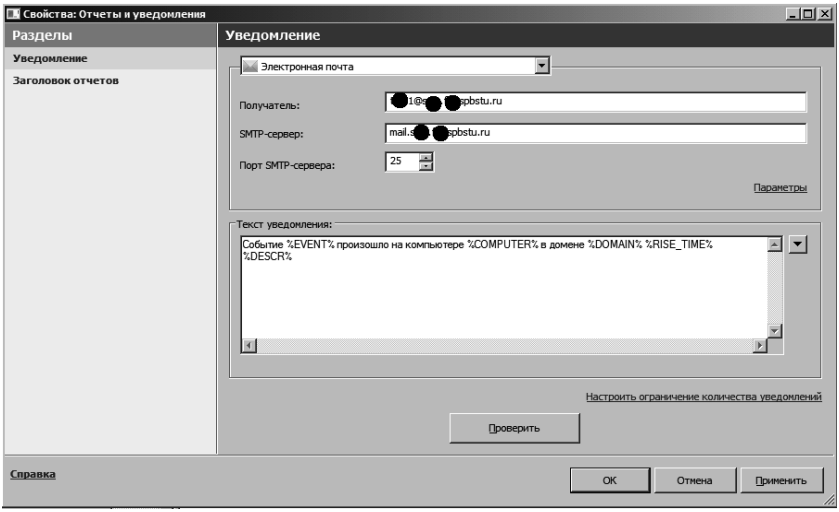


Рис. 5.93. Настройка параметров отправки

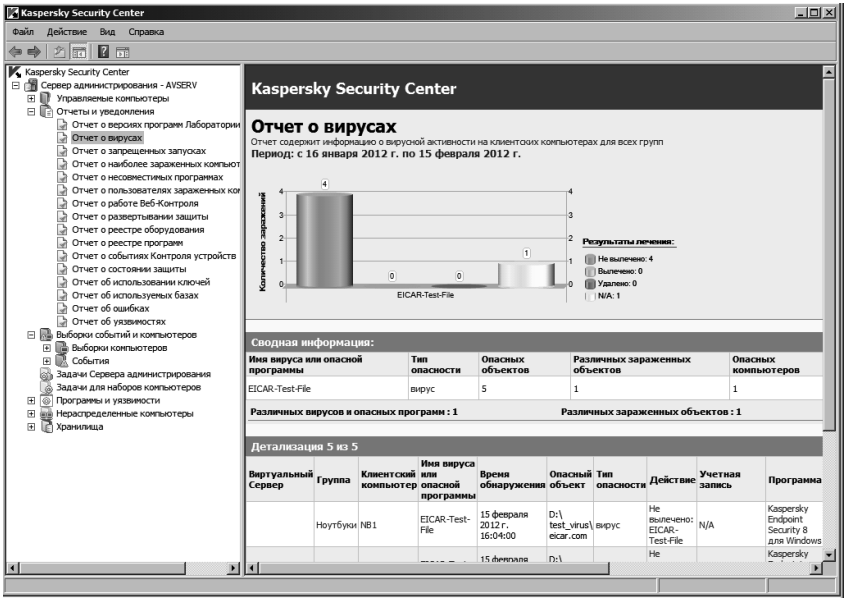


Рис. 5.94. Отчет о вирусах

С задачей, создающей по расписанию отчет об обнаруженных вирусах, мы уже работали в ходе выполнения предыдущей лабораторной работы. Быстро просмотреть отчет о текущем состоянии системы можно, перейдя на узел *Отчеты и уведомления* и раскрыв *Отчет о вирусах*, см. рис. 5.94. Из отчета можно получить информацию об обнаруженных вирусах, числе зараженных компьютеров, времени обнаружения и т. д. В частности, на рис. 5.94 видно, что на компьютере NB1 был несколько раз обнаружен тестовый файл EICAR.

Задание 4.

Сформируйте отчет о вирусах, ознакомьтесь с его содержимым.

