

Искандер Конеев, Андрей Беляев

Информационная безопасность предприятия

Санкт-Петербург

«БХВ-Петербург»

2003

УДК 681.3.06
ББК 32.973
К64

Конеев И. Р., Беляев А. В.

К64 Информационная безопасность предприятия. — СПб.: БХВ-Петербург, 2003. — 752 с.: ил.

ISBN 5-94157-280-8

В книге рассматривается технология информационной безопасности, в необходимом объеме приведены сведения для организации службы информационной безопасности в типичном учреждении, начиная с малого предприятия (10 компьютеров) и до большой многофилиальной корпорации (более 1000 компьютеров). Подробно изложены вопросы управления рисками, данные современных исследований в криптографии. Большое внимание уделено практическому и прикладному применению информационной безопасности. При создании книги использовались материалы промышленных стандартов, лучшие российские и западные публикации, личный опыт авторов в рассматриваемой области.

Для специалистов в области защиты информации

УДК 681.3.06
ББК 32.973

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Полина Столбова</i>
Компьютерная верстка	<i>Татьяны Олоновой</i>
Корректор	<i>Вера Александрова</i>
Дизайн обложки	<i>Игоря Цырульников</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 20.04.03.

Формат 70×100¹/₁₆. Печать офсетная. Усл. печ. л. 60,63.

Тираж 3000 экз. Заказ № 872

"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02 от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в Академической типографии "Наука" РАН
199034, Санкт-Петербург, 9 линия, 12.

ISBN 5-94157-280-8

© Конеев И. Р., Беляев А. В., 2003
© Оформление, издательство "БХВ-Петербург", 2003

Содержание

Предисловие	1
Хочу ознакомиться с основами и базовым уровнем	1
Хочу детально разобраться в данном вопросе	2
Не планирую детально изучать тему: мне необходимо получить знания для контроля работы службы информационной безопасности	2
Благодарности	3
Глоссарий	6
ЧАСТЬ I. СЛУЖБА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ.	21
Глава 1. Общие понятия информационной безопасности	23
Определение и цели информационной безопасности	23
Механизмы информационной безопасности	25
Инструментарий информационной безопасности	26
Основные направления информационной безопасности	27
Глава 2. Терминология	29
Глава 3. Задачи и принципы организации службы информационной безопасности	35
Задачи службы информационной безопасности	35
Чем должна заниматься служба безопасности?	36
Как происходит взаимодействие со службой информационных технологий, а конкретно с администраторами сетей и систем?	37
Сколько ступеней принятия решения по вопросам информационной безопасности должно существовать?	37
Сфера охвата информационных систем	38
Сфера охвата персонала	39

Сфера охвата проектов	40
Оценка эффективности службы информационной безопасности	41
Контроль функционирования службы	41
Восемь шагов к эффективному развитию и использованию средств измерения производительности в информационных технологиях (ИТ)	42
Шаг 1. Привязать ИТ-проекты к целям и задачам предприятия	42
Шаг 2. Разработать средство измерения производительности	43
Шаг 3. Установить основу для сравнения с изменением производительности в будущем	43
Шаг 4. Выбрать наиболее значимые ИТ-проекты	43
Шаг 5. Собирать данные	43
Шаг 6. Анализировать результаты	43
Шаг 7. Объединяться с процессом управления	44
Шаг 8. Сообщать о результатах	44
Глава 4. Создание службы безопасности	45
Критерии необходимости создания службы	45
Состав службы информационной безопасности	46
Уровень подготовки специалистов	48
Подбор кадров: теория или практика?	52
Контроль процесса подбора кадров	53
ЧАСТЬ II. ЧТО ЗАЩИЩАТЬ	55
Глава 5. Инвентаризация информационных систем	57
Общий характер инвентаризации информационных систем	57
С чего начать: нормативные документы и круг респондентов	59
Принципы и направления	60
Примерный объем собираемой информации	61
Контроль инвентаризации	63
Место инвентаризации в жизненном цикле предприятия	66
Глава 6. Классификация информационных систем	67
Введение в классификацию ИС	67
Терминология и постановка задачи	67
Основные регламенты классификации	69
Классификация информационных объектов	70
Типовые сферы применения	70
Примерная модель классификации	71
Классификация средств обработки информации: стандарт CCITSE	72
Краткий обзор стандарта	72
Требования к функциональности безопасности	74
Требования к достоверности безопасности	80

Пример требований к классу: класс FCS — поддержка криптографии.....	84
Контроль классификации	87
Контроль классификации информации.....	87
Контроль классификации средств работы с информацией.....	88
Проверка соответствия информации средствам работы с ней.....	88
Глава 7. Роли и ответственность субъектов.....	90
Субъекты информационного пространства.....	90
Принципы распределения прав и ответственности	91
Упрощенная модель классификации субъектов	92
Полная модель классификации субъектов.....	93
Сопоставление ролей функциональным обязанностям.....	98
Сопоставление ролей классам обрабатываемой информации	99
Результаты классификации информационных систем	104
Глава 8. Модель информационных потоков.....	105
Направления защиты процессов обмена информацией	105
Защита сетевых процессов обмена данными	107
Защита экспорта файлов	108
Вопросы построения полной и функциональной схемы информационных потоков.....	109
Средства создания схем информационных потоков.....	110
Контроль построения модели информационных потоков.....	110
ЧАСТЬ III. ОТ ЧЕГО ЗАЩИЩАТЬ.....	113
Глава 9. Типовая модель нападения.....	115
Классификация атак	115
Немного о хакерах.....	115
Типовая атака на систему.....	118
Дополнительные возможности атак "изнутри"	123
Глава 10. Локальные атаки	124
Социальная инженерия	124
Закладки в аппаратном обеспечении.....	126
Преодоление ограничений доступа на уровне firmware	128
Получение доступа на этапе загрузки ОС.....	129
Атаки на средства аутентификации	130
Биометрические средства аутентификации	131
Токены	131
Пароли.....	132
Атаки класса "повышение привилегий"	136
Постороннее программное обеспечение	137

Пример комплексной локальной атаки.....	139
Утилиты локальных атак	142
Предварительные выводы.....	143
Глава 11. Удаленные атаки	145
Классификация удаленных атак	145
По сценарию	145
По цели	146
По характеру взаимодействия с жертвой	147
Удаленный сбор информации о жертве	148
Выяснение адресов	148
Поиск уязвимостей	149
Наиболее распространенные классы удаленных атак.....	153
Вирусы и троянские программы.....	153
Отказ в обслуживании.....	160
Маскировка.....	166
Атаки на маршрутизацию	167
Атаки на серверы: CGI и HTTP.....	168
Атаки на клиентов: ActiveX, Java и другие	170
Переполнение буфера	171
Глава 12. Атаки на поток данных	173
Пассивные атаки	173
Прослушивание сетей	173
Прослушивание в коммутируемых сетях Ethernet	175
Активные атаки	177
Атака повтором	177
Атака "злоумышленник-посредник"	179
Атаки на основе сетевой маршрутизации	181
Перехват сессии	183
ЧАСТЬ IV. АДЕКВАТНОСТЬ МЕХАНИЗМОВ ЗАЩИТЫ.	187
Глава 13. Управление рисками	189
Что такое "управление рисками"	189
Методика управления рисками в целом	192
Глава 14. Методики оценки рисков.....	194
Модель качественной оценки	194
Количественная модель рисков.....	195
Общая методика	195
Определение вероятности события	196
Определение стоимости активов.....	197
Модель обобщенного стоимостного результата Миоры (GCC)	199

Глава 15. Высоко вероятные атаки.....	200
Рейтинги атак	200
10 наиболее уязвимых продуктов и протоколов по рейтингу SANS.....	200
Рейтинг SANS-20.....	202
Часть 1: методологические ошибки	202
Часть 2: Windows-системы	203
Часть 3: UNIX-системы	205
Использование списков уязвимостей в управлении рисками	205
ЧАСТЬ V. КРИПТОГРАФИЯ.....	209
Глава 16. Основные принципы криптографии	211
Немного истории.....	211
Классификация шифров.....	218
Классификация по ключевой информации.....	218
Симметричное / асимметричное шифрование.....	219
Поточное / блочное шифрование	220
Практическая стойкость шифров.....	221
Глава 17. Симметричная криптография	224
Поточные шифры.....	224
Общие сведения	224
Стойкость шифров гаммирования	227
Линейные регистры сдвига.....	228
Стойкость линейных регистров сдвига	231
Нелинейные поточные шифры	231
Основы блочного шифрования.....	234
Основные идеи.....	234
Обратимые операции в блочном шифровании	237
Необратимые операции в блочном шифровании	239
Сети	241
Практика блочного шифрования	246
Аппаратное шифрование: DES	246
Эпоха пост-DES	251
Конкурс AES	261
Основы криптоанализа симметричных шифров	292
Криптоатаки непосредственно на алгоритмы	293
Криптоатаки на реализации алгоритмов.....	296
Глава 18. Асимметричная криптография и цифровая подпись.....	299
Основные принципы	299
Практические схемы асимметричного шифрования.....	303
Схема RSA: структура.....	303

Схема RSA: анализ.....	305
Схема RSA: пример	307
Схема Рабина.....	307
Схема Эль Гамаль: структура.....	309
Схема Эль Гамаль: анализ	310
Схема Эль Гамаль: пример	311
Схемы на основе эллиптических кривых	311
Схема Мак-Элиса	312
Электронная цифровая подпись.....	314
Схема RSA	314
Схема Рабина.....	316
Схема Эль Гамаль	317
ГОСТ Р 34.10–2001	319
Глава 19. Прочие криптографические алгоритмы	322
Хэш-функции	322
Свойства и назначение хэш-функций.....	322
Основы построения хэш-функций	324
Хэш-функции на основе блочных шифров	326
Самостоятельные хэш-алгоритмы.....	327
Генерация случайных чисел.....	330
Генерация псевдослучайных чисел	333
Глава 20. Сложные криптографические протоколы.....	337
Криптосистемы.....	337
Общие сведения	337
Общий вид симметричной криптосистемы	338
Процедура создания ключа.....	338
Мастер-ключ.....	339
Ключ сеанса.....	341
Алгоритмы создания цепочек.....	342
Асимметричные криптосистемы	346
Прочие компоненты криптосистем	346
Системы управления ключами	351
Функции и классификация	351
Системы управления симметричными ключами	351
Системы управления асимметричными ключами.....	356
Депонирование ключей.....	358
Разделение ключей	359
Аутентификация пользователей.....	361
Задачи удаленной аутентификации	361
Однопакетная парольная аутентификация	362
Аутентификация "запрос-ответ"	363
Перспективы криптографии	364

ЧАСТЬ VI. ЧЕМ ЗАЩИЩАТЬ: ОБЩЕТЕХНИЧЕСКИЕ СРЕДСТВА.....365**Глава 21. Контроль физического доступа 367**

Проблема контроля физического доступа.....	367
Механизмы контроля физического доступа	368
Общие сведения	368
Система охраны периметра	369
Система управления доступом	370
Система видеонаблюдения	372
Система охранной (пожарной) сигнализации.....	373
Система хранения	374
Интеграция систем контроля	374
Прочие аспекты	375

Глава 22. Протоколы сетевой безопасности 377

Задачи протоколов сетевой безопасности	377
Техническое описание протоколов	379
Point-to-Point Protocol	380
Password Authentication Protocol	380
Challenge Handshake Authentication Protocol	382
Extensible Authentication Protocol.....	384
Система одноразовых паролей S/Key.....	387
Terminal Access Controller Access Control System.....	389
Remote Authentication Dial-In User Service	394
Kerberos	399
Secure European System for Applications in a Multivendor Environment.....	407
Secure-HTTP	410
Secure Socket Layer.....	412
Secure Shell.....	417
Протокол SOCKS, версия 5.....	422
Протокол IP-Security	427
Simple Key Management for Internet Protocol	435
Layer 2 Forwarding, Point-to-Point Tunneling Protocol, Layer 2 Tunneling Protocol	436
Инфраструктура открытых ключей.....	442
Цифровые сертификаты X.509 v3	446
Privacy Enhanced Mail	450
Secure/Multipurpose Internet Mail Extensions.....	453
Secure Electronic Transaction	455
Аудит использования сетевых протоколов безопасности	462

Глава 23. Автоматизированные средства безопасности.....464

Антивирусы	464
Межсетевой экран (брандмауэр, firewall)	466
Классификация МСЭ.....	466

Распределенные или персональные межсетевые экраны	470
Демилитаризованная зона	471
Транслятор сетевых адресов (NAT), транслятор портов.....	473
Правила доступа, листы доступа.....	474
Альтернативный доступ	474
Уплотнение операционной системы	475
Бастион хост	475
Приобретение МСЭ.....	475
Технология VPN	477
Технология VLAN	478
Криптографические средства.....	479
Сканеры уязвимости	481
Системы обнаружения атак.....	483
Технология сравнения с образцами.....	484
Технология соответствия состояния.....	484
Анализ с расшифровкой протокола.....	485
Статистический анализ	486
Анализ на основе аномалий	486
Топология систем обнаружения атак	487
Автоматизированные средства управления политикой безопасности	491
Дополнительные автоматизированные средства	492
Инструменты злоумышленников	493
 Глава 24. Выбор конкретных автоматизированных средств безопасности.....	494
Нормативные акты и аналитические материалы.....	494
Межсетевые экраны. Показатели защищенности от несанкционированного доступа к информации	494
Руководства для приобретающих межсетевой экран, приобретающих систему обнаружения атак	498
Практика технологий систем обнаружения атак	500
Системы обнаружения атак: групповой тест.....	501
Примеры технических заданий	502
Выдержка из технического задания на межсетевой экран	502
Выдержка из технического задания на систему обнаружения атак.....	504
 Глава 25. Разработка приложений	507
Защита интеллектуальной собственности программного обеспечения	507
Проектирование механизмов безопасности	508
Механизмы информационной безопасности в программном обеспечении	508
Абстрактные модели доступа.....	509
Типовое техническое задание по безопасности программного обеспечения	513
Контроль жизненного цикла программного обеспечения	522

Глава 26. Работа с персоналом и оборудованием.....	526
Работа с персоналом	526
Общие вопросы.....	526
Прием на работу	527
Методическая работа с персоналом	529
Бывший кадровый состав предприятия	531
Работа с оборудованием	532

ЧАСТЬ VII. ОБЕСПЕЧЕНИЕ НОРМАТИВНО-ПРАВОВОГО ПРОСТРАНСТВА 537

Глава 27. Создание политики безопасности.....	539
Концепция	540
Структура документа	540
Примеры: выдержки из концепции предприятия	541
Стандарты.....	543
Структура документа	543
Примеры: выдержки из стандартов предприятия	544
Процедуры.....	548
Структура документа	548
Примеры: выдержки из процедур предприятия.....	549
Методы	554
Структура документа	554
Примеры: выдержки из методов предприятия.....	555
Аварийный план.....	557
Политики по отдельным направлениям	558
Глава 28. Работа с пользователями, администраторами и разработчиками 559	
Рекомендации по взаимодействию с персоналом.....	560
Работа с пользователями.....	560
Работа с администраторами.....	560
Работа с разработчиками.....	562
Вопросы подчинения и взаимодействия служб.....	562
Выдержки из типового регламента заявок в службу информационных технологий	564
Создание заявки на выполнение работ.....	564
Право подписи заявки.....	565
Прохождение заявки.....	565
Классификация видов работ по заявкам.....	565
Формы приложений к заявке 06	568

Глава 29. Законы и нормативы	570
Обзор законодательных актов.....	570
Избранные статьи и положения	571
Закон Российской Федерации "О государственной тайне"	571

Федеральный закон Российской Федерации "Об информации, информатизации и защите информации"	573
Закон Российской Федерации "О правовой охране программ для электронных вычислительных машин и баз данных"	575
Федеральный закон Российской Федерации "Об электронной цифровой подписи"	576
Положение по аттестации объектов информатизации по требованиям безопасности информации	576
Положение о сертификации средств защиты информации по требованиям безопасности информации	578
Положение о государственном лицензировании деятельности в области защиты информации	579
Классификация автоматизированных систем и требования по защите информации	581
Концепция защиты средств вычислительной техники и автоматизированных систем от несанкционированного доступа к информации	582
Межсетевые экраны. Показатели защищенности от несанкционированного доступа к информации	583
Показатели защищенности от несанкционированного доступа к информации	585
Документ "Защита от несанкционированного доступа к информации. Термины и определения"	586
Перечень документов	587
Глава 30. Тестирование процедур и механизмов безопасности	589
Общие вопросы	589
Необходимость тестирования	589
Классификация	589
Дружеский взлом	591
Тестирование аварийного плана	592
Обратная связь от пользователей	592
Порядок и методика тестирования	593
Вопросы создания методики	593
Пример методики тестирования: "Проблема 2000"	593
Анализ результатов	597
Общая схема тестирования	597
ЧАСТЬ VIII. ЕСЛИ НАРУШЕНИЕ БЕЗОПАСНОСТИ ВСЕ ЖЕ ПРОИЗОШЛО.	599
Глава 31. Аварийный план	601
Что такое аварийный план?	601
Принципы создания аварийного плана	602
Общие соображения	602

Классификация событий	602
Классификация объектов по приоритетам	603
Шкала резервных объектов.....	604
Структура аварийного плана.....	605
Определения аварийных ситуаций и расстановка приоритетов по объектам и системам	605
Предварительные меры	605
Мероприятия во время катастрофы.....	608
Возвращение к бизнесу.....	610
Методология работы с аварийным планом.....	611
Извлечение уроков.....	611
Тестирование плана.....	612
Обновление плана.....	613
Сохранение данных.....	614
Пример аварийного плана.....	617
Комитет по регламенту административного расчета.....	617
Команда управления продолжением бизнес-деятельности	617
Классификация технологий RAID	619
Глава 32. Реагирование на нарушения информационной безопасности.....	621
Основа функционирования группы реагирования.....	621
Нормативные акты	623
Глава 33. Проведение расследования	626
Предварительные меры.....	626
Определение целей и задач расследования	628
Оперативные действия.....	629
Наблюдение за злоумышленником и сбор улик	631
Удаленное наблюдение за злоумышленником	631
Принципы сбора улик о компьютерных инцидентах.....	631
Методика сохранения данных в затронутых системах	633
Анализ результатов расследования.....	636
Вместо заключения: идеальная служба информационной безопасности.....	637
Послесловие	641
ПРИЛОЖЕНИЯ	643
Приложение 1.....	645
Приложение 2.....	650
Приложение 3.....	662

Приложение 4	664
Приложение 5	667
Приложение 6	670
Приложение 7	672
Приложение 8	675
Приложение 9	676
Приложение 10	682
Приложение 11	685
Приложение 12	686
Приложение 13	687
Приложение 14	708
Перечень документов и литературы	718
Предметный указатель	725

Предисловие

Уважаемый читатель! Перед вами книга по информационной безопасности. Не хочется добавлять слово "очередная". Хорошие книги по информационной безопасности уже присутствуют на рынке. Поэтому, чтобы не просто увеличить список тематических книг еще на одну, мы решили использовать несколько иной подход в изложении.

Во-первых, и "в самых главных", мы постарались не только изложить технические, академические и производственные подходы, ставшие стандартами де-юре или де-факто, но и предложить свои взгляды на тему. Мы хотим поделиться своим опытом, который кто-то примет, а кто-то отвергнет. Главное, чтобы вы, уважаемые читатели, задумались над вопросами, часть ответов, на которые вы найдете в книге, а часть захотите найти вне ее рамок.

Во-вторых, мы попытались объединить свои знания — опыт академического преподавания информационной безопасности с опытом создания и руководства подразделением информационной безопасности крупной финансовой организации, для того чтобы тема была интересна и теоретикам, и практикам. Сама книга построена так, чтобы ее могли читать и те, кто намерен слегка коснуться некоторых вопросов, и те, кто хочет узнать больше, и те, кто хочет досконально разобраться в интересующих их проблемах.

В-третьих, на наш взгляд, уникален и коллектив авторов, которые, создавая книгу, не видели друг друга в лицо, обмениваясь материалами только по электронной почте. Не имея общих интересов (кроме принадлежности к информационной безопасности и желания создать книгу, которую вы держите сейчас в руках), мы, споря друг с другом, критикуя и соглашаясь, пытались выстроить наиболее объективный взгляд на данную тему.

Для того, чтобы эта книга помогла вам в освоении или продвижении в теме информационной безопасности, вы должны позиционировать себя в одной из следующих категорий:



Хочу ознакомиться с основами и базовым уровнем

В эту категорию могут войти специалисты, начинающие свой путь в мир информационной безопасности, руководители подразделений, которые хотят иметь представление, чем занимается их служба защиты информации, а также студенты, которым необходимо быстро написать реферат.



Хочу детально разобраться в данном вопросе

Материалы под данным значком могут использовать специалисты, непосредственно ответственные за те или иные аспекты системы защиты информации предприятия. Они предназначены для тех, кто хочет не только использовать определенные инструменты и механизмы безопасности, но и свободно ориентироваться в том, как они работают.



Не планирую детально изучать тему: мне необходимо получить знания для контроля работы службы информационной безопасности

В данную категорию могут войти менеджеры и технический персонал, курирующие проекты по информационной безопасности, выполняемые другими специалистами.

В зависимости от выбранной вами категории, вы можете читать только то, что написано в текущей главе от начала, обозначенного вашим значком, до появления следующего "невашего" значка.

Если тема предназначена всем трем категориям читателей, дополнительные значки не ставятся.

В качестве обзорного вступления, некоего графического путеводителя по книге, предлагается следующая схема (рис. 1, с. 4—5). На ней кратко указаны этапы и основные моменты жизненного цикла информационной безопасности не только как одной из служб предприятия, но и как отдельного производственного процесса. Названия этих элементов близки к названиям отдельных глав книги, а сами номера глав указаны в кружке рядом с элементом.

Кроме того, вся схема разбита на отдельные этапы, через которые должен проходить процесс функционирования информационной безопасности. Необходимо отметить, что в этом процессе, как и в любой другой постоянно развивающейся системе, образуется цикл возвращений на предыдущую позицию после периодических тестирований, анализа, переосмысления состояния и развития информационной безопасности на конкретном предприятии.

Часть I посвящена этапу создания службы информационной безопасности на предприятии — подбору персонала и постановке задач ее функционирования в самом общем виде. В *части II* детально описан процесс инвентаризации информационных ресурсов — неотъемлемый процесс начала первого или очередного этапа в жизненном цикле информационной безопасности.

Часть III — это попытка изучить и классифицировать те несанкционированные действия, от которых служба будет защищать наш бизнес-процесс, а в *части IV* предложены несколько подходов к определению того, какие из этих действий являются наиболее опасными именно для нашего предприятия. Части *V* и *VI* посвящены непосредственно техническим методам и средствам защиты информации (отдельно изложена теория криптографии — самого мощного на сегодняшний день технического средства). В *части VII* рассматриваются возможные административные меры в области информационной безопасности, а также юридические вопросы, возникающие в связи с применением как технических, так и административных средств. *Часть VIII* описывает методики оперативного реагирования на произошедшие инциденты — очень важный аспект функционирования службы информационной безопасности.

Некоторые электронные материалы, упомянутые в данной книге, размещены в Интернете на сайте <http://www.metacom.ru/ise2003/>. Авторы с благодарностью примут любые конструктивные замечания и предложения по электронной почте ise2003@metacom.ru, ise2003@hotmail.ru.

Желаем успехов в освоении этой увлекательной и сложной области информационных технологий.

Благодарности

Хочу сказать огромное спасибо своим родным, близким и друзьям, которые всегда помогали мне и принимали меня таким, какой я есть. Также моя благодарность всем, с кем я работал и общался в сфере информационных технологий.

Искан

С глубоким уважением приношу благодарности моим Учителям: Анохину Михаилу Борисовичу, Пышницкому Константину Михайловичу, Приходскому Виктору Ефимовичу и Мыльникову Михаилу Радиевичу за качественное базовое образование и веру в мои начинания. Без их огромного влияния мое участие в создании подобного материала было бы просто невозможно.

Также мне очень приятно выразить признательность специалистам по информационной безопасности: Андрею Хабачеву (ООО "Метаком"), Сергею Цветкову (ООО "Метаком") и Дмитрию Шаратинову (ОАО КБ "СеверГазБанк") за плодотворное обсуждение многих вопросов, освещенных в данной книге.

Андрей Беляев

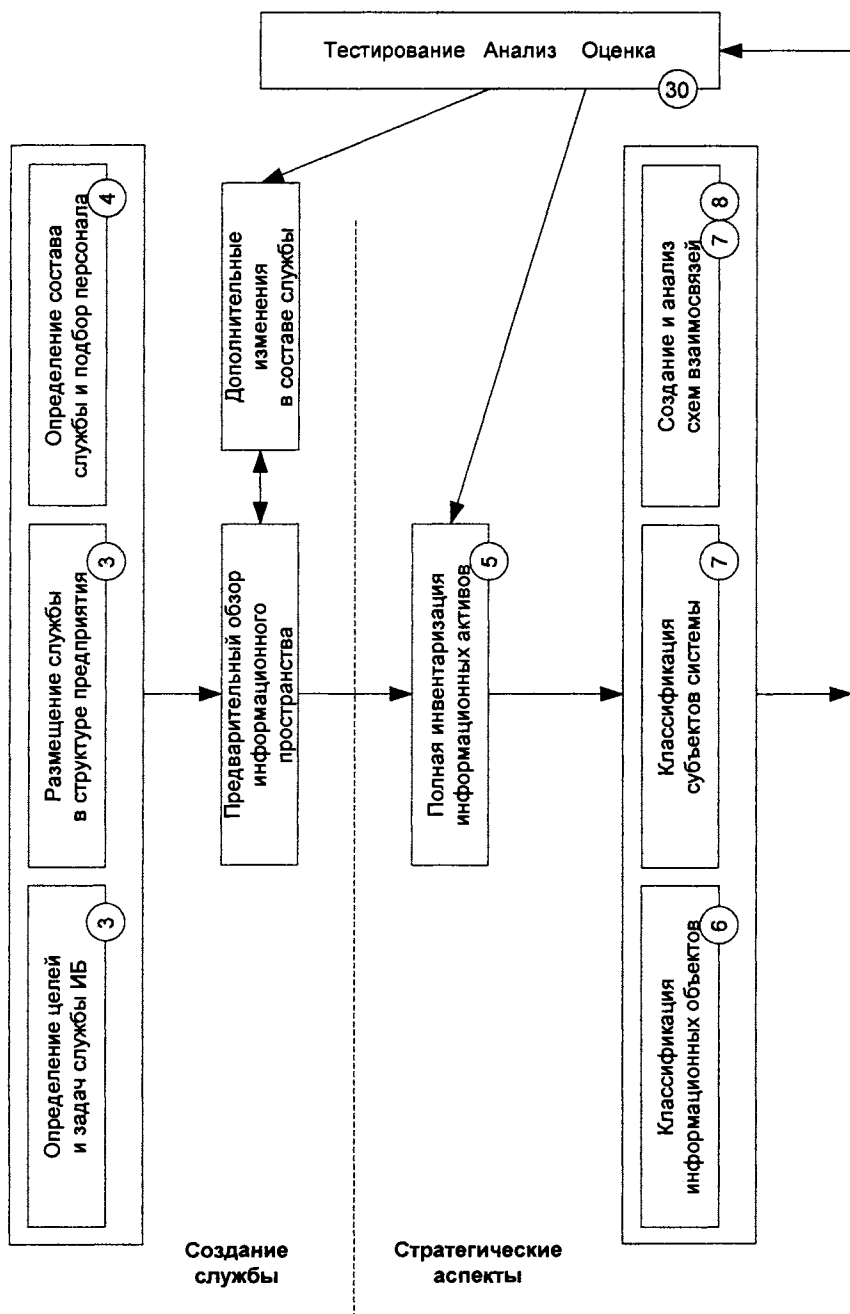
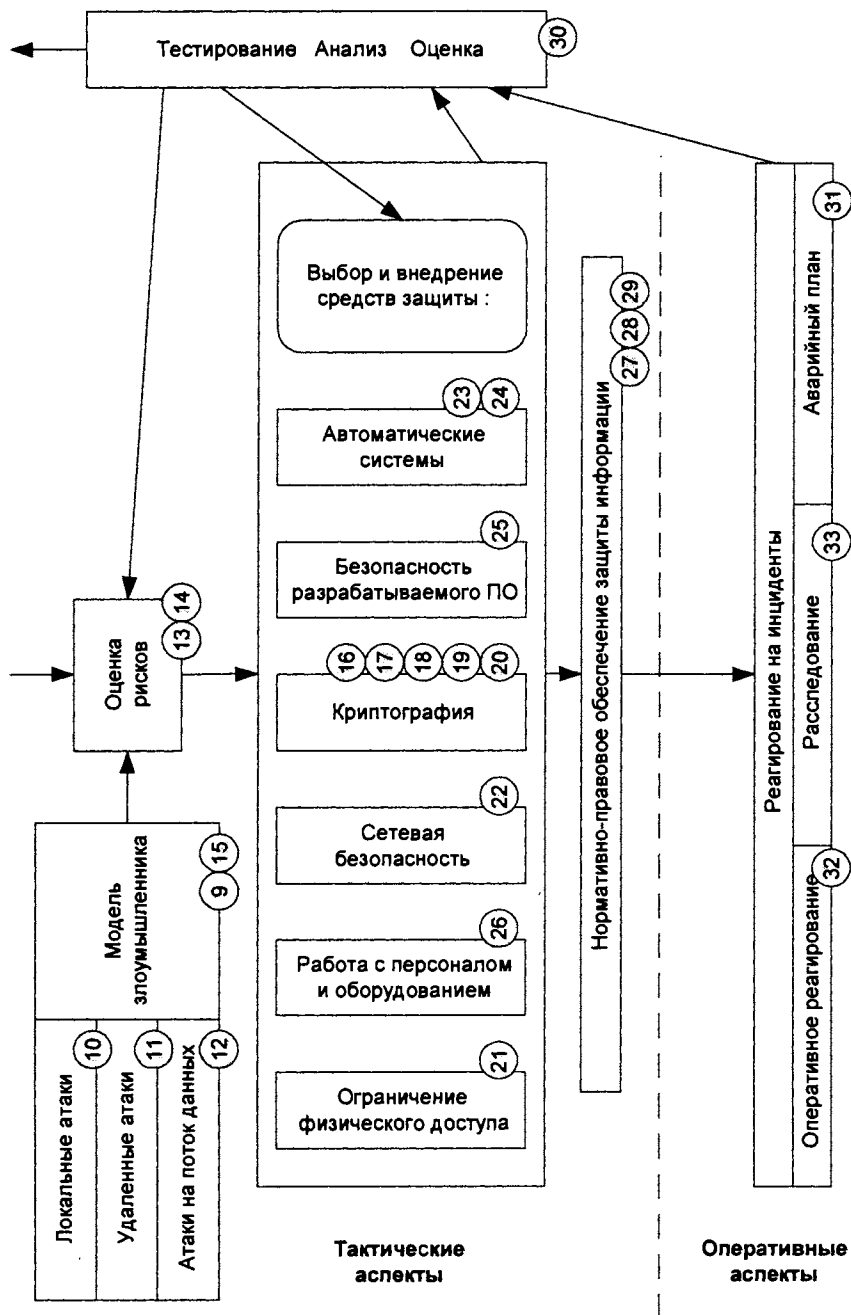


Рис. 1. Путеводитель-схема по книге:



Глоссарий

<i>API</i>	<i>(Application Programming Interface)</i> интерфейс прикладного программирования — набор протоколов и точек программных вызовов, обеспечивающий взаимодействие с информационной системой через вызов ее исполняемого кода
<i>ASCII</i>	<i>(American Standard Code Information Interchange)</i> Американский стандартный код обмена информацией — стандарт 8-битного кодирования символов
<i>BIOS</i>	<i>(Basic Input/Output System)</i> базовая система ввода/вывода — <i>программное обеспечение</i> , выполняющее функции низкоуровневого доступа к аппаратному обеспечению и его стандартизации
<i>CGI</i>	<i>(Common Gateway Interface)</i> интерфейс единого шлюза — технология динамического создания <i>HTML</i> -страниц, заключающаяся в запуске на сервере <i>WWW</i> при каждом запросе связанного со страницей <i>программного обеспечения</i> в автоматическом режиме с последующей передачей пользователю результата его работы в виде <i>HTML</i> -страницы
<i>checklist</i>	список мероприятий, направленных на достижение определенного результата (например, повышения безопасности данной информационной системы до заданного уровня)
<i>debugger</i>	(англ.) отладчик
<i>e-mail</i>	(англ.) электронная почта
<i>firmware</i>	<i>программное обеспечение</i> , созданное для управления <i>аппаратным обеспечением</i> ; в большинстве случаев поставляется на <i>носителе</i> , встроенном в аппаратное обеспечение, и исполняется на микроконтроллере аппаратного обеспечения

<i>hardware</i>	(англ.) <i>аппаратное обеспечение</i>
<i>HTML</i>	(<i>HyperText Markup Language</i>) язык разметки гипертекста — стандарт гипертекстовой разметки; позволяет включать в текст стили оформления, графику, аудиофон и ссылки на другие ресурсы
<i>hub</i>	(англ.) узел, технически то же, что <i>концентратор</i>
<i>IP-адрес</i>	<i>сетевой адрес</i> в протоколе <i>IP</i>
<i>MAC-адрес</i>	<i>сетевой адрес</i> в протоколе <i>Ethernet</i>
<i>OSI</i>	(<i>Open System Interconnection</i>) <i>взаимодействие открытых систем</i> — унифицированная модель взаимодействия открытых систем, разработанная Международной организацией по стандартизации (ISO — International Organization for Standardization); состоит из 7 уровней, каждый из которых выполняет определенную задачу при передаче данных: физический, канальный, сетевой, транспортный, сеансовый, представления, прикладной; см. также <i>стек сетевых протоколов</i>
<i>patch</i>	(англ.) <i>заплата</i>
<i>repeater</i>	(англ.) <i>повторитель</i>
<i>screen saver</i>	<i>модуль операционной системы</i> , производящий выключение монитора и/или автоматическую блокировку консоли компьютера по прошествии определенного времени с момента последней активности пользователя за консолью
<i>software</i>	(англ.) <i>программное обеспечение</i>
<i>switch</i>	(англ.) <i>коммутатор</i>
<i>TCP-порт</i>	<i>адрес сетевой службы (сервиса, демона)</i> в протоколе <i>TCP (Transmission Control Protocol)</i>
<i>UDP-порт</i>	<i>адрес сетевой службы (сервиса, демона)</i> в протоколе <i>UDP (User Datagram Protocol)</i>
<i>Unicode</i>	стандарт 16-битного кодирования символов
<i>URL</i>	(<i>Uniform Resource Locator</i>) — уникальное глобальное имя ресурса в сети <i>Интернет</i>

<i>WWW</i>	<i>(World Wide Web, "всемирная паутина")</i> – система <i>HTML</i> -серверов в сети <i>Интернет</i>
<i>Администратор</i>	сотрудник предприятия, выполняющий <i>администрирование</i> системы
<i>Администрирование</i>	процесс установки, обновления, управления, поддержания работоспособности информационной системы и динамической ее настройки в соответствии с требованиями бизнес-процесса; состоит из анализа производственных требований и служебных сообщений самой системы с последующей модификацией ее <i>настроек</i> ; однако часто кроме этого включает в себя обучение пользователей специфике работы в данной системе
<i>База данных</i>	информационная система, имеющая целью хранение и обеспечение доступа к определенному (обычно достаточно большому) объему строго структурированной информации; основными целями имеет обеспечение надежности хранения информации, удобства и скорости доступа к ней
<i>Браузер</i>	программа для загрузки из сети на <i>хост</i> и дальнейшего просмотра <i>HTML</i> -страниц; в общем случае программный модуль для последовательного просмотра информации
<i>Виртуальная машина</i>	информационная система (обычно модуль, подсистема), предоставляющая услуги пошагового исполнения программ, состоящих из команд, отличных от кода микропроцессора (возможно, даже в виде исходного текста); то же, что <i>интерпретатор</i>
<i>Выделенная линия</i>	физический канал связи, предоставленный в полное и постоянное владение двум <i>сетевым узлам</i> , соединенным с его помощью; противопоставляется <i>коммутируемой линии</i>
<i>Демон</i>	<i>программное обеспечение (в операционной системе Unix)</i> , выполняющееся в фоновом режиме в течение всего процесса функционирования информационной системы

Директория	то же, что <i>каталог</i>
Домен	в общем случае информационная система, состоящая из нескольких информационных систем, объединенных какими-либо общими характеристиками (задачами, месторасположением, администрированием, правовыми аспектами)
Драйвер	единица <i>программного обеспечения</i> , обеспечивающая взаимодействие между двумя объектами в рамках <i>операционной системы</i> ; обычно прерогативой драйвера является взаимодействие между <i>аппаратным</i> и <i>программным обеспечением</i> информационной системы, однако с появлением множества <i>стеков</i> различных <i>протоколов</i> драйверами стали называть и ПО, выполняющее роль адаптера между двумя программными объектами (<i>протоколами</i>)
Журнал	структурированный список служебных (информационных) сообщений <i>программного обеспечения</i> (в том числе <i>операционной системы</i>) обслуживающему техническому персоналу
Заплатка	информация, поставляемая производителем <i>программного обеспечения</i> пользователям, о том, что нужно изменить в ПО, с тем чтобы исправить ошибку, допущенную на этапе создания ПО; обычно оформляется в виде <i>исполнимого файла</i>
Имя регистрации	уникальная в рамках данной информационной системы символьная информация, однозначно идентифицирующая конкретного пользователя
Инсталляция (объект)	набор <i>исполнимых</i> и информационных <i>файлов</i> , производящий установку (добавление) нового <i>программного обеспечения</i> в информационную систему
Инсталляция (процесс)	процесс установки (добавления) в данную информационную систему нового <i>программного</i> или <i>аппаратного обеспечения</i> ; в отношении программного обеспечения различают <i>автоматическую</i> (без участия человека) и <i>автоматизированную</i> (при участии человека, обычно <i>администратора</i>) инсталляцию
Интернет	<i>глобальная компьютерная сеть</i> , основанная на сетевом протоколе <i>IP</i> и получившая подавляющее распространение на территории Земного шара; образовался в 1970-х годах на базе компьютерной сети ARPANET

<i>Интерпретатор</i>	см. <i>виртуальная машина</i>
<i>Интерфейс пользовательский</i>	комплекс методов, характеристик и специфических решений, используемый <i>программным обеспечением</i> при организации его взаимодействия с пользователем
<i>Интерфейс программный</i>	набор <i>протоколов</i> , поддерживаемый <i>программным обеспечением</i> для взаимодействия с внешним по отношению к нему программным обеспечением
<i>Исключительная ситуация</i>	ситуация, когда дальнейшее функционирование <i>аппаратного</i> или <i>программного обеспечения</i> по заложенному в них алгоритму невозможно; требует аварийной остановки процесса или принятия в специальной <i>процедуре обработки</i> нестандартного алгоритма действий; извещение об исключительной ситуации реализуется с помощью встроенных средств микропроцессора и <i>операционной системы</i>
<i>Исполнимый файл</i>	<i>файл</i> , содержащий образ <i>программного обеспечения</i> в формате, готовом для запуска в данной информационной системе
<i>Каталог</i>	именованный набор <i>файлов</i> и других <i>каталогов</i> (именуемых по отношению к нему <i>подкаталогами</i>), объединенных вместе как часть общей древовидной структуры файлов на носителе
<i>Кодек</i>	программное обеспечение, выполняющее функции прямого (кодирование) и обратного (декодирование) преобразования данных из одного формата в другой; получило название от сокращения "кодер-декодер"
<i>Коммутатор</i>	<i>сетевой узел</i> , выполняющий маршрутизацию <i>сетевых пакетов</i> на 2-м уровне сетевой модели <i>OSI</i> (например, для сети <i>Ethernet</i> маршрутизация производится на основании <i>MAC-адреса</i> получателя пакета)
<i>Коммутируемая линия</i>	физический канал связи, предоставляемый двум <i>сетевым узлам</i> , соединенным с его помощью, по их требованию на время сеанса связи; в остальное время может использоваться другими парами сетевых узлов или бездействовать; противопоставляется <i>выделенной линии</i>