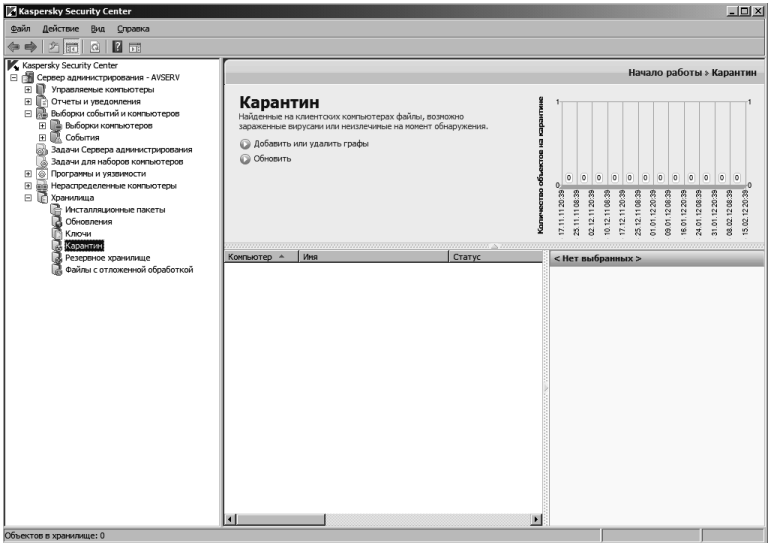


Рис. 5.95. Карантин

Как мы выяснили в ходе экспериментов, файлы, содержащие вирусы и прочее вредоносное ПО, если их нельзя сразу вылечить, удаляются с клиентского компьютера. Но эти файлы не нужно считать потерянными безвозвратно. В случае если в сети развернут Security Center, антивирус перемещает файл в одно из централизованных хранилищ. В *Карантин* (рис. 5.95) попадают файлы, которые антиви-

рус не может вылечить на момент обнаружения в них вируса. Если после очередного обновления станет известен способ излечения файла, то его можно будет восстановить и вернуть на компьютер.

В *Резервное хранилище* (рис. 5.96) помещаются копии файлов, удаляемых с клиентских компьютеров или изменяемых в процессе лечения. Например, туда попал файл eicar.com, удаленный с компьютера NB1. Если выяснилось, что действия антивируса были ошибочны, файл можно восстановить, воспользовавшись соответствующей ссылкой.

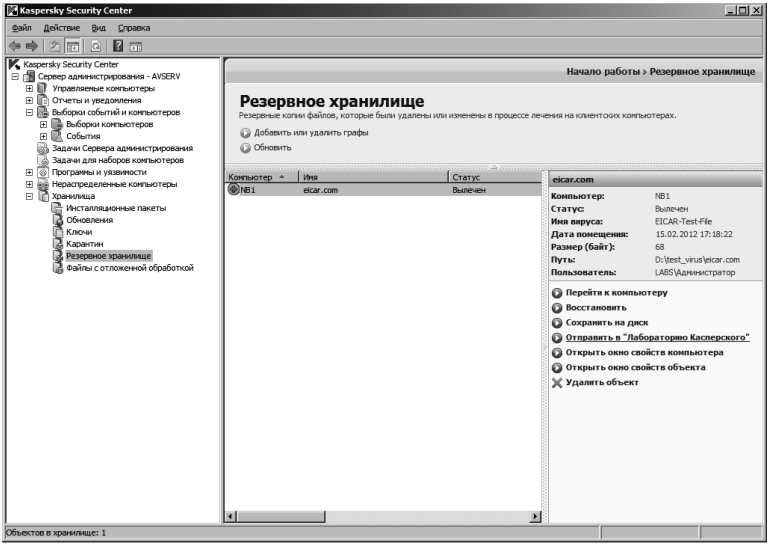


Рис. 5.96. Резервное хранилище

И последнее рассматриваемое в данной работе хранилище — *Файлы с отложенной обработкой* (рис. 5.97). Решение о действиях в отношении таких файлов должен принять администратор антивирусной защиты.

Задание 5.

Ознакомьтесь с содержимым централизованных хранилищ на вашем сервере администрирования.

Объясните, почему там оказались найденные в хранилищах файлы.

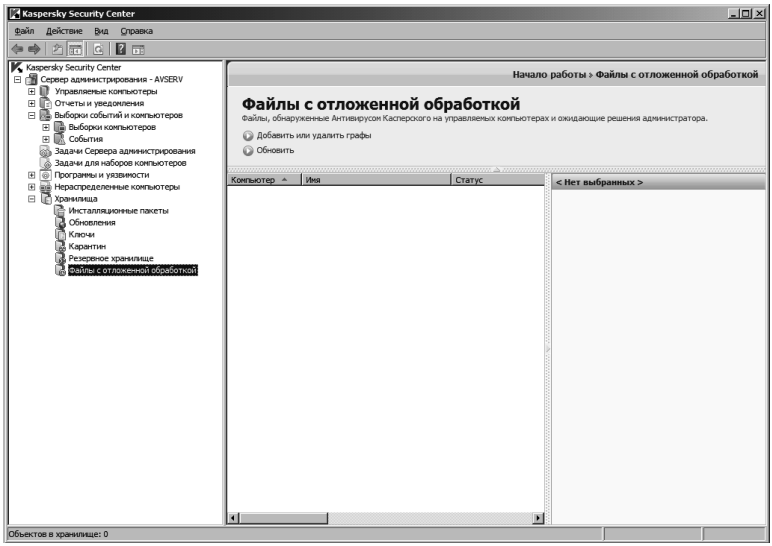


Рис. 5.97. Файлы с отложенной обработкой

При обнаружении в сети вирусного заражения можно рекомендовать сначала для получения последних обновлений вручную запустить задачу *Загрузка обновлений в хранилище* (в задачах для сервера администрирования). После этого можно воспользоваться выборкой *Компьютеры со статусом «Критический»*, чтобы определить те компьютеры, на которых обнаружены вирусы или давно не выполнялась проверка. Для отобранных компьютеров следует создать задачу *Поиск вирусов*.

Когда инцидент и его последствия устранены, для затронутых им компьютеров надо обнулить счетчик вирусов, как это показано на рис. 5.98.

Задание 6.

С помощью файла eicar.com выполните имитацию вирусной атаки. Выполните рекомендуемые при вирусном инциденте действия. В результате запуска поиска вирусов на компьютерах был ли обнаружен тестовый файл в *.zip архиве? А в *.7z? Как можно объяснить появление файла из *.7z архива в хранилище *Файлы с отложенной обработкой*?

После этого эксперимента обнулите счетчик вирусов компьютера NB1.

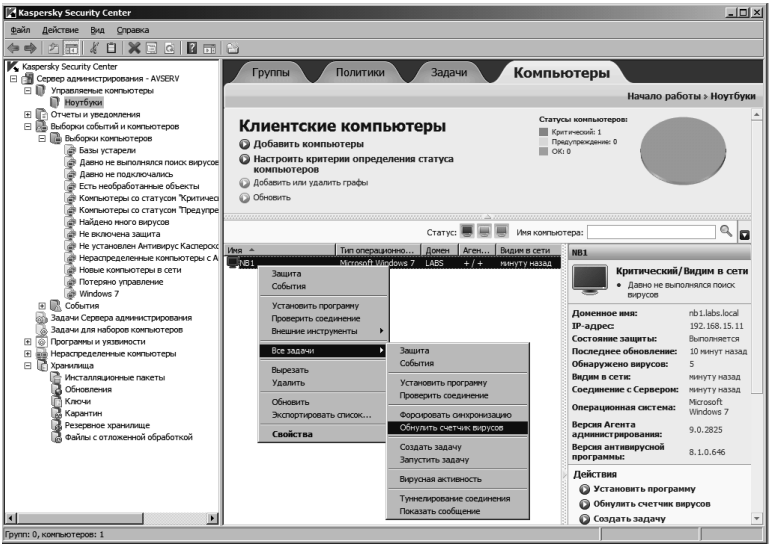


Рис. 5.98. Обнуление счетчика вирусов

Также не надо забывать политику, активирующуюся по событию «Вирусная атака», которую мы создавали в предыдущей лабораторной работе. Называлась она «Сверхнадежная защита».

Задание 7.

Инициируйте на компьютере NB1 такое число инцидентов, чтобы стала активной политика для антивируса, созданная на случай вирусной атаки. Убедитесь в этом (рис. 5.99).

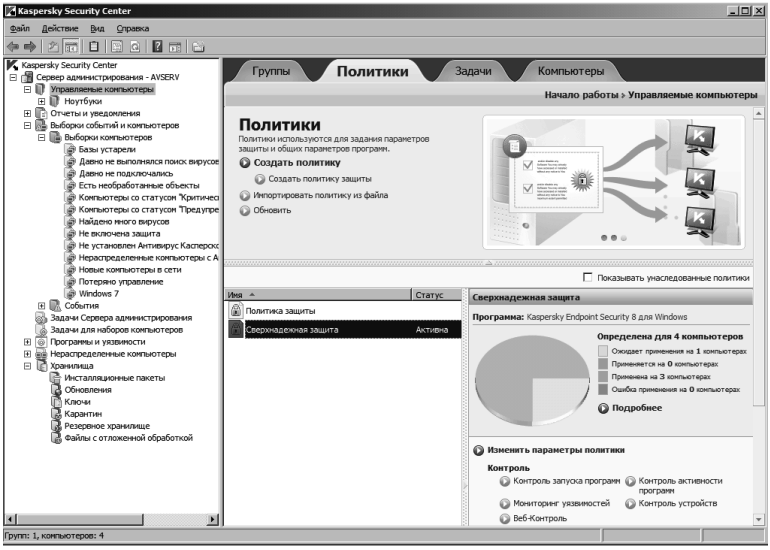


Рис. 5.99. Сработала политика на случай вирусной атаки

Стали ли проверяться открываемые *.zip архивы?

Спустя некоторое время вновь назначьте активной использующуюся по умолчанию *Политику защиты*.

5.11. НАСТРОЙКА ПРОТОКОЛА IPSEC В WINDOWS SERVER 2008

Цель работы.

В лабораторной работе рассматривается порядок настройки защищенного с помощью протокола IPSec соединения между клиентом и сервером.

Используемые программные средства.

Компьютер или виртуальная машина с ОС Windows Server 2008 и установленной и настроенной ролью Active Directory Domain Services (в описании — Server1). Второй физический компьютер или виртуальная машина с ОС Windows, включенная в домен. В описании лабораторной работы используется вторая виртуальная машина, работающая под управлением ОС Windows Vista (Vista1), но можно использовать и другие версии Windows. В домене развернут центр сертификации.