<i>Компилятор</i>	<i>программное обеспечение</i> , выполняющее преобразование исходного текста программы, написанной на каком-либо языке программирования, в программный код, готовый к исполнению на микропроцессоре и <i>виртуальной машине</i> ; обычно скомпилированный код автоматически оформляется компилятором в виде <i>исполнимого файла</i>
<i>Консоль</i>	информационное и программное обеспечение, предоставляющее интерфейс непосредственного, обычно низкоуровневого <i>администрирования</i> информационной системы или ее конкретного компонента; часто под консолью понимается место управления (администрирования) системой
<i>Конфигурация АО</i>	набор <i>аппаратного обеспечения</i> , установленного в данной информационной системе, вместе со спецификой его <i>настроек</i> для данной системы
<i>Конфигурация ПО</i>	комплекс <i>настроек программного обеспечения</i> , специфичных для данной информационной системы
<i>Концентратор</i>	в широком смысле <i>сетевой узел</i> , осуществляющий объединение нескольких сегментов сети
<i>Копирование информации</i>	процесс создания копии данных на ином или том же (под другим именем) <i>носителе информации</i>
<i>Лог</i>	<i>журнал</i> , оформленный с минимальным структурированием (обычно в виде текстового файла с переводом строки после каждого сообщения)
<i>Маршрутизация</i>	процесс выбора направления дальнейшего движения <i>сетевого пакета</i> , производимый <i>сетевым узлом</i> ; различают <i>статическую</i> (жестко заданную на этапе администрирования) и <i>динамическую</i> (варьирующуюся в зависимости от текущей ситуации в сети) маршрутизацию
<i>Модем</i>	<i>аппаратное обеспечение</i> , выполняющее функции модуляции цифрового сигнала для передачи по аналоговым линиям связи и демодуляции на приемной стороне; получило название от сокращения "модуляция-демодуляция"
<i>Модуль</i>	функционально независимая часть <i>программного обеспечения</i>

<i>Многозадачная система</i>	<i>операционная система</i> , поддерживающая (на многопроцессорных системах) или эмулирующая (на однопроцессорных системах) одновременное исполнение нескольких программных продуктов
<i>Многопользовательская система</i>	информационная система, поддерживающая различие между пользователями при работе с ней; в простейшем случае различие заключается в ограничении общего доступа к системе с ведением журнала регистрации пользователей, в более совершенных системах в зависимости от исполняемых обязанностей пользователь наделяется определенными правами доступа; ключевой информацией при выборе между пользователями является <i>имя регистрации</i>
<i>Монопольный доступ</i>	схема организации доступа к <i>ресурсу</i> , при которой в каждый конкретный момент времени права на ресурс имеет только один пользователь системы; противопоставляется <i>совместному доступу</i>
<i>Настройки</i>	конкретный набор значений варьируемых параметров информационной системы
<i>Носитель информации</i>	физический объект, обладающий свойствами хранения и возможности копирования (на него/с него) информации; классифицируют <i>электронные</i> и <i>неэлектронные</i> (бумажные, фотографические и т. п.) носители информации, <i>съёмные</i> (гибкие магнитные диски, магнитные ленты, компакт-диски, Flash-диски и т. п.) и <i>стационарные</i> (например, жесткий магнитный диск) электронные носители информации, <i>энергозависимые</i> (например, оперативная память) и <i>энергонезависимые</i> носители информации
<i>Он-лайн</i>	состояние подключенности к информационной системе; характеризуется незамедлительным отображением всех изменений в системе (близкое понятие — <i>режим реального времени</i>); противопоставляют режиму <i>офф-лайн</i>
<i>Отладчик</i>	логическое устройство (обычно программный модуль), позволяющее управлять исполнением данной программы по шагам, с возможным анализом состояния переменных, условий, циклов и т. д.

<i>Офф-лайн</i>	состояние отключенности от информационной системы; характеризуется отсутствием актуализированности локальных данных по отношению к текущим изменениям в информационной системе; противопоставляют режиму <i>он-лайн</i>
<i>Папка</i>	то же, что <i>каталог</i>
<i>Повторитель</i>	<i>сетевой узел</i> , производящий копирование (размножение) полученного <i> сетевого пакета </i> в адрес всех остальных сетевых узлов, подключенных к нему
<i>Поисковая система</i>	система, осуществляющая поиск информации (файлов) по какому-либо критерию, заданному пользователем; термин часто употребляется в отношении <i>глобальных поисковых систем</i> , осуществляющих поиск информации в <i>WWW</i> и оформляющих результат как HTML-страницы
<i>Почтовая программа</i>	<i>программное обеспечение</i> , предоставляющее пользователю услуги создания, отправки, получения и хранения <i>электронной почты</i>
<i>Приложение</i>	информационная система (набор <i>программного обеспечения</i>), выполняющая предопределенную производственную задачу по созданию, хранению, обработке, пересылке информации
<i>Продвинутый пользователь</i>	пользователь, обладающий расширенными знаниями о системе (либо ее части), на которого возложены некоторые права и обязанности <i>администратора</i> (обычно в отношении конкретного приложения, но не всей информационной системы в целом);
<i>Протокол</i>	набор соглашений, определяющих правила обмена информацией в рамках информационной системы или на границах различных информационных систем
<i>Профиль</i>	набор <i>настроек</i> системы (в отношении служебных параметров или конкретного пользователя) в целостности своих конкретных значений; наличие нескольких различных профилей для одного и того же набора параметров позволяет производить быструю переконфигурацию системы в случае изменения бизнес-требований и в экстренных ситуациях

<i>Процесс</i>	субъект, выполняющий активные действия над объектом или системой, при этом обычно не контролируемый напрямую пользователем; обычно процесс является способом функционирования программного обеспечения; для <i>авторизованного</i> (санкционированного к использованию) программного обеспечения пользователь может знать принципы, механизм и результат работы процесса, а также при желании может контролировать ход его исполнения с помощью специальных программ-отладчиков
<i>Рабочая станция</i>	компьютер, используемый для взаимодействия человека и информационной системы; противопоставляется <i>серверу</i>
<i>Реестр</i>	<i>база данных настроек операционной системы Windows и приложений</i> , работающих под ее управлением
<i>Репитер</i>	(калька от англ. <i>Repeater</i>) — <i>повторитель</i>
<i>Ресурс</i>	информационный объект, уникально характеризующийся производственным или вспомогательным наполнением, месторасположением и правами доступа к нему
<i>Сайт</i>	набор HTML-страниц, объединенных смысловым наполнением
<i>Свитч</i>	(калька от англ. <i>Switch</i>) — <i>коммутатор</i>
<i>Сегмент сети</i>	участок <i>сети</i> — канал связи и набор узлов между активными сетевыми устройствами; различают <i>сегменты коллизий</i> (ограничиваемые устройствами маршрутизации 2-го уровня) и <i>сегменты широковещания</i> (ограничиваемые устройствами маршрутизации 3-го уровня)
<i>Сервер</i>	компьютер, не используемый для непосредственного взаимодействия человека и информационной системы; обычно выполняет функции хранения и/или определенных видов обработки информации и используется одновременно несколькими рабочими станциями; противопоставляется <i>рабочей станции</i>

<i>Сервис</i>	<i>программное обеспечение (в операционных системах Windows NT, 2000, XP), выполняющееся в фоновом режиме в течение всего процесса функционирования информационной системы; то же, что демон</i>
<i>Сервис</i>	<i>услуга, предоставляемая пользователям Интернета удаленно</i>
<i>Сессия</i>	<i>в общем случае сеанс взаимодействия между субъектом и объектом информационной системы</i>
<i>Сетевой адрес</i>	<i>набор информации, уникально идентифицирующий хост или узел сети в пределах определенной зоны; используется для маршрутизации сетевых пакетов</i>
<i>Сетевой интерфейс</i>	<i>точка соприкосновения хоста или сетевого узла с сегментом сети; устройство взаимодействия хоста или сетевого узла и среды передачи данных</i>
<i>Сетевой пакет</i>	<i>блок передаваемых по сети данных, дополненный служебной информацией сетевых служб</i>
<i>Сеть</i>	<i>программно-аппаратный комплекс, выполняющий функции соединения территориально разнесенных компьютеров (хостов) для обеспечения возможности копирования информации без использования съемных носителей; состоит из узлов сети и сегментов сети; различают локальную (в пределах одного здания), региональную (в пределах района, города, области, страны) и глобальную (охватывающую Земной шар) сеть</i>
<i>Скрипт</i>	<i>программа, исполняемая с помощью интерпретатора</i>
<i>Совместный доступ</i>	<i>схема организации доступа к ресурсу, при которой в каждый конкретный момент времени права на ресурс имеет несколько пользователей системы; в простейшем случае один пользователь системы имеет право на модификацию ресурса, неограниченное количество — на чтение, при использовании более сложных алгоритмов возможно предоставление доступа на модификацию одновременно нескольким пользователям; противопоставляется монопольному доступу</i>

Стек	организованная специальным образом служебная область оперативной памяти микропроцессорной системы; используется для временного хранения данных, управления вызовами процедур и передачей им параметров вызова; в общем случае схема управления очередностью порций информации, организованная по принципу <i>"последним пришел — первым ушел"</i>
Стек сетевых протоколов	набор <i>протоколов</i> сети, используемых одновременно на разных уровнях сетевого взаимодействия модели <i>OSI</i>
СУБД	система управления базами данных; предоставляет функции единообразного <i>администрирования</i> и доступа к <i>базам данных</i>
Счет	то же, что <i>учетная запись</i>
Терминал	устройство — точка физического взаимодействия пользователя с информационной системой; в подавляющем числе случаев роль терминала выполняет персональный компьютер (<i>рабочая станция</i> или портативный компьютер), однако к терминалам относятся и информационные табло, и аппаратура контроля физического доступа, банкоматы, телефонные аппараты с интерактивными голосовыми функциями и т. п.
Транзакция	некое действие или набор действий над объектом или группой объектов, воспринимаемое как единое целое; транзакция не может быть выполнена частично — в случае ее аварийного останова все объекты, вовлеченные в транзакцию, возвращаются в исходное состояние; чаще всего данная технология используется при передаче информации от одного объекта к другому, в этом случае транзакция завершается только после того, как данные восприняты и начали обрабатываться вторым объектом; программно транзакции реализуются с помощью дополнительных специализированных процедур
Трафик	поток данных, передаваемых по <i>сети</i>
Узел сети	служебное устройство сети; выполняет функции <i>маршрутизации</i> и/или преобразования сетевых <i>протоколов</i>
Утилита	единица <i>программного обеспечения</i> , выполняющая обычно одну или небольшой набор определенных системных функций

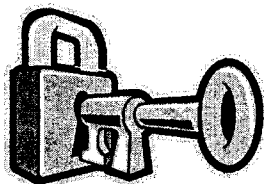
Учетная запись	набор данных, привязанных к данному субъекту (обычно пользователю); включает <i>имя регистрации</i> , пароль, время доступа, адрес разрешенного доступа и т. п.
Файл	именованный упорядоченный набор данных на <i>носителе</i>
Файлы конфигураций	<i>файлы</i> , содержащие в том или ином виде информацию о <i>конфигурации программного обеспечения</i>
Флаг	параметр программного или аппаратного обеспечения, принимающий только два значения — за ними закрепились термины <i>поднят</i> ("1") и <i>опущен</i> ("0")
формат	комплекс соглашений о взаимном расположении и типе кодирования применительно к конкретному функциональному набору информации
Форматирование информации	приведение информации в соответствие определенному <i>формату</i>
Форматирование носителя	процесс подготовки носителя к возможности записи на него информации; выполняет полное удаление ранее записанных на носитель данных, если таковые были
Хаб	(калька от англ. <i>Hub</i>) — узел
Хост	обычно компьютер в аспекте подключения его к <i>сети</i> — начальная и конечная точка передачи информации по сети; в общем случае <i>сетевой узел</i> или устройство, имеющее адрес сетевого уровня (например, <i>IP-адрес</i>)
Чат	одновременная беседа в виде текстовых сообщений нескольких пользователей, реализованная с помощью средств, предоставляемых <i>сетью</i>
Широковещательный сетевой пакет	служебный <i>сетевой пакет</i> , имеющий целью достичь несколько (более одного) <i>хостов</i> или <i>узлов сети</i> ; различают <i>multicast-пакеты</i> (адресованные нескольким узлам/хостам) и <i>broadcast-пакеты</i> (адресованные всем узлам/хостам в текущем <i>сегменте широковещания</i>)
Экспорт информации	<i>копирование информации</i> вовне информационной системы
Электронная почта	система электронных сообщений с помощью средств, предоставляемых <i>сетью</i> ; ориентирована (в отличие от, например, <i>чата</i>) на связь между двумя пользователями — <i>отправителем</i> и <i>получателем</i>

Таблица. Наиболее распространенные сетевые протоколы и стандарты

Название	Стандарт	Функции
ARP	RFC-826	Преобразование IP-адреса в MAC-адрес
ATM	ATM	Физический, канальный и сетевой уровни
BGP	RFC-1771	Динамическая маршрутизация между крупными участками сети
BOOTP	RFC-1533	Удаленная загрузка программного обеспечения при старте компьютера
DHCP	RFC-1541	Автоматическое динамическое выделение IP-адресов хостам и настройка их сетевых параметров
DNS	RFC-1034	Преобразование доменного имени в IP-адрес
Ethernet	IEEE 802.3	Канальный уровень
Frame Relay	FRF.*	Канальный уровень
FTP	RFC-959	Передача файлов
HTTP	RFC-2068	Передача HTML-страниц и включенных в них ресурсов
ICMP	RFC-792	Служебная и тестовая информация об IP-сети
IMAP	RFC-2060	Электронная почта
IP	RFC-791	Сетевой уровень
IPX	Novell IPX	Сетевой уровень
ISDN	ISDN	Канальный уровень
MIME	RFC-2045	Уровень представления
NETBIOS	RFC-1001	Сетевой уровень
NNTP	RFC-977	Электронные конференции ("электронные новости")
NTP	RFC-1119	Синхронизация времени
OSPF	RFC-1583	Динамическая маршрутизация в мелких и средних участках сети
POP	RFC-1939	Прием и хранение электронной почты
PPP	RFC-1661	Канальный уровень
RIP	RFC-2453	Динамическая маршрутизация в мелких и средних участках сети

Таблица (окончание)

Название	Стандарт	Функции
RPC	RFC-1831	Удаленный вызов исполнимых процедур
RTP	RFC-1889	Передача информации в реальном режиме времени (аудио, видео)
SMTP	RFC-821	Отправка электронной почты
SNMP	RFC-1157	Администрирование сетевых устройств
TCP	RFC-793	Транспортный и сеансовый уровни
TELNET	RFC-854	Удаленная консоль
TFTP	RFC-1350	Передача файлов
UDP	RFC-768	Транспортный уровень
X.25	ITU-T X.25	Сетевой уровень



Часть I

Служба информационной безопасности

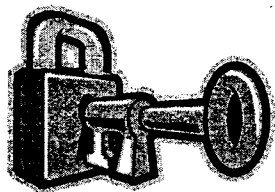
**Глава 1. Общие понятия информационной
безопасности**

Глава 2. Терминология

**Глава 3. Задачи и принципы организации службы
информационной безопасности**

Глава 4. Создание службы безопасности

Глава 1



Общие понятия информационной безопасности

Определение и цели информационной безопасности



В беседах со специалистами по защите информации зачастую обнаруживается, что взгляды и терминология в этой относительно новой области различаются иногда до почти противоположных. При прямом вопросе об определении информационной безопасности можно услышать такие разноречивые термины, как "защита данных", "контроль использования", "борьба с хакерами" и т. д.

Между тем существуют сложившиеся определения самой информационной безопасности и примыкающего к ней круга понятий. Иногда они различаются у различных специалистов (или школ). Случается, в определениях просто используют синонимы, иногда — меняются местами даже целые группы понятий. Поэтому первоначально необходимо четко определиться, о чем же будет идти речь в данной книге. Хотя оба автора получали образование в области информационной безопасности совершенно независимо друг от друга, тем не менее их определения и понятия практически совпадают, поэтому было решено использовать именно данный подход.

Для всей области знаний, охватываемой данной книгой, будет использоваться термин "информационная безопасность". Иногда, особенно в классификации зарубежных агентств по найму на работу, "информационная безопасность" рассматривается как один из подразделов общей безопасности, наряду с такими понятиями как "компьютерная безопасность", "сетевая безопасность", "безопасность телекоммуникаций", "безопасность данных".

На наш взгляд, понятие "информационная безопасность" более широкое, так как охватывает всё, что взаимодействует с информацией, и все вышеперечисленные понятия — это подразделы или отдельные направления информационной безопасности.

Информационная безопасность — это комплекс мероприятий, обеспечивающий для охватываемой им информации следующие факторы:

- ❑ *конфиденциальность* — возможность ознакомиться с информацией (именно с данными или сведениями, несущими смысловую нагрузку, а не с последовательностью бит их представляющих) имеют в своем распоряжении только те лица, кто владеет соответствующими полномочиями;
- ❑ *целостность* — возможность внести изменение в информацию (опять речь идет о смысловом выражении) должны иметь только те лица, кто на это уполномочен;
- ❑ *доступность* — возможность получения авторизованного доступа к информации со стороны уполномоченных лиц в соответствующий санкционированный для работы период времени.

Иногда можно встретить определения перечисленных факторов в варианте от противного, например разглашение или раскрытие, модификация (изменение или искажение) и уничтожение или блокирование. Главное, чтобы не был искажен смысл, заложенный в указанных определениях.

Это не полный список факторов, выделены именно эти три понятия, поскольку они обычно встречаются практически во всех определениях информационной безопасности и не вызывают споров. На наш взгляд, необходимо включить дополнительные факторы и понять различие между ними, а именно:

- ❑ *учет*, т. е. все значимые действия лица, выполняемые им в рамках, контролируемых системой безопасности (даже если они не выходят за рамки определенных для этого лица правил), должны быть зафиксированы и проанализированы;
- ❑ *неотрекаемость или апеллируемость* (характерно для организаций, в которых функционирует обмен электронными документами с юридической, финансовой или другой значимостью), т. е. лицо, направившее информацию другому лицу, не может отречься от факта направления информации, а лицо, получившее информацию, не может отречься от факта ее получения.

Отличие между двумя этими факторами, возможно, видимое не сразу, заключается в следующем. Учет обычно ведется средствами электронных регистрационных журналов, которые используются в основном только уполномоченными службами, и его основное отличие — в регулярности анализа этих журналов. Апеллируемость обеспечивается средствами криптографии (электронно-цифровой подписью), и ее характерная черта — возможность использования в качестве доказательного материала во внешних инстанциях, например в суде, при наличии соответствующего законодательства.