Описание работы.

Целью работы является настройка протокола IPSec для шифрования данных, передаваемых между сервером и рабочей станцией, находящимися в одном домене.

Для работы с политиками IPSec существует оснастка IPSecurity Policy Management. Если запустить консоль mmc и добавить эту оснастку, появится запрос, для какого объекта будет использоваться оснастка (рис. 5.100).

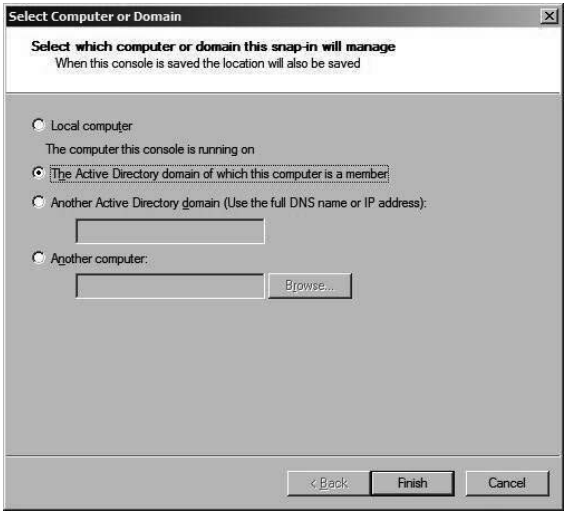


Рис. 5.100. Выбираем объект для работы

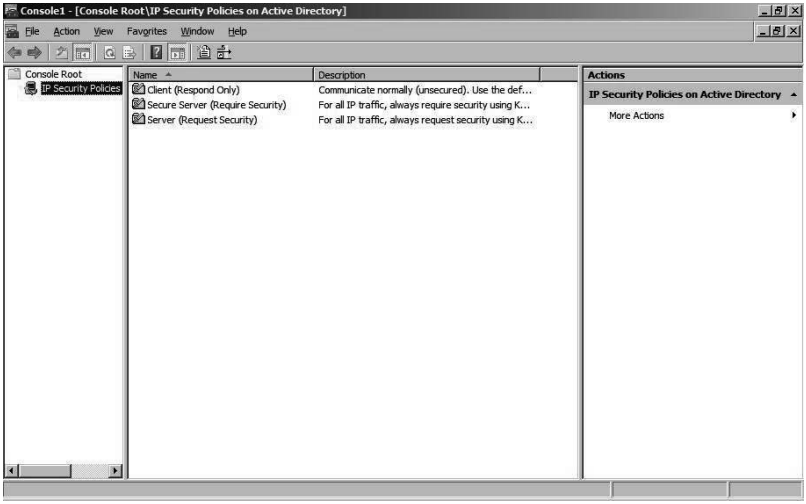


Рис. 5.101. Предопределенные политики IPSec

Настройку будем делать с помощью доменной политики. В ней уже существуют три предопределенные политики (рис. 5.101). Но нам нужна будет новая, управляющая работой конкретного сервера и клиента. Поэтому в контекстном меню выбираем пункт *Create new security policy* и по запросу мастера назначаем ей имя *Server1_Vista1*.

Рис. 5.102. Окно мастера IP Security Policy

Настройка в следующем окне понадобится в случае, если используются предыдущие (по сравнению с Windows Server 2008/ Windows Vista) версии операционных систем, его пропускаем.

Выбрав в окне (рис. 5.102) пункт *Edit Properties,* переходим непосредственно к созданию настроек.

Нам понадобится новое правило, поэтому в окне, представленном на рис. 5.103, нажимаем кнопку *Add*. В следующем окне указываем, надо ли определять туннель (рис. 5.104). Так как мы планируем использовать IPSec в транспортном режиме (см. раздел 3.3 данного пособия), это не понадобится.

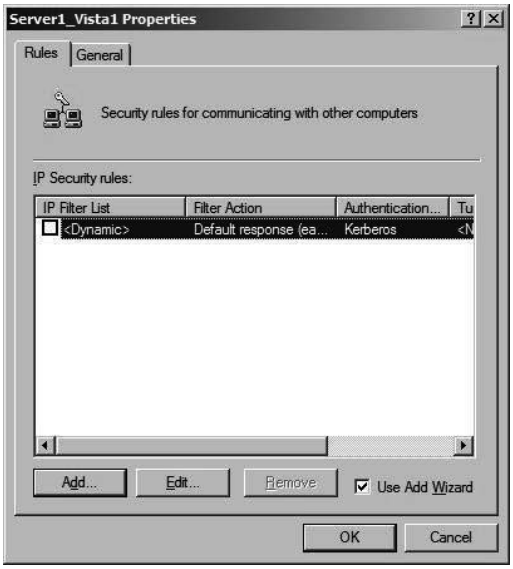


Рис. 5.103. Добавляем правило

Следующий запрос касается того, для каких подключений действует правило — для всех, подключений из локальной сети или извне. Нас устроит вариант «для всех» (*англ.* All network connections). После этого будет предложено определить, в отношении какого типа трафика действует создаваемая политика (рис. 5.105).

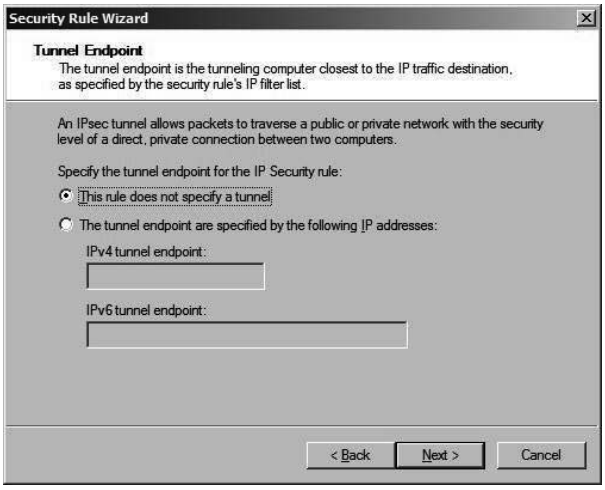


Рис. 5.104. Выбор типа соединения

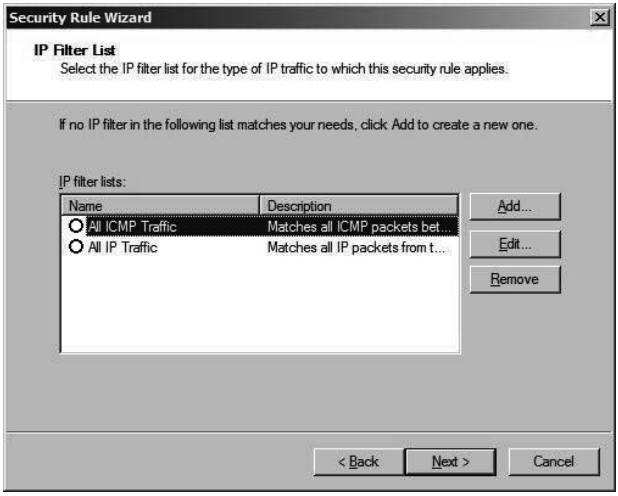


Рис. 5.105. Фильтры позволяют определить,
какие пакеты будут защищаться IPSec

Предустановленные правила нас не устраивают, так как нам нужно защищенное соединение между двумя конкретными узлами. Нажимаем кнопку *Add*, чтобы добавить новый список фильтров

(рис. 5.106). Задаем ему имя и нажимаем кнопку *Add*, что приводит к запуску очередного мастера.