Механизмы информационной безопасности



Перечисленные объективные факторы или цели информационной безопасности обеспечиваются применением следующих механизмов или принципов:

- ❑ *политика* — набор формальных (официально утвержденных либо традиционно сложившихся) правил, которые регламентируют функционирование механизма информационной безопасности;
- ❑ *идентификация* — определение (распознавание) каждого участника процесса информационного взаимодействия перед тем как к нему будут применены какие бы то ни было понятия информационной безопасности;
- ❑ *аутентификация* — обеспечение уверенности в том, что участник процесса обмена информацией идентифицирован верно, т. е. действительно является тем, чей идентификатор он предъявил;
- ❑ *контроль доступа* — создание и поддержание набора правил, определяющих каждому участнику процесса информационного обмена разрешение на доступ к ресурсам и уровень этого доступа;
- ❑ *авторизация* — формирование профиля прав для конкретного участника процесса информационного обмена (аутентифицированного или анонимного) из набора правил контроля доступа;
- ❑ *аудит и мониторинг* — регулярное отслеживание событий, происходящих в процессе обмена информацией, с регистрацией и анализом predetermined значимых или подозрительных событий. Понятия "аудит" и "мониторинг" при этом несколько различаются, так как первое предполагает анализ событий постфактум, а второе приближено к режиму реального времени;
- ❑ *реагирование на инциденты* — совокупность процедур или мероприятий, которые производятся при нарушении или подозрении на нарушение информационной безопасности;
- ❑ *управление конфигурацией* — создание и поддержание функционирования среды информационного обмена в работоспособном состоянии и в соответствии с требованиями информационной безопасности;
- ❑ *управление пользователями* — обеспечение условий работы пользователей в среде информационного обмена в соответствии с требованиями информационной безопасности. В данном случае под пользователями понимаются все, кто использует данную информационную среду, в том числе и администраторы;
- ❑ *управление рисками* — обеспечение соответствия возможных потерь от нарушения информационной безопасности мощности защитных средств (то есть затратам на их построение);

- *обеспечение устойчивости* — поддержание среды информационного обмена в минимально допустимом работоспособном состоянии и соответствии требованиям информационной безопасности в условиях деструктивных внешних или внутренних воздействий.

Таким образом, перечислено то, за счет чего достигаются определенные выше цели информационной безопасности (в некоторых источниках описанные принципы, например, аутентификация, переносятся в цели). На наш взгляд, аутентификация сама по себе не может быть целью информационной безопасности. Она является лишь методом определения участника информационного обмена, чтобы далее определить, какая, например, политика в отношении конфиденциальности или доступности должна быть применена к данному участнику.

Инструментарий информационной безопасности



Теперь рассмотрим, какие существуют средства или инструменты, которыми реализованы описанные принципы или механизмы. Естественно, что привести полный список здесь просто невозможно — он в значительной степени зависит от конкретной ситуации, в свете которой рассматривается тот или иной аспект информационной безопасности. Кроме того, возможно, кто-либо захочет переместить некоторые пункты из списка механизмов в список средств или наоборот. Наши рассуждения имеют следующий базис. Например, персонал занимается аудитом, который обеспечивает учет. Значит, персонал — это средство, аудит — механизм, а учет — цель. Или пароли, обеспечивающие аутентификацию, сохраняются в зашифрованном виде, аутентификация предшествует, например, разрешению на модификацию. Значит, криптография — средство защиты паролей, пароли используются для механизма аутентификации, аутентификация предшествует обеспечению целостности.

Перечислим основные средства (инструменты) информационной безопасности:

- *персонал* — люди, которые будут обеспечивать претворение в жизнь информационной безопасности во всех аспектах, то есть разрабатывать, внедрять, поддерживать, контролировать и исполнять;
- *нормативное обеспечение* — документы, которые создают правовое пространство для функционирования информационной безопасности;
- *модели безопасности* — схемы обеспечения информационной безопасности, заложенные в данную конкретную информационную систему или среду;
- *криптография* — методы и средства преобразования информации в вид, затрудняющий или делающий невозможным несанкционированные

операции с нею (чтение и/или модификацию), вместе с методами и средствами создания, хранения и распространения *ключей* — специальных информационных объектов, реализующих эти санкции;

- ❑ *антивирусное обеспечение* — средство для обнаружения и уничтожения зловредного кода (вирусов, троянских программ и т. п.);
- ❑ *межсетевые экраны* — устройства контроля доступа из одной информационной сети в другую;
- ❑ *сканеры безопасности* — устройства проверки качества функционирования модели безопасности для данной конкретной информационной системы;
- ❑ *системы обнаружения атак* — устройства мониторинга активности в информационной среде, иногда с возможностью принятия самостоятельного участия в указанной активной деятельности;
- ❑ *резервное копирование* — сохранение избыточных копий информационных ресурсов на случай их возможной утраты или повреждения;
- ❑ *дублирование (резервирование)* — создание альтернативных устройств, необходимых для функционирования информационной среды, предназначенных для случаев выхода из строя основных устройств;
- ❑ *аварийный план* — набор мероприятий, предназначенных для претворения в жизнь, в случае если события происходят или произошли не так, как было предопределено правилами информационной безопасности;
- ❑ *обучение пользователей* — подготовка активных участников информационной среды для работы в условиях соответствия требованиям информационной безопасности.

Возможно, некоторые понятия слишком укрупнены (криптография), некоторые, наоборот, детализированы (сканеры). Основной целью этого списка ставилось показать типовой набор, характерный для предприятия, которое развивает у себя службу информационной безопасности.

Основные направления информационной безопасности



Теперь осталось рассмотреть основные направления информационной безопасности, которые иногда различают между собой. Собственно, на наш взгляд их всего два — физическая и компьютерная безопасность. Однако, учитывая имеющиеся различия в определениях, можно охарактеризовать их следующим образом.

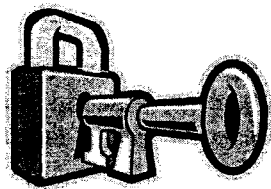
Физическая безопасность — обеспечение сохранности самого оборудования, предназначенного для функционирования информационной среды, контроль доступа людей к этому оборудованию. Дополнительно сюда может быть включено понятие защиты самих пользователей информационной сре-

ды от физического воздействия злоумышленников, а также защиты информации неvirtуального характера (твердых копий — распечаток, служебных телефонных справочников, домашних адресов сотрудников, испорченных внешних носителей и т. п.).

Компьютерная безопасность (сетевая безопасность, телекоммуникационная безопасность, безопасность данных) — обеспечение защиты информации в ее виртуальном виде. Возможно выделять этапы нахождения информации в среде, и по этим принципам разделять, например, компьютерную (на месте создания, сохранения или обработки информации) и сетевую (при пересылке) безопасность, но это, в принципе, нарушает комплексную картину безопасности. Единственное, с чем логично было бы согласиться, — это термин *безопасность данных*, или скорее, *безопасность данных в рамках данного приложения*. Дело в том, что в конкретном программном комплексе модель безопасности может быть реализована таким образом, что это потребует отдельного специалиста (или даже службы) по ее поддержанию. В этом случае возможно разделить понятия *безопасность данных* (конкретного приложения) и *безопасность сети* (всей остальной информационной среды).

Следующий этап предусматривает создание условий, чтобы мы говорили на одном языке, т. е. введение основного набора определений.

Глава 2



Терминология

Итак, для того, чтобы говорить на понятном всем языке, необходимо ввести ряд определений. Список преднамеренно несколько расширен, чем необходимо для данной книги, в том смысле что не все указанные здесь термины будут использоваться далее по тексту. С одной стороны, это сделано специально в помощь тем, кто будет в дальнейшем самостоятельно создавать нормативные, методические и иные материалы по информационной безопасности. Некоторые из этих документов, которые, возможно, будут носить сложный или технический характер, обязательно потребуются предварять разделом определений во избежание разночтений и искажения смысла. В этом случае можно будет взять за основу данный раздел и либо использовать определения, приведенные здесь, либо при необходимости модифицировать их.

Тщательная формулировка подобных определений особенно важна, когда организация, начинающая работу по развитию информационной безопасности, вступает в некие формализованные (например, договорные) отношения с субъектами, имеющими слабое представление о предмете (например, со своими клиентами, партнерами или поставщиками). Дело в том, что иногда даже простые и кажущиеся очевидными понятия, такие как *информация* или *пароль*, могут восприниматься по-разному и в дальнейшем стать предметами спора.

С другой стороны, читателю нет необходимости запоминать все указанные здесь термины для успешного освоения материала книги. По мере необходимости при дальнейшем использовании необходимые определения будут повторены, возможно, в более упрощенном варианте.

Информацией будем называть данные, представленные в том или ином виде, пригодном для хранения, обработки и/или передачи. Применительно к информационной безопасности с практической точки зрения удобно выделить следующие формы информации:

- ☐ электронная (электрические сигналы, области намагничивания и т. п.);
- ☐ печатная (распечатки, книги и т. п.);
- ☐ визуальная (изображения на экране монитора, слайды, плакаты и т. п.);
- ☐ аудиальная (разговор людей, звуковые автоинформаторы и т. п.).

Объектом будем называть собственно информационный ресурс, т. е. некий целостный набор информации, в который входят данные, объединенные общей темой, задачей, способом обработки и т. п.

Субъектом — пользователя информации или процесс, обрабатывающий данный набор информации (объект) и учитываемый, применительно к данному рассмотрению, в момент того или иного использования объекта.

Следовательно, объект — это некий пассивный, а субъект — активный участник процесса обмена информацией.

Информационная система — это некий потенциально открытый набор субъектов, воздействующих на набор объектов, причем субъекты одной системы обычно имеют общую цель или цели.

Информационное пространство — это совокупность информационных систем, взаимодействующих между собой, причем одна часть этих систем может иметь иные, в том числе прямо противоположные, интересы, чем другая.

Правила информационной безопасности — список разрешенных и/или запрещенных действий для субъектов информационной системы. Такие правила могут быть определены не только для человека (пользователя, администратора и т. д.), но и для процесса (например, процесса предоставления доступа на межсетевом экране).

Права (англ. *rights*) — набор разрешенных действий (правил) для данного субъекта, часть профиля субъекта.

Несанкционированным будем считать такое действие, которое производится субъектом (или частью субъекта), для которого данное действие не определено как разрешенное (либо определено как запрещенное) правилами информационной безопасности.

Средство работы с информацией — устройство информационного обеспечения, с помощью которого производится или обеспечивается работа в информационной системе.

Угроза — возможность реализации нарушения того или иного правила информационной безопасности.

Уязвимость (англ. *vulnerability*) — незащищенность или ошибка в объекте или информационной системе, которая приводит или может привести к возникновению угрозы.

Атака — практическая реализация угрозы или попытка ее реализации с использованием той или иной уязвимости.

Авария — незлоумышленное происшествие неординарного характера, несущее деструктивное воздействие на объект или информационную систему.

Пользователь — человек, субъект информационной системы, выполняющий в ней бизнес-функции, т. е. использующий объект в целях производства.

Администратор — человек, субъект информационной системы, создающий условия для работы в ней пользователей.

Контролер — субъект (не обязательно человек!) информационной системы, контролирующий использование информационной системы пользователями и администраторами в соответствии с предопределенными правилами (правилами информационной безопасности).

Злоумышленник — человек, субъект информационной системы, преследующий корыстные или деструктивные цели, противоречащие бизнес-целям системы.

Идентификатор (имя, логин) — набор символов, представляющий уникальное наименование данного объекта или субъекта в данной (в другой системе имя может повториться) информационной системе. Позволяет однозначно идентифицировать пользователя при входе его в систему, определить его права в ней, фиксировать действия и т. п.

Пароль (англ. password) — секретная последовательность символов, связанная с субъектом и известная только ему, позволяющая его аутентифицировать, т. е. подтвердить соответствие реальной сущности субъекта предъявляемому им при входе идентификатору.

Профиль (англ. profile) — набор установок и конфигураций, специфичный для данного субъекта или объекта и определяющий его работу в информационной системе.

Шифрование (англ. encryption) — процесс приведения информации в форму, при которой невозможно или существенно затруднено извлечение из нее осмысленных данных без обладания специфическими дополнительными знаниями (*ключом*).

Дешифрование (расшифрование) (англ. decryption) — процесс, обратный процессу шифрования, т. е. восстановление с помощью соответствующего ключа информации в исходной форме, позволяющей извлечь из нее смысловые данные.

Криптоанализ — набор методов и средств для выполнения (или сам процесс) дешифрования информации без обладания необходимым ключом.

Электронный документ — набор данных в электронном представлении, который может быть путем стандартных преобразований представлен в понятном человеку виде, в том числе как документ на бумаге; является неделимой единицей обмена информацией, т. е. он может быть передан (получен) либо не передан (не получен) только целиком; состоит из элементов, называемых реквизитами документа.

Закрытый ключ (приватный ключ, секретный ключ) — секретная последовательность байт, предназначенная для формирования субъектом электронной цифровой подписи для электронных документов.

Открытый ключ (публичный ключ) — последовательность байт, связанная с закрытым ключом и предназначенная для проверки электронно-цифровой подписи у электронных документов. Открытый ключ должен быть в распоряжении проверяющей стороны.

В этой главе умышленно дается не совсем полное определение закрытого и открытого ключей. Данные определения скорее наиболее прикладные, так как указанные ключи чаще всего используются для электронной цифровой подписи. Эти термины необходимо вводить в тех случаях, когда распределяется ответственность за сохранность секретных ключей и возникает необходимость своевременного представления открытых. Подробно использованию всех видов ключей, а также шифрованию и алгоритмам ЭЦП посвящена *часть V* данной книги.

Электронная цифровая подпись (ЭЦП) (англ. *digital signature*) — блок данных определенного формата, формируемый на основе закрытого ключа, находящегося в распоряжении только уполномоченного субъекта, однозначно сопоставляемый с конкретным электронным документом и используемый для проверки аутентичности (авторства) и целостности данного электронного документа.

Корректная электронная цифровая подпись — электронная цифровая подпись электронного документа, дающая положительный результат при ее проверке соответствующим открытым ключом.

Модель безопасности — набор принципов, схем и механизмов (иногда с указанием конкретных средств и инструментов), предназначенных для обеспечения информационной безопасности в данной конкретной информационной системе.

Ошибка — непреднамеренная, не планируемая заранее акция (в виде действия или бездействия), совершаемая субъектом (пользователем или процессом), которая представляет или может представлять угрозу информационной безопасности.

Зловредная программа — исполняемый программный модуль, скрипт, макрос или прочий программный код, созданный с целью нарушения информационной безопасности. Задачей зловредной программы ставится: минимально — несанкционированное использование части информационных ресурсов, максимально — приобретение полного контроля над объектом или информационной системой с целью ее дальнейшего несанкционированного использования. К зловредным программам относят вирусы и троянские программы.

Вирус — зловредная программа, основным свойством которой является возможность автоматического размножения (возможно, с самомодификацией)

и распространения на новые информационные системы без контроля со стороны создателя. Обычно основной целью вирусов является выполнение деструктивных действий, хотя встречаются вирусы, созданные для развлечения. Чаще всего никакой материальной выгоды создатель вируса от его функционирования не получает, но иногда, например, в рамках промышленной конкурентной борьбы, создание вирусов может быть оплачено заказчиком.

Троянская программа, троян, троянский конь (англ. *trojan*) — зловредная программа, выполняющая несанкционированные пользователем и недокументированные действия, порой даже с функциями удаленного управления со стороны злоумышленника, созданная с целью намеренного нарушения информационной безопасности данной системы. Часто при этом троянские программы маскируют несанкционированную деятельность выполнением ряда полезных для пользователя документированных действий.

С развитием сетевых технологий появились троянские программы, проникающие в систему по принципу компьютерных вирусов. Подобные гибриды используют в целях злоумышленника свойства и первого и второго класса зловредных программ. Наиболее сложные из них, будучи "выпущенными на свободу" где-либо в сети предприятия и размножаясь как обычный вирус, могут со временем добраться до своей основной цели, которая, возможно, хорошо защищена от прямых атак извне. После этого троян выполняет свою задачу по нарушению безопасности и информирует создателя о том, что его задача выполнена. Примитивные варианты троянских программ могут делать мелкие пакости типа рассылки от имени пользователя, неосторожно запустившего программу, угроз или ругательств другим пользователям сети.

Закладка (аппаратная, программная) — зловредная функциональность (программа), реализованная как одна из скрытых функций системы или объекта. В отличие от троянских программ, попадающих в систему извне, создается в подавляющем большинстве разработчиками системы. Дополнительно встречаемые термины — *"задняя дверь", люк* (англ. *trapdoor* — недокументированная функция, заложенная разработчиком при создании системы/объекта и не ликвидированная по каким-либо причинам, *логическая бомба* — срабатывание функции происходит при выполнении определенных условий).

Доверительное состояние системы — состояние, при котором поведение системы в плане информационной безопасности точно соответствует спецификации, в системе не функционируют зловредные программы, не было и не существует возможности несанкционированного доступа к данным на чтение и/или модификацию.

ПИН — персональный идентификационный номер, цифровой идентификатор, присваиваемый объекту или субъекту. Иногда используется как аналог пароля, например, в системах пластиковых карт.

Токен (англ. *token*) — в общем случае носитель данных, параметров безопасности своего владельца. Обычно некий внешний энергонезависимый носитель, предназначенный для хранения длинных паролей, криптографических ключей и другой информации по безопасности, которую человек не способен сохранять в своей памяти. Доступ к данным токена дополнительно может быть защищен паролем или ПИН;

Системный мусор (англ. *system garbage*) — набор данных, используемых системой или объектом для работы и не предназначенный для постороннего анализа, но по каким-либо причинам попавший в распоряжение третьих лиц. Представляет интерес для злоумышленников, так как может хранить данные об особенностях функционирования системы, косвенную информацию о конфиденциальных данных или даже пароли и криптографические ключи.

В связи с этим термином необходимо упомянуть понятие *дамп памяти* (англ. *memory dump*) — "снимок" состояния оперативной памяти устройства, используемый для выяснения причин сбоев при выходе системы из строя. Обычно создается и хранится в виде файла на дисковом носителе. В случае попадания к злоумышленникам может нанести очень серьезный урон безопасности системы.

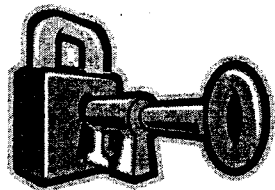
Постороннее программное обеспечение — программное обеспечение, не авторизованное к использованию в данном информационном пространстве и не зарегистрированное уполномоченными службами.

Ряд используемых далее терминов, напрямую не связанных с информационной безопасностью, перенесены в Глоссарий, с тем чтобы не загромождать данную главу слишком большим количеством информации.

Дополнительно необходимо отметить появление в тексте слов "организация", "корпорация", которые являются синонимами слова "предприятие" в смысле тематики данной книги.

Кроме того, будут встречаться термины "бизнес-процесс", "бизнес-логика". В данном случае слово "бизнес" означает "то, что предопределено целями производства данного предприятия".

Глава 3



Задачи и принципы организации службы информационной безопасности

В этой главе речь пойдет о необходимых бюрократических процессах формирования службы безопасности на предприятии, предназначенных, скорее, для руководителей службы или для тех, кто профессионально занимается стратегией и тактикой структурного развития организации. Поэтому читатели, более интересующиеся техническими аспектами, могут смело переходить к *главе 4*.

Для тех, кто все-таки решит прочитать эту главу, отметим, что решение вопросов и проблем, описываемых здесь, может оказаться полезным для предотвращения производственных конфликтов и несогласованностей, которые могут возникнуть в силу специфики функционирования службы информационной безопасности.

Задачи службы информационной безопасности



Несмотря на, казалось бы, прямую очевидность задач службы информационной безопасности ("служба безопасности должна обеспечивать безопасность — что же еще?"), возникает множество вопросов, не бросающихся в глаза на первый взгляд. Разобьем эти вопросы или проблемы на следующие группы:

- ☐ размещение службы информационной безопасности;
- ☐ взаимодействие ее с другими службами;
- ☐ иерархия подчинения.

Иначе говоря, следует определить, где в штатной структуре предприятия должна размещаться служба информационной безопасности, как она будет взаимодействовать с другими подразделениями (особенно с подразделением информационных технологий), сколько начальников должно стоять между руководителем службы безопасности и директором предприятия.

Существует ряд рекомендаций по вопросам размещения, взаимодействия и подчинения службы безопасности, как продвинутые, в основном западные, ориентированные на электронный мир, так и старые, еще советские, пришедшие от первых отделов. На наш взгляд, эти процедуры очень зависят от множества факторов конкретного предприятия, быть может даже таких, как сложившиеся неформальные взаимоотношения между сотрудниками. Если во главу угла на предприятии ставится эффективность работы службы информационной безопасности, то такие аспекты невозможно не учитывать.

Чтобы не очерчивать жестких рамок для всего этого, предлагаем просто рассмотреть проблемы и ответить для себя на возникающие вопросы. Из полученных ответов станет ясно, как и где должна располагаться служба информационной безопасности на организационном дереве предприятия.