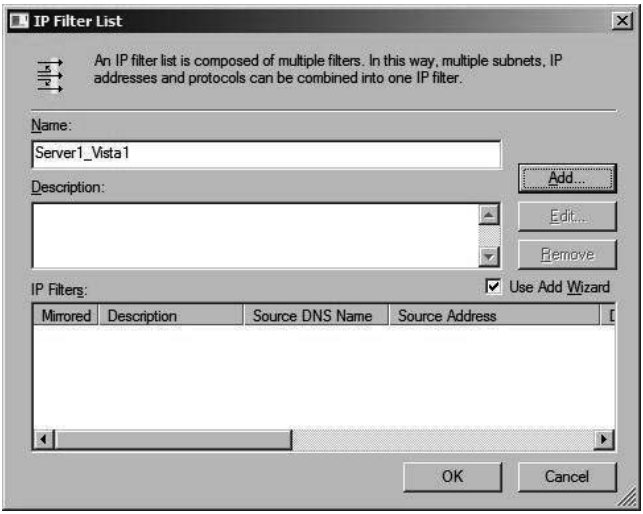


Рис. 5.106. Добавляем новый список фильтров

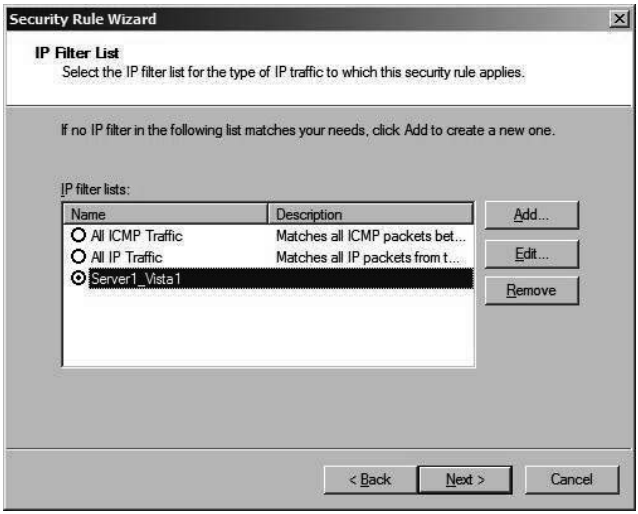


Рис. 5.107. Выбираем созданный фильтр

Работая с мастером, определим источник (*англ.* source) пакетов (в выпадающем списке выберем «A specific DNS name» и укажем имя Vista1.test.domain), получатель — Server1.test.domain. Далее можно выбрать защищаемый протокол. В нашем примере — любой (*англ.* Any). Таким образом, мы создали фильтр и теперь нужно отметить его, как использующийся (рис. 5.107). В следующем окне запрашивается действие для случая, если приходит незащищенный пакет. Его можно принять, при этом отвечая защищенной посылкой, а можно заблокировать (для этого предопределенного правила нет, нужно создать новое, нажав кнопку *Add*). Выбранный на рис. 5.108 вариант *Require Security* предполагает, что сервер может принимать незащищенные пакеты, но в ответ предлагает установку защищенного соединения.

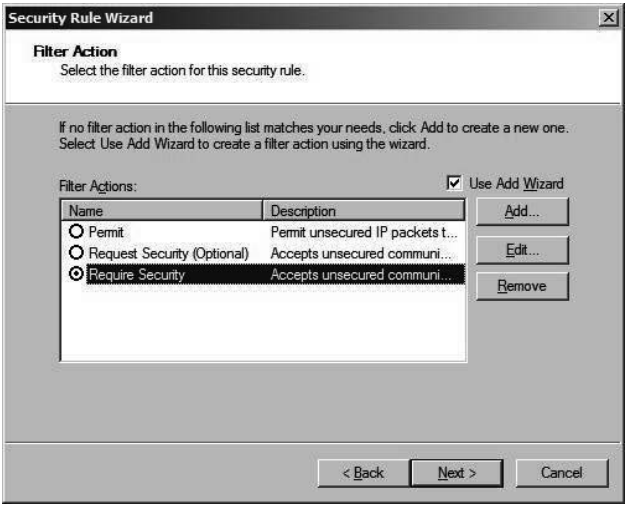


Рис. 5.108. Действия при получении незащищенного пакета

Далее предлагается выбрать метод аутентификации (рис. 5.109). Выбор делается между использованием протокола Kerberos, аутентификацией на основе цифровых сертификатов и предопределенным

ключом. Последний вариант наименее надежен. Что же касается первых двух, то если подключения производятся внутри домена, можно выбрать Kerberos. Если узел внешний, но для него нашим корпоративным центром сертификации выпущен сертификат, можно применить второй метод аутентификации.

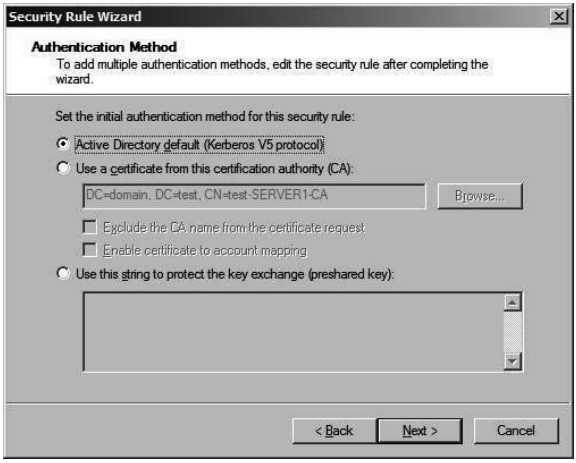


Рис. 5.109. Выбор метода аутентификации

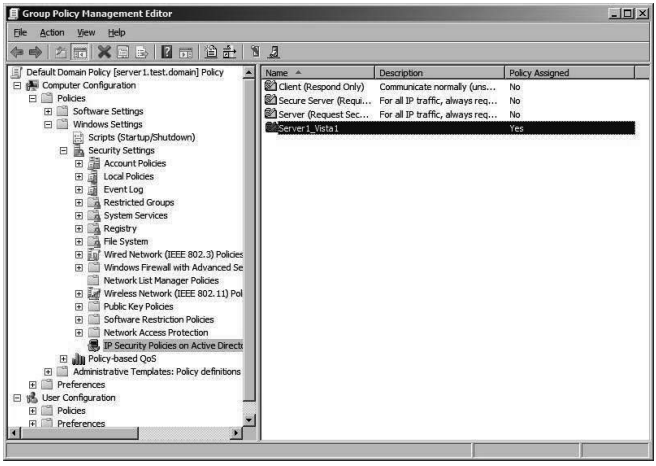


Рис. 5.110. Назначение политики

Таким образом, мы создали новую политику. Теперь ее надо назначить (*англ.* Assign). Сделать это можно в редакторе доменной политики: меню *Start-> Administrative Tools-> Group Policy Management* найти *Default Domain Policy* и в контекстном меню выбрать Edit, после чего в разделе *Computer Configuration ->Policies-> Windows Settings->Security Settings* найти политики IPSec, выбрать нужную и в контекстном меню выбрать *Assign* (рис. 5.110).