Чем должна заниматься служба безопасности?

- ☐ Администрировать имеющиеся средства безопасности (межсетевые экраны, антивирусные пакеты, системы обнаружения атак и пр.)?
- ☐ Разрабатывать модели и схемы защиты информации, принимать решения о приобретении новых средств безопасности?
- ☐ Контролировать работу пользователей информационного пространства предприятия? Каких групп — только конечных пользователей или также и администраторов систем?
- ☐ Куда направлен основной фокус внимания службы — на внутренних пользователей (по статистике больше всего нарушений информационной безопасности — как умышленно, так и неумышленно — происходит изнутри предприятия) или на защиту от доступа из внешнего информационного пространства?

Без сомнения, вопросы информационной безопасности должны учитываться в каждом из перечисленных пунктов. Однако прикладное применение может быть различным и зависеть от множества причин, которые обычно не предусмотрены западными рекомендациями, например таких как нехватка квалифицированного персонала.

Примером различных решений могут быть две следующие ситуации: служба информационной безопасности только производит анализ ситуации, разработку модели и принимает решение о необходимости приобретения средства защиты, а его администрирование производится в рамках обычной работы подразделения информационных технологий. Противоположный вариант — приобретение средств защиты производится в рамках общей стратегии развития информационных технологий предприятия, а конкретные работы по установке и поддержке средства защиты производит служба информационной безопасности.

Как происходит взаимодействие со службой информационных технологий, а конкретно с администраторами сетей и систем?

- ❑ Специалисты информационной безопасности имеют полный контроль над информационной системой, равный правам администратора системы.
- ❑ Специалисты по безопасности принимают частичное участие в администрировании системы, например в настройке прав пользователей.
- ❑ Специалисты по безопасности имеют доступ в информационной системе ко всем объектам, но имеют право только читать сведения о них.
- ❑ Специалисты информационной безопасности не имеют доступа в систему, используют для контроля работы администраторов регистрационные журналы, конфигурационные отчеты и т. п.

Все эти варианты указаны преднамеренно, так как вопрос взаимоотношений службы безопасности и, скажем, администратора локальной сети может быть очень напряженным, особенно если служба только создается, а администратор уже несколько лет выполнял свои функции. Как быть, если администратор действительно честно работает долгое время, а специалисты по безопасности только пришли в организацию, но требуют значительных прав в системе для себя и ограничения прав администратора? Что делать, если администратор в этом случае решил покинуть организацию? А что если квалификация специалистов по безопасности по конкретной информационной системе значительно ниже, чем у администратора, и они могут, при наличии определенных прав, внести помехи в работу системы? Что если система устроена таким образом, что для того, чтобы контролировать администратора, необходим полный контроль над всей системой? Вопросов больше, чем ответов, решение необходимо принимать с учетом всех этих проблем.

Возможно, болезненный процесс передачи или разделения прав придется растянуть на продолжительный период, пока специалисты службы безопасности не приобретут соответствующий опыт, а администраторы постепенно не привыкнут к частичному, а затем и полному контролю.

Сколько ступеней принятия решения по вопросам информационной безопасности должно существовать?

- ❑ Если руководитель службы информационной безопасности непосредственно подчинен директору предприятия и вносит проекты решений напрямую, то это дает возможность злоупотребления своим положением (так как топ-менеджер скорее всего не очень компетентен в информационных технологиях, но на слово "безопасность" реагирует немедленным визированием всех документов).

- ❑ Если процесс согласования решения чересчур распределен по руководителям и растянут во времени, есть риск, что произойдет запаздывание в принятии жизненно важного решения (впрочем, как и для любой другой службы). Кроме того, если представители точек согласования не компетентны в вопросе безопасности, то много времени непродуктивно будет потрачено на выяснение проблем и согласование мнений.
- ❑ Согласован ли процесс принятия решения по информационной безопасности с общей стратегией развития информационных технологий? Не будут ли в информационном пространстве предприятия установлены средства защиты, которые внесут помехи в работу других информационных систем?

В данном случае существуют конкретные рекомендации, которые советуют подчинять службу информационной безопасности не напрямую директору, а некоему комитету по безопасности, в котором будут присутствовать, с одной стороны, специалисты, способные оценить качество предлагаемого решения, с другой — руководители, способные утвердить принятое решение к обязательному исполнению.

Сфера охвата информационных систем



Для полного контроля ситуации в информационном пространстве служба информационной безопасности должна иметь необходимые знания обо всех информационных системах, объектах и субъектах. Однако как быть в следующих ситуациях.

- ❑ Данная конкретная информационная система для своего функционирования уже предусматривает разделение административных функций между двумя персонами, контролирующими друг друга.
- ❑ Функции данной информационной системы, с одной стороны, достаточно специфичны и требуют серьезной подготовки для ее администрирования и/или контроля, а, с другой стороны, слишком незначительны в рамках предприятия для того, чтобы содержать (обучать) двух администраторов для ее контроля.
- ❑ Информационная система имеет вид "черного ящика": все работы по ее администрированию выполняются внешним провайдером, который не может допустить посторонних к этому процессу.

Если в третьей ситуации возможна организация соответствующего распределения ответственности в рамках контракта на поставку услуг, то с первыми двумя случаями дело обстоит сложнее. Возможно, придется назначить отдельного специалиста по безопасности данной информационной системы из имеющихся администраторов и подчинить его (полностью или частично)

службе информационной безопасности или комитету, о котором шла речь выше. Возможно, придется выделить понятия "прикладной безопасности", оставив ее в ведении службы, администрирующей данную систему, и "системной безопасности" (то есть безопасности взаимодействия данной системы с остальными), которую отдать общей службе информационной безопасности. Возможно, будет достаточно предоставить службе информационной безопасности доступ к упомянутым электронным регистрационным журналам, файлам конфигураций, учету пользователей и т. д.

Однако для всех трех ситуаций мы бы рекомендовали разработать процедуру отчета администраторов систем перед службой информационной безопасности. Детализацию и периодичность отчетов можно согласовывать дополнительно, в зависимости от конкретной системы, но службе безопасности следует иметь полную картину работы всех систем информационного пространства.

Другая проблема, приводящая иногда к выпадению отдельных аспектов из сферы деятельности службы информационной безопасности, — это представление о ней только как о компьютерной службе. Это допустимо, но тогда необходимо подумать о следующем.

□ Кто в организации курирует вопросы безопасности телефонной связи (сотовые телефоны, мини-АТС, факсимильную связь)? Безопасно ли уничтожение бумажных отчетов предприятия? Как защищены источники электроэнергии и т. п.?

Если всем этим занимаются отдельные службы, то следует также продумать вопросы связи их со службой информационной безопасности. Если же есть необходимость передать контроль над ними в службу информационной безопасности, то следует добавить и необходимое число сотрудников, провести их соответствующее обучение.

Сфера охвата персонала



Казалось бы, чего проще — сфера действия информационной безопасности, в том числе и правила политики безопасности, едины для всех. И это справедливо. Вот только для всех ли эти правила должны быть одинаковы? Например, нормальным требованием безопасности является запрет для пользователей загружать из Интернета исполняемые модули. А для специалистов по информационным технологиям? Откуда им брать обновления и заплатки для информационных систем? Очевидно, что правила должны быть дифференцированными для таких случаев.

Другой пример — доступ к письмам электронной почты. Вопрос о том, можно ли уполномоченным сотрудникам службы безопасности просматривать электронную почту работников предприятия, продолжает обсуждаться.

Следует учесть, что если организация предоставляет своим сотрудникам определенные ресурсы, она должна иметь возможность контроля их использования. Допустим, этот вопрос решен положительно, и работники службы безопасности имеют право просматривать электронную почту пользователей. Означает ли это, что они имеют право следить за перепиской директора и его заместителей? А за почтой своего собственного начальника? Чтобы не нарушать общие правила такими негласными исключениями, следует заранее оговорить сферу применения правил безопасности.

Еще один важный вопрос, возникающий в связи с персоналом организации, — это осведомленность службы безопасности о текущем статусе каждого из работников. В идеале служба безопасности должна принимать участие в приеме на работу (если не в принятии решения о приеме, то в инструктаже по нормативным документам и правилам), увольнении с должности (подписании обходного листа), а также периодически получать информацию об отпусках, командировках, болезнях и прочих данных.

Сфера охвата проектов



Мы уже рассматривали вопрос о необходимости согласования решения по информационной безопасности с общим направлением развития информационных технологий. Но должно быть верно и обратное. Тот, кто начинал развивать службу информационной безопасности, наверняка сталкивался с ситуацией, когда полгода назад была внедрена информационная система, но оказалось, что работники данной организации могут ее использовать во вред предприятию. Что же делать?

Очевидно, что теперь, когда вложены средства на приобретение и установку системы, никто не захочет ее заменять. А что же делать, если в ней не предусмотрено модели безопасности необходимого уровня? Придется окружать ее дополнительными системами, вырабатывать сложные организационные процедуры и многое другое. Все потому, что служба информационной безопасности не была привлечена к работе над проектом по выбору и внедрению системы на раннем этапе.

Таким образом, мы рассмотрели ряд проблем, с которыми можно столкнуться в процессе формирования или реформирования службы информационной безопасности. Скорее всего в каждом конкретном случае в этот список добавится еще большое число вопросов, специфичных для данного предприятия. От их решения зависят и принципы формирования службы. При этом следует помнить, что каждая новая задача, возложенная на службу безопасности, потребует, как и в любом другом случае, дополнительных людских ресурсов, расходов на обучение, других операционных расходов.

Оценка эффективности службы информационной безопасности



Дополнительный интересный вопрос, который может быть поднят в связи с деятельностью службы информационной безопасности, — это ее эффективность.

Рано или поздно этот вопрос в большей или меньшей степени должен заинтересовать каждого руководителя, которому подотчетен менеджер службы информационной безопасности.

Если, например, возможно представить доказательства, что службой безопасности было предотвращено десять атак с потенциальной угрозой ущерба по 20 тысяч долларов каждая, то все просто. А если не происходит никаких инцидентов — это показатель хорошей или плохой работы службы?

Один из способов оценки результативности службы основан на технологии управления рисками, которая будет рассмотрена ниже. Другой способ (который может быть применен, впрочем, не только для службы информационной безопасности, но и для любой ИТ-службы) предложен General Services Administration's (GSA) Office of Governmentwide Policy (США) (Бюро правительственной политики по управлению общими службами) в книге "Восемь шагов к эффективному развитию и использованию средств измерения производительности в информационных технологиях" (Eight Steps To Develop and Use Information Technology Performance Measures Effectively). Выдержки из этого документа представлены в разд. *"Восемь шагов к эффективному развитию и использованию средств измерения производительности в информационных технологиях (ИТ)"* этой главы.

Контроль функционирования службы



Тем, кто собирается контролировать работу службы информационной безопасности, следует в первую очередь ознакомиться с организационной структурой предприятия и местом в ней службы безопасности. Далее следует опираться на имеющийся опыт проверки работы других предприятий и подразделений либо на логическое мышление. В конце концов, все организационные схемы, в том числе и оптимальные, придумывают люди, и человек с аналитическим складом ума может разобраться в их принципах.

Если при проведении проверки возникают вопросы в целесообразности организации работы именно сложившимся способом, необходимо получить разъяснения, на основании чего было принято решение именно такое. Вполне возможно, что это обусловлено спецификой работы предприятия.

В разделах данной главы для специалистов был обозначен круг вопросов, которые следует определить при формировании службы, повторим их кратко в форме, необходимой для проведения проверки:

- ☐ в рамках какой структуры функционирует служба информационной безопасности;
- ☐ кому подчиняется служба и как происходит принятие решений;
- ☐ как соотносятся друг с другом служба информационной безопасности и подразделение информационных технологий;
- ☐ к каким информационным системам имеет доступ служба безопасности и с какими правами, как эти права соотносены с правами администратора системы;
- ☐ каковы правила информационной безопасности для разных категорий пользователей;
- ☐ принимает ли участие служба безопасности в разработке проектов по новым информационным системам;
- ☐ как контролируется статус работника (принят на работу, уволен, болен, в отпуске, в командировке и т. п.).

При ответах на эти вопросы необходимо выяснить, почему было сделано именно так, а не иначе и как можно оптимизировать функционирование службы.

Восемь шагов к эффективному развитию и использованию средств измерения производительности в информационных технологиях (ИТ)



Шаг 1. Привязать ИТ-проекты к целям и задачам предприятия

Эффективное измерение вклада ИТ в общие достижения предприятия начинается на стадии планирования и основывается на главной задаче организации и стратегическом бизнес-плане. Связь ИТ-проектов с целями и задачами предприятия может быть установлена с помощью норматива "Сбалансированный учет" (Balanced Scorecard), состоящего из четырех направлений: финансового, работы с заказчиком, внутреннего производства, инноваций и обучения.

Шаг 2. Разработать средство измерения производительности

Выбирается ограниченный набор значимых мер по кратко- и долгосрочным целям. Результаты ИТ-инвестиций выражаются не только в стоимости, своевременности и качестве. Результат — это эффект, который оказали ИТ-инвестиции на организацию. Примерами могут служить измеряемые улучшения в качестве и в доставке продуктов и услуг организации до клиента. Для конкретизации мер надо определить задачи проекта, решить, за счет чего будет достигаться выполнение требований, знать, как влияет результат на последующее развитие. Эффективность их обычно отражает сфокусированность на клиенте.

Шаг 3. Установить основу для сравнения с изменением производительности в будущем

Данная основа должна определить, улучшилась или ухудшилась производительность в результате вложений в ИТ. Эта основа должна быть задокументирована, рассмотрена и утверждена заказчиками и организаторами работ. Обычные отчеты могут служить основой, если они соотнесены с предопределенным индикатором производительности.

Шаг 4. Выбрать наиболее значимые ИТ-проекты

Поскольку организация может финансировать ограниченное число ИТ-проектов, надо выбрать те, которые наиболее значимы. Выбор основывается на оцениваемой экономической прибыли от ИТ-инвестиций плюс предполагаемый вклад в бизнес-приоритеты организации. Для оценки значимости и риска каждого вложения создается Совет по анализу инвестиций (Investment Review Board).

Шаг 5. Собирать данные

На шагах 2 и 3 необходимо задуматься, какие данные необходимы для определения результата и эффективности проекта. Выбор пакета данных зависит от того, что имеется в распоряжении, от затрат на их получение и актуальности их значений.

Шаг 6. Анализировать результаты

После того как результаты получены, необходимо их рассмотреть, чтобы определить, соответствует ли выполнение проекта задачам и адекватно ли индикаторы отражают результаты. Основной вопрос в том, отличаются ли результаты от того, что ожидалось. Во время анализа необходимо произво-

дить поиск способов для улучшения производительности, уточнения индикаторов и приобретения опыта. Хороший отчет о производительности отслеживает результат во времени и определяет тенденции.

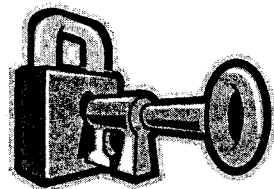
Шаг 7. Объединяться с процессом управления

Для того чтобы быть уверенным, что результаты улучшают производительность, необходимо оценивать их с точки зрения текущего процесса управления. Ведь если результаты не будут использоваться предприятием, то и процесс их измерения не нужен. Если выполнение проекта растянуто на несколько лет, то определять результаты нужно ежегодно.

Шаг 8. Сообщать о результатах

Необходимо информировать работников и руководителей о результатах внутри предприятия для улучшения скоординированности и целеустремленности. Если требуется поддержка и дополнительное финансирование проекта, то о результатах надо сообщать в вышестоящие инстанции. Для укрепления партнерских отношений о результатах информируют клиентов.

Глава 4



Создание службы безопасности

Критерии необходимости создания службы



Нужна или не нужна служба информационной безопасности? Как определить время, когда необходимо ответить на этот вопрос положительно (если такая служба еще не создана)? Это зависит от множества причин и условий, но первая стадия может определяться исходя из комбинации следующих пунктов:

- ☐ в вашей организации уже больше 10 компьютеров, распределенных по различным помещениям;
- ☐ в вашей организации создана локальная сеть;
- ☐ один из компьютеров вашей организации соединен с модемом;
- ☐ вы храните/обрабатываете на компьютере информацию, разглашение или утеря которой может принести вашей организации существенный ущерб.

Почему любой из перечисленных пунктов требует внимания?

1. Если у вас достаточно много компьютеров и к ним имеет доступ значительное число обслуживающего персонала, то необходимо задуматься о сохранности вашего оборудования. Пусть у вас еще нет даже локальной компьютерной сети и в компьютерах нет никакой существенной информации — сами по себе они представляют значительную ценность. Порой, рассуждая о вопросах информационной безопасности, специалисты упускают из виду такой ее аспект, как контроль физического доступа к информационным ресурсам. Между тем, по существующим оценкам, соотношение масса/цена у ряда компьютерных составляющих (например, у плат оперативной памяти) сравнимы с соотношением масса/цена золота. А это означает, что они сами по себе представляют мишень для злоумышленников.

Продолжая пример с платами оперативной памяти, можно привести типовой сценарий преступления — из 128 или 256 Мбайт оперативной памяти

похищается половина (или четверть). А теперь задумайтесь, все ли ваши сотрудники внимательно следят за процессом загрузки при включении компьютера и способны определить, что оперативная память сократилась? Каждый ли из них немедленно сообщит, если компьютер стал медленнее работать? А может, у вас есть компьютеры, которые никто не включает по несколько дней? Через какой срок тогда обнаружится пропажа?

2. Если у вас появилась локальная сеть, это означает, что вы начали использовать компьютеры в производственном процессе, даже если ваши сотрудники всего лишь обмениваются файлами через совместно используемые каталоги. Предположим, что даже при этом никакой существенной информации в вашей сети не присутствует. Однако, как вы будете себя чувствовать, если обнаружите, что утром очередного рабочего дня ни один из ваших компьютеров не сможет загрузиться из-за ошибки сбоя в операционной системе, так как накануне вечером один из ваших сотрудников запустил на своем компьютере игру с пиратского диска? Где в таком случае взять пусть не очень важный, но очень обширный отчет, который вы в течение последней недели готовили в Microsoft Word?
3. У вас появился модем — его сразу же попробуют подключить к одному из компьютеров. Ну и что, скажете вы, телефонная сеть — это еще не Интернет с его злобными хакерами. Но, может быть, недавно один из ваших сотрудников купил своему сыну (который знает папины рабочие телефоны) домашний компьютер с модемом, а у того есть диск с программами, которые скачал с сервера <http://hackzone.ru/>, и много свободного времени. Не обнаружите ли вы в один из дней ситуацию, аналогичную описанной в предыдущем пункте?
4. Вы храните в своем компьютере (даже не включенном в общую локальную сеть) конфиденциальную информацию и считаете, что она защищена, потому что у вас стоит пароль на включение компьютера, а сам компьютер находится в вашем персональном кабинете. А пароль — это, случайно, не номер вашего домашнего телефона? И всегда ли вы присутствуете в кабинете, когда производится уборка помещений?

Если вы задумались над этими вопросами, значит, вы готовы к созданию соответствующей службы или найму сотрудника, которому надо их переадресовать. А, может, этот специалист и есть вы? Тогда эта книга для вас.

Состав службы информационной безопасности



Как создать такую службу "с нуля" и кто должен входить в ее состав?

Конечно, ответы на эти вопросы также зависят от самой организации, ее целей, от условий, в которых она существует, и многих других факторов. В качестве модели будет рассматриваться достаточно крупное предприятие, в котором описываемые задачи и функции могут быть распределены между

различными сотрудниками службы информационной безопасности. Для более скромных компаний различные роли могут быть совмещены меньшим количеством сотрудников или даже одним.

Итак, состав службы. Нужен собственно *руководитель службы* — тот, кто будет определять ее общую стратегию и тактику, отчитываться руководству, принимать оперативные решения и нести за них ответственность. Это должен быть достаточно подготовленный специалист, так как ему необходимо следующее.

1. Разбираться в общих чертах в технических особенностях используемого информационного обеспечения (включая аппаратное и программное обеспечение, принятые де-юре и де-факто методы работы в информационном пространстве организации и т. п.). Иначе он просто не будет понимать, чем занимаются его подчиненные, а те в свою очередь смогут манипулировать своим руководителем.
2. Осуществлять надзорные функции как формализованные, так и нет, в том числе управление проектами, так как внедрение многих механизмов безопасности (например, таких как приобретение и внедрение технических решений) потребует проектной работы.
3. Разбираться в психологии работников, уметь разрешать конфликты (так как служба безопасности часто сама служит основой для репрессивного или ограничительного воздействия). Возможно, придется использовать слабые или сильные стороны различных сотрудников организации.
4. Знать основы существующего законодательства, так как, возможно, придется производить расследования, в которых будут фигурировать такие понятия, как "право частной собственности на информацию", "улики и доказательства", "свидетельства" и т. п.
5. Налаживать связи как с коллегами в других организациях, так и с вышестоящими организациями для защиты интересов своей компании в соответствующей сфере.
6. Пользоваться доверием руководства, так как при определенном желании можно создать такую ситуацию, когда на службу безопасности будет замкнуто очень многое, что даст возможность ее руководителю использовать свой пост в личных интересах.

Подчиненные описанного выше "идеального" руководителя будут специалистами, подбор которых также возможно осуществлять по различным критериям. В первую очередь, следует учесть характер функций, которые им предстоит выполнять.

□ *Операционный.* Сюда входит выполнение ежедневных процедур по мониторингу сети и отдельных сервисов (приложений), инструктаж и регист-

рация пользователей, контроль выполнения процедур (например, резервного копирования) и т. п. Можно добавить к перечисленному также проведение расследований.

- ❑ *Исследовательский.* Сюда входит изучение текущей обстановки в информационном пространстве как ближайшем, так и общемировом, анализ новых возможностей и уязвимостей. Возможно, здесь будет производиться выбор, разработка и внедрение новых технологий.
- ❑ *Методический.* Данный набор функций является связующим между первыми двумя. Завершенный проект по внедрению технологии необходимо поставить на рельсы ежедневного рутинного функционирования, т. е. обеспечить соответствующими процедурами и порядками, нормативной и технической документацией и т. п. Эта же группа может производить анализ рисков.

Теперь, когда руководитель и специалисты подобраны, можно ли начинать работать? Не можно, а нужно, только это еще будет не работа, а подготовка к ней.

Ведь вам предстоит определить:

- ❑ стратегию и тактику вашей защиты;
- ❑ что, от чего и как вы будете защищать;
- ❑ необходимые средства для этого (в том числе и финансовые);
- ❑ обеспечить необходимое нормативно-правовое пространство.

Уровень подготовки специалистов



Итак, вы — не новичок в области информационной безопасности. Вы уже решили, что такая служба нужна и, возможно, уже наметили первые шаги по планированию работы. Как же определить, достаточно ли вы и ваши сотрудники подготовлены для выполнения возложенной на вас миссии?

Можно рекомендовать обратиться за опытом к профессионалам — службам обеспечения квалифицированным персоналом. Достаточно посетить несколько крупных агентств по трудоустройству (например, их интернет-сайты) с ключевым запросом "information security", и в списках требований к кандидатам часто можно увидеть сочетание "CISSP preferred". Что оно означает?

Существует специальная организация International Information Systems Security Certification Consortium, Inc. (Международный консорциум по сертифи-

кации в области информационной безопасности), которая занимается тестированием и сертификацией специалистов по информационной безопасности, а CISSP — это ни что иное, как Certified Information Systems Security Professional (Сертифицированный специалист в области информационной безопасности). Другая категория, по которой проводится сертификация, — это SSCP — Systems Security Certified Practitioner (Сертифицированный практикующий специалист по безопасности систем). Если вы, читатель, уже имеете подобный сертификат, то примите наше уважение и можете лишь бегло проглядеть эту книгу — большинство из написанного здесь вам уже известно.

Если же вы еще ни разу не пытались произвести независимую оценку своих способностей, то вас должен заинтересовать список требований компании к сертифицируемым кандидатам. По дополнительному запросу (по адресу <http://www.isc2.org/>), вам по электронной почте бесплатно вышлют список тематических вопросов для подготовки.

Не нарушая авторские права компании на разработку данного материала, рассмотрим кратко список тем, по которым производится тестирование:

- ☐ системы контроля доступа, методология;
- ☐ телекоммуникации и сетевая безопасность;
- ☐ практическое управление безопасностью;
- ☐ безопасность разработки приложений и систем;
- ☐ криптография;
- ☐ архитектура и модели безопасности;
- ☐ безопасность операций;
- ☐ планирование продолжения производственной деятельности и восстановления после аварий;
- ☐ законность в проведении расследований и этика;
- ☐ физическая безопасность.

Если большинство этих тем вам знакомы и вы считаете, что достаточно подготовлены в них, тогда давайте углубимся немного в конкретные вопросы представленных разделов:

- ☐ в чем отличие контроля доступа, основанного на правилах (rule-based) и основанного на ролях (role-based);
- ☐ что представляют собой модели контроля доступа Bell-LaPadula, Biba и Clark-Wilson;
- ☐ в чем преимущества и недостатки протоколов контроля доступа RADIUS и TACACS;

- ☐ что такое "атака методом грубой силы" (brute force) или "злоумышленник-посредник" (man-in-the-middle);
- ☐ какие способы аутентификации и конфиденциальности поддерживает технология IPSec;
- ☐ чем отличаются методы аутентификации PAP и CHAP;
- ☐ как вычисляются среднегодовые ожидаемые убытки (Annual Loss Expectancy);
- ☐ чем обеспечивается неотрекаемость (non-repudiation);
- ☐ что такое "восстановление ключей" (key recovery) и "слабые ключи";
- ☐ что такое "демилитаризованная зона" (DMZ) и как она используется;
- ☐ какая из систем биометрического контроля доступа обеспечивает наибольшую точность?

На большинство таких вопросов мы постараемся ответить в этой книге, для оставшихся — укажем источники информации и направления поиска.

Ряд других компаний, занимающихся информационной безопасностью (например, компания-производитель известных межсетевых экранов — CheckPoint Inc.), также предлагают сертификацию специалистов как по своим конкретным продуктам, так и по общим направлениям безопасности. Не так давно компания ICSA (<http://www.trusecure.com/>) анонсировала программу сертификации специалистов по безопасности, в которую входят следующие категории: ICSA (The ICSA Certified Security Associate — Сертифицированный специалист по безопасности ICSA), ICSE (The ICSA Certified Security Expert — Сертифицированный эксперт по безопасности ICSA) и ICSP (The ICSA Certified Security Professional — Сертифицированный профессионал по безопасности ICSA).

Получение сертификации CISSP или других известных компаний, без сомнения, показывает высокий уровень подготовки специалиста, однако он приемлем не для всех из-за стоимости самого экзамена и расходов на поездку до учебного центра.

К счастью, существуют альтернативные методы подтверждения квалификации специалиста по информационной безопасности (как, впрочем, и по другим информационным технологиям и ряду других направлений). Это широко известная компания по он-лайн тестированию, обслуживающая более трех миллионов пользователей по всему миру — Brainbench Inc. (бывшая Tekmetrics).

Проверить специалиста по безопасности можно по следующим экзаменам, предлагаемым компанией на 1 сентября 2002 года:

- ☐ Internet security (проверяются знания как основ, так и конкретных методов обеспечения безопасности в TCP/IP сетях);

- ☐ Network security (экзамен интересен тем, что требует специальных знаний в конкретных областях, таких как Unix, Windows-системы, Solaris, Novell, Cisco IOS и т. п.);
- ☐ Disaster recovery and planning (проверяет способности к организации аварийного плана действий);
- ☐ TCP/IP administration (проверяет глубину знаний в архитектуре протоколов TCP/IP);
- ☐ NT/Win2000/Unix/Linux administration (проверяет уровень знаний в конкретной операционной системе);
- ☐ Network/Internet/Telecommunication concepts (проверяет знания в соответствующей области).

Есть ряд экзаменов по конкретным направлениям безопасности, таким как:

- ☐ Check Point Firewall-1 administration (экзамен по широко известному межсетевому экрану);
- ☐ MS Proxy 2.0 administration (экзамен по межсетевому экрану производства компании Microsoft);
- ☐ VPN with Windows (экзамен по использованию PPTP и L2TP в виртуальных частных сетях);
- ☐ Security Industry (здесь можно проверить свои знания по обеспечению физической безопасности).

Кроме того, на наш взгляд, руководителю службы безопасности будет полезно сдать следующие экзамены:

- ☐ Project management (проверяются способности возглавить команду по выполнению проекта);
- ☐ Technical writing (проверяются способности к написанию документации и отчетов);
- ☐ E-commerce concept (организация бизнес-деятельности с использованием информационных технологий);
- ☐ Object-oriented development (полезен для проверки специалиста как потенциального участника команды по постановке задачи на создание или тестирование имеющегося приложения);
- ☐ RDBMS development (экзамен используется для проверки знаний по основам работы с СУБД).

В стоимость сдачи экзамена включается выпуск и почтовая доставка сертификата. Экзамен сдается по двум уровням, в зависимости от результата —

обычный (испытуемый получает более 2,75 балла) и мастер-уровень (при прохождении экзамена с результатом более 4 баллов из 5 возможных).

Данный метод сертификации выгодно отличает возможность прохождения обучения и экзамена без выезда в учебный центр, используя только выход в Интернет и удаленное средство оплаты (пластиковую карту).

Дополнительные подробности можно узнать по адресу <http://www.brainbench.com/>.

Подбор кадров: теория или практика?

При всех положительных свойствах сертификатов хотелось бы предостеречь от слепого доверия к их наличию. К сожалению, авторам известны случаи, когда экзамены на право обладания самыми знаменитыми сертификатами за деньги сдавали специалисты более высокого уровня, чем потенциальные владельцы. Кроме того, и честные владельцы именитых сертификатов не всегда могли показать серьезного уровня знаний в решении практических, повседневных проблем, например, системной поддержки локальной компьютерной сети.

Таким образом, можно утверждать, что наличие у специалиста сертификата означает, что он, как минимум, один раз разобрался в теоретических аспектах темы настолько, что смог сдать указанный экзамен. Это дает возможность предположить, что при наличии необходимой литературы, времени и технической базы он сможет подойти к решению практического вопроса лучше, чем другой неподготовленный специалист. Однако подобные теоретические знания отличаются от практической деятельности, которая, например, включает способность понять сущность проблемы при общении с неквалифицированным пользователем, который с трудом может объяснить симптомы ситуации.

Иначе говоря, достаточным критерием оценки может служить только наличие двух факторов:

1. Сертификата, как подтверждения теоретических знаний в данной области.
2. Документа о практической работе в должности, соответствующей интересующей области.

В целом, в этой главе очерчена обширная область знаний, необходимая команде по информационной безопасности для успешного выполнения своих обязанностей. В последующих частях книги будет предпринята попытка раскрыть некоторые наиболее важные вопросы. Кроме того, дополнительные сведения об обучении в области информационной безопасности можно

получить из документа "Information Technology Security Training Requirements: A Role- and Performance-Based Model", NIST Special Publication 800-16.

Контроль процесса подбора кадров



Если вас не очень интересуют все внутренние тонкости "кухни" обеспечения информационной безопасности, однако, вы хотите быть уверенными, что она тем не менее находится на должном уровне (например, вы — внешний аудитор для данной организации), можно порекомендовать следующую последовательность действий.

Во-первых, вам следует ознакомиться с письменными документами по информационной безопасности, которые присутствуют в организации. Какие конкретно документы должны входить в этот список, перечислено в *главе 27*, однако, общий совет может быть таким.

- ❑ Ознакомьтесь со всеми имеющимися документами с точки зрения обычного пользователя, умеющего обращаться с графическим интерфейсом своего десктопа и знающего одно-два приложения типа текстового редактора или интернет-браузера. Оцените, какие плюсы вам принесло, как пользователю, ознакомление с этими документами.
- ❑ Если позволяет квалификация, ознакомьтесь с документами с точки зрения продвинутого системного администратора и оцените, как вы можете использовать пробелы и ошибки в этих документах для своих целей.

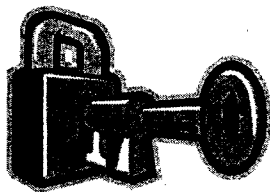
Во-вторых, ознакомьтесь с уровнем знаний специалистов информационной безопасности (или хотя бы руководителя службы), изучив соответствующие сертификаты (с учетом их срока действия), а также историю прохождения обучения и перспективный план.

Далее, вам необходимо узнать, какие операционные системы используются в организации и какие основные приложения применяются для критически значимых производственных направлений. После этого вам необходимо получить (по возможности независимо от специалистов самой организации) *проверочные анкеты по безопасности* (англ. *security checklist*) для каждой операционной системы и приложения (большинство наиболее свежих вариантов этих анкет всегда доступно в Интернете). С помощью специалистов организации или, если имеется возможность, без них убедитесь, что соответствующие пункты анкет для системы исполнены. Если же нет — узнайте, почему так произошло.

Самые каверзные контролеры могут зайти в Интернет на сайты с последними известными *уязвимостями* в системах, прочитать их описания и узнать об имеющихся *заплатках* (англ. *fix* или *patch*) на эти уязвимости. При про-

верке систем необходимо проверить, были ли установлены на нее заплатки в соответствии с новыми обнаруженными уязвимостями. Наиболее известные из них — <http://www.cert.org/> , <http://www.securityfocus.com/> .

Есть и другой путь — предпринять (возможно, с помощью стороннего специалиста) тестовое вторжение в информационное пространство организации. Но это сложный способ не только технически (то есть то, что вы не смогли проникнуть внутрь, еще не означает, что уровень защиты хорош — быть может, просто квалификация взломщика невысока), но и с точки зрения законности (если это действительно тестовое, а не реальное вторжение, то необходимо заручиться согласием самой организации, причем в письменной форме с указанием способов разрешения всех возможных конфликтов в этой связи).



Часть II

Что защищать

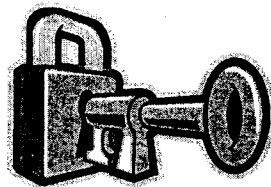
Глава 5. Инвентаризация информационных систем

Глава 6. Классификация информационных систем

Глава 7. Роли и ответственность субъектов

Глава 8. Модель информационных потоков

Глава 5



Инвентаризация информационных систем

Вообще говоря, данная тема должна рассматриваться в *части IV*, где будут освещены вопросы построения моделей защиты, так как все это относится к области информационной безопасности, называемой "Анализ и управление рисками". Однако мы разделили *части II* и *IV* описанием различных атак, так как, на наш взгляд, это облегчит читателю изучение темы, связанной с рисками, поскольку он будет иметь представления о конкретных, а не абстрактных угрозах информационным системам. Поэтому не удивляйтесь, изучая в дальнейшем другую литературу по информационной безопасности, что инвентаризация и классификация объектов и субъектов, возможно, будут представлены как первая часть оценки рисков.

Итак, теперь мы представляем, что такое информационная безопасность, пришли к единому языку и терминологии, создали команду, с которой будем работать. Прежде чем начинать строить защиту информационного пространства, необходимо перейти от абстрактных понятий "объект/субъект" к конкретным информационным системам или, проще говоря, ответить на вопрос: а что мы будем защищать?

Общий характер инвентаризации информационных систем



Инвентаризация — в данном случае это составление списка систем, т. е. объектов, которые будут подлежать защите и субъектов, которые задействованы в данном информационном пространстве, и будут влиять на информационную защиту системы. При этом необходимо не просто составить список, а указать ряд особенностей той или иной системы, т. е. кратко описать ее с точки зрения информационной безопасности. Чем подробнее сделать это на начальном этапе, тем легче будет дальше производить уточнения и строить окончательную модель защиты.

Данная работа обычно инициируется службой информационной безопасности, но выполняется обычно с привлечением специалистов других служб.

Это происходит потому, что специалисты по безопасности скорее всего не обладают полным видением модели и способов функционирования конкретного объекта или информационной системы, таким, каким обладает администратор системы или ее активные пользователи.

Способы проведения такой инвентаризации могут быть различными, но мы бы рекомендовали следующий. На первом этапе уполномоченный специалист службы информационной безопасности составляет, при необходимости консультируясь с подразделением информационных технологий, общий список объектов/систем и связанных с ними субъектов. Затем в этот список вносятся первичные характеристики объектов с целью описать их именно с точки зрения информационной безопасности. На следующем этапе начинается работа с администраторами (если таковые есть), пользователями и/или бизнес-менеджерами объектов и систем. В рамках заданных специалистом по безопасности характеристик они производят уточнения и дополнения описаний объектов, с тем чтобы описать де-факто сложившиеся процедуры и способы работы с системой для выявления в дальнейшем возможных уязвимостей и угроз. Извлечение данных о субъектах специалист по безопасности может производить самостоятельно, опираясь на данные, полученные из описаний работы с объектами/системами, либо он может выделить его в отдельный раздел, ориентированный только на субъектов.

Проведение такого рода обследования обычно осуществляется по следующей схеме:

1. Общее знакомство с системой, визуальный осмотр физического размещения, отдельных компонент или составляющих.
2. Предварительная беседа с администратором/менеджером об общем направлении функционирования системы.
3. Ознакомление с документацией по информационной системе.
4. Составление описания системы с точки зрения информационной безопасности.
5. Уточнение описания на основе работы с документацией и с привлеченными специалистами.

Необходимо отметить, что для качественной дальнейшей работы данные характеристики следует хорошо структурировать, например, по следующим параметрам:

- ☐ аппаратное обеспечение информационной системы (компьютеры, модемы, маршрутизаторы, мосты, повторители, принтеры и прочие периферийные устройства);
- ☐ отдельно можно выделить сетевое обеспечение (сетевые кабели, разъемы, розетки, коннекторы и т. п.);
- ☐ системное программное обеспечение (операционная система, другие средства создания среды работы, например, программы резервного копирования или СУБД);

- ❑ прикладное программное обеспечение, т. е. программы, выполняющие собственно функции производственные, вспомогательные и сопутствующие производству;
- ❑ организационное обеспечение, т. е. пользователи или субъекты системы и их функциональные обязанности в системе;
- ❑ нормативное обеспечение — правила и инструкции работы с системой, возможно, отдельные выдержки из них;
- ❑ данные — информация, которая используется в работе системы в ее производственном значении.

В зависимости от конкретной организации, описание может быть дополнено другими разделами. Например, можно включать функциональное назначение отдельных объектов системы.

Кроме того, необходимо определиться с тем, что считать отдельным объектом системы, подлежащим защите? Отдельный компьютер? Отдельный логический модуль? Если взять в качестве примера систему с трехзвенной архитектурой (клиент—сервер приложений—сервер данных), то в зависимости от особенностей, классификация может быть различной. Можно посчитать всю систему единым объектом, а можно каждое звено рассматривать отдельно (получив три объекта).

Подробному рассмотрению данных, используемых в системе, будет посвящена *глава 6*.

С чего начать: нормативные документы и круг респондентов



Если вы специалист по информационной безопасности, который серьезно берется за проведение инвентаризации информационных активов предприятия, то и подготовка к данному мероприятию должна быть соответственно серьезной.

При чересчур сильной ориентации только на себя и свое подразделение специалист по безопасности рискует получить качественный и легко используемый в работе отчет, который тем не менее может не содержать очевидных для пользователей систем уязвимостей и угроз и быть, таким образом, неполным. Пример — описана система, защита которой базируется на контроле доступа с распределением ролей и авторизацией по паролю. Система использования паролей, сертифицированная по уровню XYZ — максимально надежному. Общий вывод — система достаточно надежна. И вдруг... Оказывается, пароль пользователь системы выбирает не сам — ему назначает его администратор и при этом еще сообщает этот пароль руководителю подразделения данного пользователя. Достаточно ли надежна такая система? Ответ, конечно, очень зависит от многих факторов, но в общем случае весьма близок к отрицательному.