Задание.

Создайте политику IPSec. Применив политику, проверьте соединение между компьютерами.

Вполне возможно, что сразу установить соединение не получится. Проблемы могут вызвать использование межсетевых экранов (как встроенных в Windows, так и отдельных решений) и трансляция адресов (NAT), если она применяется. Возможны и другие причины.

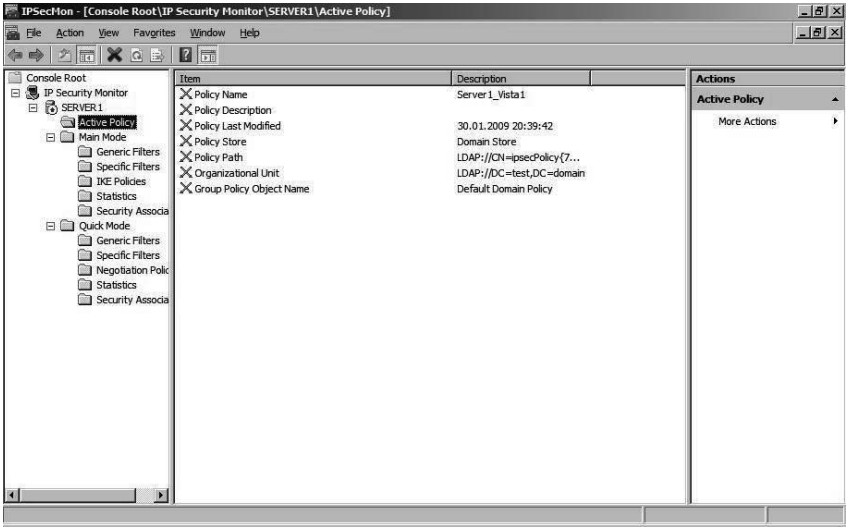


Рис. 5.111. Оснастка IPSec Monitor

При выяснении причин неправильной работы может использоваться оснастка mmc IPSec Monitor (рис. 5.111), по умолчанию она не

устанавливается, ее надо добавлять. Помощь может также оказать использование утилиты Network Monitor и анализ журналов межсетевых экранов.

БИБЛИОГРАФИЧЕСКИИ СПИСОК

1. ГОСТ Р 50922-2006. Защита информации. Основные термины и определения. – М.: Стандартинформ, 2008. – 12 с.

2. ГОСТ Р 51275-2006. Защита информации. Объект информатизации. Факторы, воздействующие на информацию. Общие положения. – М.: Стандартинформ, 2007. – 11 с.

3. *Девянин П. Н.* Теоретические основы компьютерной безопасности: Учеб. пособие для вузов / П. Н. Девянин [и др.]. – М.: Радио и связь, 2000. – 192 с.

4. ГОСТ Р ИСО/МЭК 15408-1-2002. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 1. Введение и общая модель. – М.: Госстандарт России, 2002. – 40 с.

5. *Зегжда Д. П.* Основы безопасности информационных систем / Д. П. Зегжда, А. М. Ивашко. – М.: Горячая линия – Телеком, 2000. – 452 с.

6. *Молдовян Н. А.* Проблематика и методы криптографии / Н. А. Молдовян. – СПб.: Изд-во СПбГУ, 1998. – 212 с.

7. Ященко В. В. Введение в криптографию / В. В. Ященко [и др.]; под общ. ред. В. В. Ященко. – М.: МЦНМО, «ЧеРо», 1998. – 272 с.

8. *Грунтович М. М.* Основы криптографии с открытыми ключами: учеб. пособие / М. М. Грунтович. – Пенза: Изд-во Пензен. госуд. ун-та, 2000. – 65 с.

9. *Романец Ю. В.* Защита информации в компьютерных системах и сетях / Ю. В. Романец, П. А. Тимофеев, В. Ф. Шаньгин. – М.: Радио и связь, 1999. – 328 с.

10. *Зима В. М.* Компьютерные сети и защита передаваемой информации / В. М. Зима, А. А. Молдовян, Н. А. Молдовян. – СПб.: Изд-во СПбГУ, 1998. – 328 с.

11. *Конеев И. Р.* Информационная безопасность предприятия / И. Р. Конеев, А. В. Беляев. – СПб.: БХВ-Петербург, 2003. – 752 с.

12. *Василенко О. Н.* Теоретико-числовые алгоритмы в криптографии / О. Н. Василенко. – М.: МЦНМО, 2003. – 328 с.

13. *Зима В. М.* Безопасность глобальных сетевых технологий / В. М. Зима, А. А. Молдовян, Н. А. Молдовян. – 2-е изд. – СПб.: БХВ-Петербург, 2003. – 368 с.

14. *Рудаков О. И.* Протоколы защиты информации в сети: IPSEC и SKIP / О. И. Рудаков, М. М. Грунтович // Специальная техника средств связи. Серия: Системы, сети и технические средства конфиденциальной связи. – 1999. – № 1, С. 79–85.