Если просто перечислить список информационных систем и отдать для насыщения его конкретикой конечным менеджерам и пользователям, всегда есть риск быть заваленным какой угодно информацией, только не той, которая нужна. Пример — указана система хранения ключевых отчетов предприятия. В качестве описания можно получить информацию обо всех видах имеющихся отчетов, экранных формах, скорости доступа и пр., но обнаружить во всем этом схему работы подсистемы безопасности очень затруднительно. Причем это в лучшем случае, если респонденты добросовестно отнесутся к своей работе. А ведь можно получить и формальные отписки.

Таким образом, перед началом инвентаризации необходимо разработать, согласовать и утвердить порядок и методику проведения обследования. Этот документ (или документы) должен содержать цели и принципы проведения данного мероприятия, с тем чтобы они были прозрачны для тех, кто будет вовлечен в процесс. Возможно, необходимо утвердить у руководителя предприятия полномочия службы или специалиста, производящего инвентаризацию, а также список участников — привлеченных специалистов, для придания процессу необходимого статуса значимости.

В методической части документа должно быть описано, что нужно сделать привлеченным специалистам, чтобы выполнить свою работу. Целесообразно укомплектовать документ в качестве приложения анкетой, которую структурировать таким образом, чтобы максимально облегчить ее заполнение — выбор из уже имеющихся вариантов, числовые оценки и т. д.

Принципы и направления



Ниже приведены несколько выдержек из подобного документа, которые могут помочь соответствующему специалисту в создании своего варианта.

Принципы проведения инвентаризации.

- ❑ *Принцип единообразного подхода* подразумевает рассмотрение любого объекта/системы с точки зрения технологии создания, обработки, хранения, отправки или приема информации.
- ❑ *Принцип объективности* означает подход с позиций оценки информационной безопасности при критическом анализе системы/объекта.
- ❑ *Принцип многоуровневого подхода* означает рассмотрение объекта/системы путем разделения его на составные части (аппаратное обеспечение, программное обеспечение и т. п.).
- ❑ *Принцип сопряжения* означает, что необходимо указать, от каких систем информация поступает в данную, и в какие системы информация направляется от данной.

Направления проведения инвентаризации.

- ☐ *Физическое* — описывает географическое расположение и размещение системы или отдельных компонент с учетом описания подсистем разграничения доступа.
- ☐ *Технологическое* — описывает используемые технические средства (аппаратное и программное обеспечение, алгоритмы и схемы и т. п.).
- ☐ *Функциональное* — описывает место системы в производственном процессе и выполняемые задачи.
- ☐ *Организационное* — описывает персонал, задействованный в работе системы, и его обязанности.
- ☐ *Нормативное* — описывает имеющиеся документы, регламентирующие работу системы.
- ☐ *Информационное (в смысле данных)* — описывает производственный или бизнес-характер информации, с которой работает система.

Примерный объем собираемой информации



Наряду с проведением общей инвентаризации, имеющей целью определить объекты защиты, из подобного обследования можно получить и другие данные, являющиеся для самой инвентаризации вспомогательными, но имеющие самостоятельную ценность. Это, например, список программного обеспечения (системного и прикладного), используемого на предприятии. Имея подобный список, при условии поддержания его в актуальном состоянии, можно отслеживать автоматизированными или иными способами появление в информационном пространстве предприятия постороннего программного обеспечения, установленного без санкции уполномоченных служб. Ближайший пример — компьютерные игры.

Можно просто составить список программного обеспечения с рядом ключевых характеристик (производитель, номер версии, дата установки, назначение, если есть возможность — контрольная сумма файлов), но лучше сразу произвести его классификацию, например, по следующим параметрам:

- ☐ категория применения (общее, специализированное, индивидуальное);
- ☐ функциональное назначение (производственная или иного рода задача, для которой используется данное программное обеспечение);
- ☐ принадлежность пользователю (кто управляет использованием данного программного обеспечения, т. е. его администратор);
- ☐ размещение компонент программного обеспечения (рабочие станции, серверы);
- ☐ способы доступа (локальный, удаленный).

Дополнительно в порядке инвентаризации стоит предусмотреть формы документов, которые будут использованы, а именно:

- ☐ опросные листы или анкеты — источник получения данных для анализа;
- ☐ промежуточные документы — средства обработки данных;
- ☐ отчеты — результат проведенного обследования.

При работе по направлениям инвентаризации информационной системы необходимо будет зафиксировать ряд деталей, которые могут показаться излишними на данном этапе, но в дальнейшем будут весьма полезными.

По физическому размещению:

- ☐ здание, помещение;
- ☐ номера телефонов помещений;
- ☐ механизмы контроля доступа в помещение и другие защитные механизмы;
- ☐ номера или адреса сетевых точек;
- ☐ ответственный за помещение, если таковой существует.

По аппаратному обеспечению:

- ☐ физическое размещение в помещении;
- ☐ наименование фирмы-производителя;
- ☐ серийный номер;
- ☐ функциональное назначение в системе;
- ☐ встроенные механизмы защиты;
- ☐ адреса портов и прочие сетевые адреса (MAC, IP и др.);
- ☐ ответственный за данную единицу оборудования, если есть.

По программному обеспечению (кроме уже указанных параметров):

- ☐ производитель;
- ☐ номер версии;
- ☐ встроенные механизмы защиты;
- ☐ статистика сбоев, если есть.

По функциональному назначению (данный раздел очень зависит от особенностей производства, но скорее всего следующие пункты должны присутствовать):

- ☐ какие подразделения являются потребителем данных системы;
- ☐ кто занимается технической поддержкой системы;
- ☐ роль системы в общем производственном цикле.

По организационному обеспечению:

- ☐ функциональные и/или должностные обязанности персонала;
- ☐ профессиональный уровень подготовки персонала (образование, дополнительное обучение);
- ☐ существующие процедуры по обеспечению безопасности;
- ☐ временной график выполнения работ в системе;
- ☐ логические схемы движения информации между пользователями.

По нормативному обеспечению:

- ☐ список имеющихся документов — правил, инструкций и т. п.;
- ☐ отдельно — документы по безопасности;
- ☐ даты последнего внесения изменений в документы;
- ☐ сертификаты на аппаратное и/или программное обеспечение.

По данным (этот раздел также очень зависит от особенностей производства, но следующие пункты обязательны вне зависимости от его специфики):

- ☐ откуда поступают данные для системы;
- ☐ куда поступают данные от системы;
- ☐ какая работа с данными происходит в системе;
- ☐ формат хранения данных в системе;
- ☐ вид данных — первичные таблицы данных, транзакции, отчеты и т. п.;
- ☐ логические схемы движения информации между объектами.

Все собранные данные, кроме того что послужат основой для модели управления рисками, еще и помогут (при поддержании их в актуальном состоянии) в дальнейшей работе, например при предварительной подготовке проведения расследования в случаях нарушения информационной безопасности, о которых будет рассказано в следующих главах.

Последний совет — необходимо продумать всю последовательность действий и необходимых данных от начала до конца, с тем чтобы не пришлось на этапе собственно оценки рисков (которому будет посвящена *часть IV*) спохватываться и возвращаться к началу — инвентаризации — за пропущенными сведениями.

Контроль инвентаризации



Данный раздел будет интересен трем категориям специалистов:

- ☐ аудиторам, проводящим проверку качества работы службы информационной безопасности предприятия, например, для вынесения решения об общем уровне надежности организации;

- руководителю предприятия, проверяющему качество работы своей или, особенно, нанятой со стороны команды по созданию или совершенствованию информационной защиты;
- привлеченной команде по созданию или совершенствованию информационной защиты для того, чтобы понять, какая работа и насколько качественно уже была проделана.

В любом из перечисленных случаев необходимо убедиться, что работа по инвентаризации информационных систем была произведена качественно. Правда, для того чтобы тщательно проверить выполненную работу, специалисту придется ознакомиться с предыдущим разделом *"Причины и направления"*.

Предложим краткий конспект проведения такой проверки.

1. Для анализа результатов следует получить отчеты по выполненной инвентаризации, структурированные по информационным системам. Отсутствие таких отчетов не обязательно означает, что подобной работы не проводилось. Возможно, она проводилась без соответствующего оформления, что бывает характерно для небольших организаций, где все сотрудники знают друг друга и предпочитают неформальное общение по служебным вопросам. Другой возможный вариант — работа по построению информационной защиты проводилась только силами службы информационной безопасности, без привлечения специалистов других подразделений, и потому документально оформлены только окончательные результаты (стандарты безопасности, процедуры и другие документы), без фиксации промежуточных результатов.

Тем не менее это все же результат скорее отрицательного характера, так как в целом он затрудняет работу службы информационной безопасности. Например, при добавлении новой системы в общее информационное пространство для качественного обеспечения ее безопасности и безопасности информационного пространства в целом, часть работы по инвентаризации придется провести заново. Особенно трудно придется в точках сопряжения новой системы с уже существующими. Если же в распоряжении службы безопасности имеются актуальные данные по инвентаризованным системам, а также соответствующие порядок и методика, интеграция новой системы будет произведена гораздо быстрее и более гладко.

6. Если результаты инвентаризации присутствуют, необходимо ознакомиться с порядком и методикой ее проведения, а также анкетами и опросными листами, если таковые имеются. Эти документы прояснят, насколько можно доверять результатам инвентаризации в том смысле, что они не взяты из головы специалистов службы безопасности, проводивших обследование.

Отсутствие этих документов еще не показатель недобросовестности работы, а вот их наличие — скорее признак качественного проведения

инвентаризации. Если же документов нет, то, возможно, способы проведения обследования были согласованы устно. В этом случае тем не менее следует попросить уполномоченного специалиста оформить принципы проведения инвентаризации письменно *постфактум*. Так будет легче их оценить.

В принципе для профессионала ознакомление с этими двумя видами документов по инвентаризации (по способам проведения и по результатам) будет вполне достаточно для того, чтобы сделать вывод о качестве проведения мероприятия. Однако для окончательной убежденности можно выбрать более коварный, правда, и более трудоемкий способ.

При необходимости, консультируясь с соответствующими специалистами, следует выбрать одну из информационных систем предприятия, не самую главную, а второго-третьего уровня важности. Далее, на выбор — либо ознакомиться с документацией по системе (руководство по установке или конфигурации, руководство администратора или продвинутого пользователя и т. д.), либо провести беседу с администратором и/или подготовленным пользователем системы. Целью исследования является выявление ключевой особенности работы системы, возможно, нестандартной настройки, используемого протокола или пользовательской процедуры.

Теперь следует обратиться к письменным результатам инвентаризации (стандартным или заранее подготовленным по запросу соответствующим специалистом) и оценить, как найденная характеристика или особенность отражена в представленных документах. Примерную категорию оценок можно распределить следующим образом.

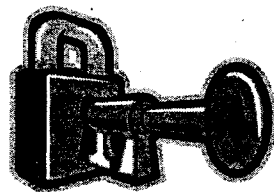
- В результирующих документах исследуемая система не упомянута вообще — очень плохо.
- Система описана, однако выявленная характеристика в ней не отражена, уполномоченный специалист ничего не может пояснить по этому поводу — плохо.
- Система описана, выявленная характеристика в ней не отражена, однако уполномоченный специалист без дополнительных усилий рассказывает по поводу особенностей выявленной характеристики — скорее всего формальные результаты инвентаризации просто устарели, либо оформлены не полностью.
- Система и соответствующая характеристика описаны достаточно подробно, чтобы служить основой для дальнейших оценок — инвентаризация проведена качественно.

Место инвентаризации в жизненном цикле предприятия

Для того чтобы поддерживать данные по инвентаризации в актуальном состоянии, необходимо выполнение ряда требований, а именно:

- ❑ служба информационной безопасности в обязательном порядке должна быть вовлечена в процесс проектирования или приобретения новых систем, реконфигурации существующих, перемещений систем в информационном пространстве, как физически, так и логически, и во все другие процессы, которые производят изменения в существующем информационном пространстве, причем на самой ранней стадии;
- ❑ сведения о приеме на работу новых сотрудников, перемещении по службе существующих, а также об отпусках, командировках, болезнях и увольнении сотрудников предприятия должны регулярно поступать в службу информационной безопасности.

Глава 6



Классификация информационных систем

Классификация субъектов и объектов информационной безопасности (ИБ), а также применяемых на предприятии средств работы с информацией — один из важнейших этапов построения комплексной системы безопасности. От корректности и тщательности ее проведения зависит то, насколько адекватно меры безопасности будут отражать специфику бизнес-процесса, и то, насколько удачно будут распределены материальные вложения в систему ИБ.

Введение в классификацию ИС

Терминология и постановка задачи

Для понимания материала, изложенного в данной главе, необходимо разделить понятие информационной системы на объекты, субъекты и средства работы с информацией. В *главе 2* эти термины были уже определены, но для избежания путаницы повторим и уточним определения.

Объектом будем называть информацию создаваемую, хранимую, обрабатываемую, отправляемую или принимаемую.

Субъектом будет выступать любой пользователь системы, ориентированный как на производственные или иные задачи, так и на поддержку самой системы (администратор).

Средство работы с информацией — это набор аппаратного и программного обеспечения, с помощью которого производится работа в системе, т. е. субъекты воздействуют на объекты.

Таким образом задача сводится к классификации:

□ объектов — по степени их важности для предприятия;

- средств работы с информацией — их способности поддерживать predetermined уровень информационной безопасности;
- субъектов — по степени их допуска к работе с тем или иным объектом (или на том или ином средстве работы с информацией).

В идеальном варианте, когда классификация полностью проведена, остается только привести в соответствие три составляющих — объект данной категории чувствительности обрабатывается только средством соответствующей категории надежности субъектом соответствующей категории доступа. Если не найдено средство или субъект соответствующего уровня — нужно сделать обновление или докупить средства и произвести обучение или прием на работу новых пользователей.

Правда, реальность в данном вопросе более сложная. Проще всего на первый взгляд классифицировать субъектов, учитывая опыт, накопленный отделами по режиму (так называемыми "первыми отделами") советских предприятий. Однако в современном информационном пространстве необходимо учитывать не только общий уровень благонадежности пользователя, но и такие факторы как:

- общая квалификация или способность работать в информационной системе, не внося в нее сбоев из-за человеческого фактора. Известна статистика нарушений информационной безопасности. Мы не будем приводить конкретных цифр — они меняются в зависимости от времени и агентства, производившего опрос. В любом случае, например, цифра, равная 1% нарушений, приемлемая для одной организации, может оказаться катастрофичной для другой;
- подготовка пользователя в сфере информационной безопасности или осведомленность его в вопросах установленных процедур безопасности. Любой доверенный и квалифицированный пользователь, не знакомый с возможностью и принципами социальной инженерии (см. разд. "Социальная инженерия" главы 10), представляет собой угрозу.

Классификация субъектов будет подробно освещена в главе 7, а в данной коснемся вопроса классификации объектов и средств работы с ними. Кроме того следует учесть, что хотя в качестве объекта классификации и выделена информация как таковая, иногда бывает необходимо классифицировать и иные ресурсы, которые могут рассматриваться как объект защиты, например, место, выделенное для хранения на жестком диске, или процессорное время. Однако, на наш взгляд, разобравшись в принципах классификации информации, можно при необходимости произвести аналогичную классификацию любых других ресурсов.

Основные регламенты классификации



Тем, кто начинает свое знакомство с информационной безопасностью, уже на первых стадиях ее изучения попадутся названия "Оранжевая книга", "Красная книга", Общие критерии (Common Criteria), TCSEC, ITSEC и др.

Все это критерии определения уровня безопасности систем. Наиболее известные из них следующие:

- ❑ *Оранжевая книга (DoD 5200.28-STD — Trusted Computer Systems Evaluation Criteria)* — выпущенные Министерством обороны США критерии оценки уровня безопасности компьютерных систем как таковых.
- ❑ *Красная книга (NCSC-TG-005 — Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria)* — расширение этих критериев для случаев использования компьютерных систем в информационной сети.

Если читателю доводилось слышать термин "система сертифицирована по классу C2", то это как раз термин, обозначающий уровень информационной безопасности по данным критериям.

При этом необходимо отметить, что в России существуют свои нормативные документы по данному вопросу. Это документы, выпущенные Государственной технической комиссией при Президенте Российской Федерации: *Классификация автоматизированных систем и требования по защите информации, Показатели защищенности от несанкционированного доступа к информации, Положение по аттестации объектов информатизации по требованиям безопасности информации* и др., о которых будет рассказано в главе 29.

Если отсутствуют соответствующие регламенты на использование того или иного подхода к оценке конкретной информационной системы, то можно выбрать наиболее понравившийся — практически все указанные документы доступны в Интернете. Хотелось бы только предостеречь от поверхностного подхода к таким оценкам.

Например, ряд руководителей, услышав, что C2 — это уровень, рекомендованный для использования в коммерческих учреждениях, начинают в качестве требований к поставщикам указывать необходимость соответствия данному уровню, до конца не понимая из чего он складывается. В таких случаях мы бы рекомендовали пройти хотя бы по названиям подпунктов раздела C2 "Красной книги", чтобы иметь общее представление о данном стандарте. В этой главе в разделе "*Классификация информационных объектов*" будут немного подробнее затронуты вопросы критериев оценки систем.

Теперь рассмотрим классификацию объектов, в данном случае информации. Широко используемая во времена бумажных документов классификация — *секретная, для служебного пользования (ДСП) и открытая* — перекочевала и в век электронной информации. Однако, на наш взгляд, данная классификация не является исчерпывающей и может с натяжкой охватывать в разрезе информационной безопасности только аспект конфиденциальности.

А для успешной работы необходимо как минимум охватить также понятия целостности и доступности информации. На классификацию информации не существует известных (во всяком случае, нам) устоявшихся моделей. Причина этого может скрываться в том, что процесс такой классификации довольно специфичен для каждого предприятия. Тем не менее мы попытаемся привести одну модель, которая может служить наиболее общей основой для построения собственной схемы. Она будет рассмотрена в разделе для специалистов.

Классификация информационных объектов

Типовые сферы применения



В *главе 1*, посвященной общим понятиям информационной безопасности, были рассмотрены основные факторы, на которых основывается информационная безопасность. Соответственно, ценность или важность любого информационного объекта необходимо рассматривать именно с такой точки зрения, так как градации "секретный — конфиденциальный — открытый" или "очень важный — важный — неважный" представляются достаточно бедными для точного определения значения объекта.

Для того чтобы до конца понять эти различия, приведем пример, как главный объект предприятия, который раньше можно было классифицировать как важный или секретный, теперь размещается в разных классификационных категориях.

Пример 1. Государственное учреждение закрытого типа, аналогичное министерству обороны или службе разведки. Для таких учреждений обычно на первом месте стоит понятие конфиденциальности, поэтому скорее будет допущена возможность повреждения или уничтожения информации, чем ее разглашение.

Пример 2. Банк. Если в качестве объекта выступает, например, значение суммы финансовых средств на счету клиента (остаток), то главная задача банка — обеспечить невозможность ее несанкционированного изменения (целостность). При этом в экстраординарных ситуациях можно пойти на временное отсутствие доступа к счету или разглашение данных.

Пример 3. Поставщик интернет-услуг (бесплатный почтовый сервер). Обычно для такого учреждения очень важно обеспечить возможность постоянного доступа пользователей к сервису. Конфиденциальность и целостность остаются также важными, но не всегда первостепенными требованиями.

Примерная модель классификации



Таким образом можно предложить следующую модель для классификации информационных объектов. Для удобства дальнейших ссылок на класс категории рекомендуем сразу ввести буквенно-цифровое обозначение (в приведенном примере литера "Д" означает "доступность", "Ц" — "целостность", "К" — "конфиденциальность", цифры возрастают с убыванием значимости критерия).

По наличию (доступность)

- ☐ Критическая — без нее работа субъекта останавливается (Д0).
- ☐ Очень важная — без нее можно работать, но очень короткое время (Д1).
- ☐ Важная — без нее можно работать некоторое время, но рано или поздно она понадобится (Д2).
- ☐ Полезная — без нее можно работать, но ее использование экономит ресурсы (Д3).
- ☐ Несущественная — устаревшая или неиспользуемая, не влияющая на работу субъекта (Д4).
- ☐ Вредная — ее наличие требует обработки, а обработка ведет к расходу ресурсов, не давая результатов либо принося ущерб (Д5). (В определенных организациях может понадобиться и такой параметр.)