15. *Герасименко В. А.* Основы защиты информации / В. А. Герасименко, А. А. Малюк. – М.: Инкомбук, 1997. – 537 с.

16. Хоффман Л. Дж. Современные методы защиты информации: [пер. с англ.] / Л. Дж. Хоффман. – М.: Советское радио, 1980. – 264 с.

17. ГОСТ Р ИСО/МЭК 17799-2005. Информационная технология. Практические правила управления информационной безопасностью. – М: Стандартинформ, 2006. – 62 с.

18. ГОСТ Р ИСО/МЭК 27001-2006. Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования. – М: Стандартинформ, 2006. – 31 с.

19. *Петренко С. А.* Анализ рисков в области защиты информации. Информационно-методическое пособие по курсу повышения квалификации «Управление информационными рисками» / А. Петренко. – СПб: Издательский Дом «Афина», 2009 – 153 с.

20. Руководство по управлению рисками в области безопасности. // Microsoft Technet. URL: http://technet.microsoft.com/ru-ru/library/cc163143.aspx (дата обращения: 01.11.2010).

21. *Симонов С.* Современные технологии анализа рисков в информационных системах / С. Симонов // PC Week («Компьютерная неделя»). – 2001. – № 7. URL: http://www.pcweek.ru/themes/detail.php? ID=59394 (дата обращения: 01.11.2010).

22. *Симонов С.* Анализ рисков, управление рисками / С. Симонов // JetInfo Online. Информационный бюллетень. Архив выпусков. – 1999. – № 1. URL: http://www.jetinfo.ru/1999 (дата обращения: 01.11.2010).

23. *Симонов С.* Технологии и инструментарий для управления рисками / С. Симонов // JetInfo Online. Информационный бюллетень. Архив

выпусков. – 2003. – № 2. URL: http://www.jetinfo.ru/2003 (дата обращения: 01.11.2010).

24. The logic behind CRAMM's assessment of measures of risk and determination of appropriate countermeasures. URL: http://www.cramm.com/downloads/techpapers.htm (дата обращения: 01.11.2010).

25. *Peltier T. R.* Information Security Risk Analysis / T. R. Peltier. – Second Edition. – Auerbach Publications, 2005. – 360 с.

26. *Alberts C.* OCTAVE threat profiles / C. Alberts, A. Dorofee. URL: http://www.cert.org/archive/pdf/OCTAVEthreatProfiles.pdf (дата обращения: 01.11.2010).

27. *Storms A.* Using vulnerability assessment tools to develop an OCTAVE Risk Profile / A. Storms // SANS Institute. Part of Information security reading room. URL: http://www.sans.org/reading_room/whitepapers/auditing/vulnerability-assessment-tools-develop-octave-risk-profile_1353 (дата обращения: 01.11.2010).

28. Risk Assessment – A Management Tool for the IT Security Infrastructure. URL: http://www.riskwatch.com/white_papers.html (дата обращения: 01.11.2010).

29. *Александрович Г. Я.* Автоматизация оценки информационных рисков компании / Г. Я. Александрович, С. А. Нестеров, С. А. Петренко // Защита информации. Конфидент. – 2003. – № 2. – С. 78–81.

30. *Hamilton C.* Risk management and security / C. Hamilton. URL: http://www.riskwatch.com/PDF/white_papers/RISKMANAGEMENTANDSECURITY.pdf (дата обращения: 01.11.2010).

31. Дюбин Г. Н. Введение в прикладную теорию игр / Г. Н. Дюбин, В. Г. Суздаль. – М.: Наука, 1981. – 336 с.

Сергей Александрович НЕСТЕРОВ
ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ
Учебное пособие
Издание третье, стереотипное

Зав. редакцией естественнонаучной
литературы *М. В. Рудкевич*

Оригинал-макет произведения предоставлен издательством
Санкт-Петербургского политехнического университета Петра Великого

ЛР № 065466 от 21.10.97
Гигиенический сертификат 78.01.10.953.П.1028
от 14.04.2016 г., выдан ЦГСЭН в СПб

Издательство «ЛАНЬ»
lan@lanbook.ru; www.lanbook.com
196105, Санкт-Петербург, пр. Ю. Гагарина, д. 1, лит. А.
Тел./факс: (812) 336-25-09, 412-92-72.
Бесплатный звонок по России: 8-800-700-40-71

Подписано в печать 01.11.16.
Бумага офсетная. Гарнитура Школьная. Формат 60×90 $^{1}/_{16}$.
Печать офсетная. Усл. п. л. 20,25. Тираж 100 экз.

Заказ № 332-16.

Отпечатано в полном соответствии
с качеством предоставленного оригинал-макета
в ПАО «Т8 Издательские Технологии».
109316, г. Москва, Волгоградский пр., д. 42, к. 5.