По несанкционированной модификации (целостность)

- ☐ Критическая — ее несанкционированное изменение приведет к неправильной работе всего субъекта или значительной его части; последствия модификации необратимы (Ц0).
- ☐ Очень важная — ее несанкционированное изменение приведет к неправильной работе субъекта через некоторое время, если не будут предприняты некоторые действия; последствия модификации необратимы (Ц1).
- ☐ Важная — ее несанкционированное изменение приведет к неправильной работе части субъекта через некоторое время, если не будут предприняты некоторые действия; последствия модификации обратимы (Ц2).
- ☐ Значимая — ее несанкционированное изменение скажется через некоторое время, но не приведет к сбою в работе субъекта; последствия модификации обратимы (Ц3).
- ☐ Незначимая — ее несанкционированное изменение не скажется на работе системы (Ц4).

Аналогичным образом вместо понятия "несанкционированного изменения" можно рассмотреть понятие "санкционированное изменение, которое не было произведено вовремя".

По разглашению (конфиденциальность)

- ☐ Критическая — разглашение информации приведет к краху работы субъекта или к очень значительным материальным потерям (K0).
- ☐ Очень важная — разглашение приведет к значительным материальным потерям, если не будут предприняты некоторые действия (K1).
- ☐ Важная — разглашение приведет к некоторым материальным (может быть, косвенным) или моральным потерям, если не будут предприняты некоторые действия (K2).
- ☐ Значимая — приносит скорее моральный ущерб, может быть использована только в определенных ситуациях (K3).
- ☐ Малозначимая — может принести моральный ущерб в очень редких случаях (K4).
- ☐ Незначимая — не влияет на работу субъекта (K5).

Для более сложных способов работы с объектами можно дополнительно ввести понятие важности по неотрекаемости и понятие важности по учету, которые упоминались в *главе 1*. Проанализировав общую схему классификации информационных объектов, несложно распространить ее и на другие понятия.

Каждая из указанных выше категорий информации имеет свой жизненный цикл, по истечении периодов которого степень важности объекта, как правило, снижается.

Жизненный цикл обычно можно обозначить следующими стадиями.

- ☐ Информация *используется в операционном режиме*, т. е. принимает участие в производственном цикле и бывает востребована практически постоянно.
- ☐ Информация *используется в архивном режиме*, т. е. не принимает непосредственного участия в производственном цикле, но периодически требуется для аналитической или другой деятельности.
- ☐ Информация *хранится в архивном режиме* для обеспечения соответствия требованиям сохранения (например, вышестоящей организации), практически не нужна самому предприятию.

Классификация средств обработки информации: стандарт CCITSE

Краткий обзор стандарта



Теперь рассмотрим вопросы классификации средств обработки информации. В разделе *"Введение в классификацию ИС"* этой главы, было упомянуто о множестве существующих классификаторов, в том числе и российских.

Подробно с этими документами читатель может ознакомиться самостоятельно. Из их числа несомненно выделяются:

- ❑ CCITSE (Common Criteria for Information Technology Security Evaluation — Общие критерии оценки безопасности информационных систем);
- ❑ наследник британского стандарта BS7799 — стандарт ISO 17799 (International Security Standard — Международный стандарт по безопасности).

Нам хотелось бы остановиться на первом документе по ряду причин. Во-первых, это качественно разработанный и продолжающий совершенствоваться документ, в создании которого принимают участие высокопрофессиональные специалисты из Великобритании, Германии, Канады, Нидерландов, США и Франции. Во-вторых, по оценкам многих специалистов, этот проект может стать единым стандартом для документов такого характера, заменив собой существующие национальные стандарты. Что касается ISO 17799, то, к сожалению, в настоящее время его нет в свободном доступе в Интернете, поэтому его обсуждение с читателями было бы затруднено.

Мы за основу взяли версию 2.1 документа CCITSE [CC1], которая была доступна в Интернете на момент создания книги. Сравнивая ее с предыдущей версией, можно предположить, что основные характеристики документа сформированы и будут перенесены в более поздние версии.

Документ состоит из трех частей, общим размером более 600 страниц. Первая часть — "Введение и общая модель", вторая — "Требования к функциональности безопасности", третья — "Требования к достоверности безопасности".

Подробно описать весь документ на страницах данной книги невозможно, более того, не каждому специалисту он целиком и понадобится. Отметим основные параметры, по которым выполняется оценка, чтобы читатели имели представление, как такая классификация проводится и, возможно, при необходимости смогли создать свою собственную, выбрать подходящую или хотя бы ориентироваться в документе такого класса.

В первой части документа говорится, что данные критерии являются составной частью общего процесса оценки наряду с такими понятиями, как методика оценки и схема оценки, которые не входят в сферу рассмотрения данного документа. Вводится понятие *предмета оценки* (англ. *Target of Evaluation — TOE*), в качестве примера к которому отнесены операционная система, компьютерная сеть, распределенная система или приложение. Также указаны ссылки на понятия конфиденциальности, целостности и

доступности. Кроме того здесь представлены принципы формирования целей информационной безопасности для выбора и определения требований и создания спецификаций для продуктов и систем. Собственно этой же цели посвящена вся книга, которую вы держите в руках, поэтому эта часть документа не представляет непосредственного интереса в данной главе. Документ разделяет потенциальных его читателей на три группы — потребитель, разработчик и специалист по оценке.

Дополнительно в этой части даны простые и понятные схематические интерпретации различных аспектов безопасности, таких как "Концепция и связи в рамках безопасности", "Концепция и связи в рамках оценки", "Модель развития предмета оценки", "Процесс оценки предмета" и др. Также определен подход к оценке путем разделения предмета на классы, семейства и компоненты, и использования требований к безопасности.

Во второй и третьей его частях определяются множества *функциональных* (англ. *functional*) и *доверенных* (англ. *assurance*) компонент как способ определения требований для предмета оценки.

Рассмотрим эти части подробнее.

Требования к функциональности безопасности

Данный раздел основан на части 2 стандарта CCITSE.

Общие понятия

В начале документа рассмотрено соответствие между понятиями "класс", "семейство" и "компонент", определяемых по возрастанию уровня иерархии.

Компонент — наименьшее разделяемое множество элементов, которые могут быть включены в профиль защиты или предмет безопасности (англ. Protection Profile / Security Target — PP/ST).

Семейство — группа компонент, которые объединяются общими целями безопасности, но могут различаться способом реализации или деталями.

Класс — группа семейств, которые объединены общей направленностью.

При этом *профиль защиты* — это независимое от уровня иерархии множество требований по безопасности для предмета оценки, которое соответствует требованиям потребителя, а *предмет безопасности* — это множество требований и спецификаций, используемых как основа для оценки предопределенного предмета оценки.

Класс определяется своим уникальным 3-символьным наименованием, необходимым для идентификации и определения категории, и описанием, которое выражает общую цель или подход семейств, составляющих класс, для соответствия целям безопасности (рис. 6.1).

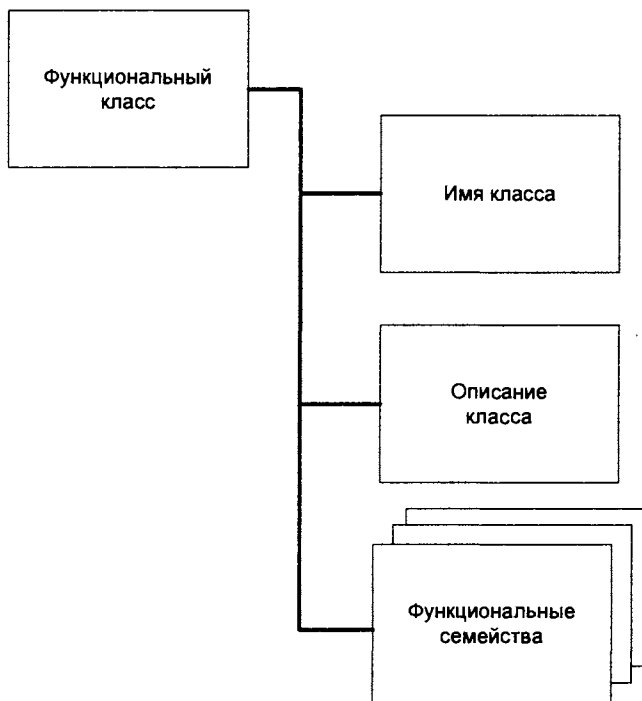


Рис. 6.1. Структура функционального класса

Семейство определяется следующими параметрами (рис. 6.2):

- ❑ *уникальным 7-символьным* (три символа от класса, подчеркивание и три символа от семейства) *наименованием*, предоставляющим информацию для определения категории функционального семейства;
- ❑ *характером* — описанием целей безопасности (проблемы безопасности, которые могут быть разрешены с помощью компонент семейства) и функциональными требованиями (сумма всех требований к компонентам);
- ❑ *уровнем компонента* — информацией для определения соответствующего компонента в рамках семейства, его описание и взаимосвязь с другими компонентами;
- ❑ *управлением* — информацией, необходимой для организации управления компонентом, требования к которой указаны в соответствующем классе (класс FMT — Управление безопасностью);
- ❑ *аудитом* — информацией по событиям, подлежащим аудиту, если соответствующая информация из класса FAU (Аудит безопасности) включена.

Компонент определяется следующими параметрами (рис. 6.3):

- ❑ *идентификация* — информация, состоящая из уникального наименования, отображающего назначение компонента; имени, предназначенного

для ссылок на компонент и отображающего класс и семейство компонента; иерархических связей с другими компонентами;

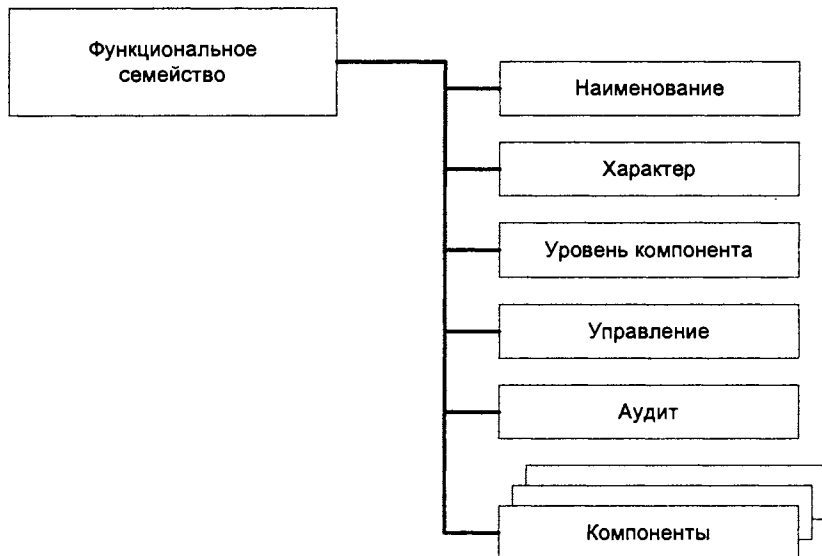


Рис. 6.2. Структура функционального семейства

- ❑ *функциональные элементы* — множество элементов каждого компонента (где элемент — это наименьшая неделимая сущность функциональности безопасности), при этом элемент не может быть отделен от компонента для включения в профиль защиты или предмет безопасности;
- ❑ *зависимости* — минимальный список взаимосвязей с другими компонентами других семейств и классов для обеспечения соответствия требованиям безопасности для данного компонента.

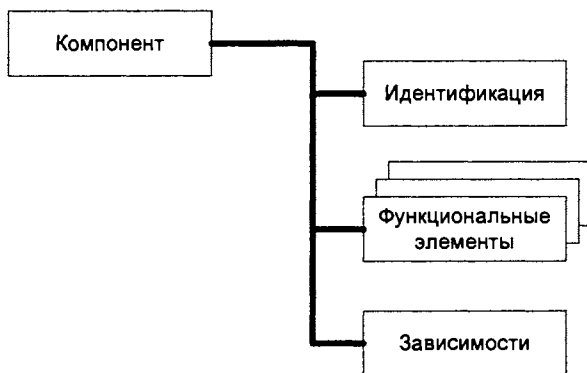


Рис. 6.3. Структура функционального компонента

Перечень классов

Теперь рассмотрим собственно представленные классы и их составляющие (семейства). Интуитивно понятные наименования семейств будем оставлять без пояснений для сокращения объема.

Класс FAU — Аудит безопасности

- ☐ Автоматическая реакция аудита безопасности FAU_ARP (определяет реакцию в случае появления события — потенциального нарушения безопасности).
- ☐ Генерирование данных аудита безопасности FAU_GEN.
- ☐ Анализ аудита безопасности FAU_SAA (для автоматизированных средств).
- ☐ Обзор аудита безопасности FAU_SAR (для человека).
- ☐ Выбор событий аудита безопасности FAU_SEL.
- ☐ Хранение событий аудита безопасности FAU_STG.

Класс FCO — Коммуникации

- ☐ Неотрекаемость источника FCO_NRO.
- ☐ Неотрекаемость получателя FCO_NRR.

Класс FCS — Поддержка криптографии

- ☐ Управление криптографическими ключами FCS_CKM.
- ☐ Криптографические операции FCS_COP.

Класс FDP — Защита данных пользователя

- ☐ Политика управления доступом FDP_ACC.
- ☐ Функции управления доступом FDP_ACF.
- ☐ Аутентификация данных FDP_DAU.
- ☐ Экспорт во внешний управляющий модуль системы FDP_ETC.
- ☐ Импорт из внешнего управляющего модуля системы FDP_ITC.
- ☐ Политика управления информационным потоком FDP_IFC.
- ☐ Функции управления информационным потоком FDP_IFF.
- ☐ Внутреннее перемещение в предмете оценки FDP_ITT (перемещение данных по внутренним каналам в самой системе — предмете оценки).
- ☐ Защита остаточной информации FDP_RIP.
- ☐ Откат FDP_ROL (возможность возврата в корректный статус в случае неверного выполнения или невыполнения модификации).
- ☐ Целостность хранимых данных FDP_SDI.

- ☐ Защита конфиденциальности данных пользователя при межсистемном перемещении FDP_UCT.
- ☐ Защита целостности данных пользователя при межсистемном перемещении FDP_UIT.

Класс FIA — Идентификация и аутентификация

- ☐ Сбой аутентификации FIA_AFL.
- ☐ Определение атрибутов пользователя FIA_ATD.
- ☐ Спецификация секретов FIA_SOS (требования к механизмам обеспечения соответствия конфиденциальных данных предопределенным метрикам).
- ☐ Аутентификация пользователя FIA_UAU.
- ☐ Идентификация пользователя FIA_UID.
- ☐ Привязка пользователя к субъекту FIA_USB (установка ассоциации между пользователем и определенными для него атрибутами безопасности).

Класс FMT — Управление безопасностью

- ☐ Управление функциями системы FMT_MOF.
- ☐ Управление атрибутами безопасности FMT_MSA.
- ☐ Управление данными системы FMT_MTD.
- ☐ Отмена FMT_REV (отмена атрибутов безопасности для различных сущностей внутри системы — предмета оценки).
- ☐ Истечение атрибутов безопасности FMT_SAE.
- ☐ Роли управления безопасностью FMT_SMR.

Класс FPR — Соккрытие данных

- ☐ Анонимность FPR_ANO.
- ☐ Псевдонимность FPR_PSE (пользователь может использовать ресурс без раскрытия свой сущности, но с предоставлением некоего *имени-псевдонима* (англ. *alias*) для средств учета).
- ☐ Разрыв связи FPR_UNL (пользователь может обращаться к различным ресурсам, но при этом невозможно установить связь между ним и использованием этих ресурсов).
- ☐ Неотслеживаемость FPR_UNO.

Класс FPT — Защита системы

- ☐ Тестирование основной абстрактной машины FPT_AMT (тестирование обеспечения системы нижнего уровня — аппаратного обеспечения и

firmware — зашитого в микросхемы программного обеспечения, например BIOS — или аппаратного и системного программного обеспечения).

- ☐ Сбой безопасности FPT_FLS (то есть предмет оценки не скомпрометирует политику безопасности в случае сбоя системы).
- ☐ Доступность данных, экспортированных из системы FPT_ITA.
- ☐ Конфиденциальность данных, экспортированных из системы FPT_ITC.
- ☐ Целостность данных, экспортированных из системы FPT_ITI.
- ☐ Внутрисистемное перемещение данных FPT_ITT.
- ☐ Физическая защита системы FPT_PHP.
- ☐ Уверенное восстановление FPT_RCV (система может стартовать или восстановиться после незавершенной операции без нарушения защиты).
- ☐ Обнаружение повтора FPT_RPL (обнаружение попыток повторной отправки для разных сущностей и соответствующая последовательность действий для исправления).
- ☐ Промежуточная ссылка FPT_RVM.
- ☐ Разделение доменов FPT_SEP.
- ☐ Протокол синхронизации состояния FPT_SSP.
- ☐ Отметка времени FPT_STM.
- ☐ Межсистемная согласованность данных FPT_TDC.
- ☐ Внутрисистемная согласованность копий данных FPT_TRC.
- ☐ Системная самопроверка FPT_TST.

Класс FRU — Использование ресурсов

- ☐ Устойчивость к сбоям FRU_FLT.
- ☐ Приоритет сервиса FRU_PRS.
- ☐ Размещение ресурса FRU_RSA.

Класс FTA — Доступ к предмету оценки

- ☐ Ограничение на объем атрибутов для выбора FTA_LSA.
- ☐ Ограничение на множество конкурирующих сессий FTA_MCS.
- ☐ Блокирование сессий FTA_SSL.
- ☐ Заставка доступа к предмету оценки FTA_TAB (предупреждение о способах использования системы перед идентификацией и авторизацией пользователя).
- ☐ История доступа к объекту оценки FTA_TAH.
- ☐ Установка сессий предмета оценки FTA_TSE.

Класс FTP — Доверенные пути/каналы

- ❑ Межсистемные доверенные каналы FTP_ITC (правила создания доверенного канала между предметом оценки и другим продуктом при выполнении операции с определенным уровнем безопасности).
- ❑ Доверенный путь FTP_TRP (установка доверенной связи между системой и пользователем).

Пример набора требований к одному из классов (поддержка криптографии) приводится в конце этой главы. Это выдержки из раздела 5 и приложения Е части 2 [CC1].

Требования к достоверности безопасности

Данный раздел основан на части 3 стандарта CCITSE.

Общие понятия

Соответствие между понятиями "класс", "семейство" и "компонент" рассмотрим для понятия *уверенность* (англ. *assurance*) — убежденность в том, что некая конкретная сущность соответствует своим целям безопасности. Если понятие класса уверенности практически совпадает с понятием класса функциональности из части 2, то другие составляющие несколько отличаются.

Как видно из рис. 6.4, у составляющих класса появились новые параметры. Рассмотрим их подробнее:

- ❑ *Цели* семейства представляют предназначение данного семейства.
- ❑ *Отметки приложения* описывают дополнительную информацию прикладного характера о семействе, такую как ограничения в использовании или области отдельного внимания.
- ❑ *Цели* компонента описывают специфическое, более детальное предназначение данного компонента.
- ❑ *Отметки приложения* компонента представляют дополнительную информацию, облегчающую использование компонента.
- ❑ *Элемент* уверенности представляет собой одно требование соответствия.

Дополнительно в документе введено понятие *уровень уверенности оценки* (англ. *Evaluation Assurance Level — EAL*). Это набор компонент уверенности, представляющих точку отсчета в предопределенном масштабе расчета уверенности. Уровень уверенности оценки сам состоит из наименования, целей, отметок приложения и компонентов (одинаковых по структуре с компонентами семейства уверенности). Связь между уверенностью как таковой и уровнем уверенности происходит на уровне компонент, т. е. для одного уровня уверенности выбирается набор компонент уверенности. Звучит несколько запутанно, но, к сожалению, приходится балансировать между смыслом и дословным переводом.

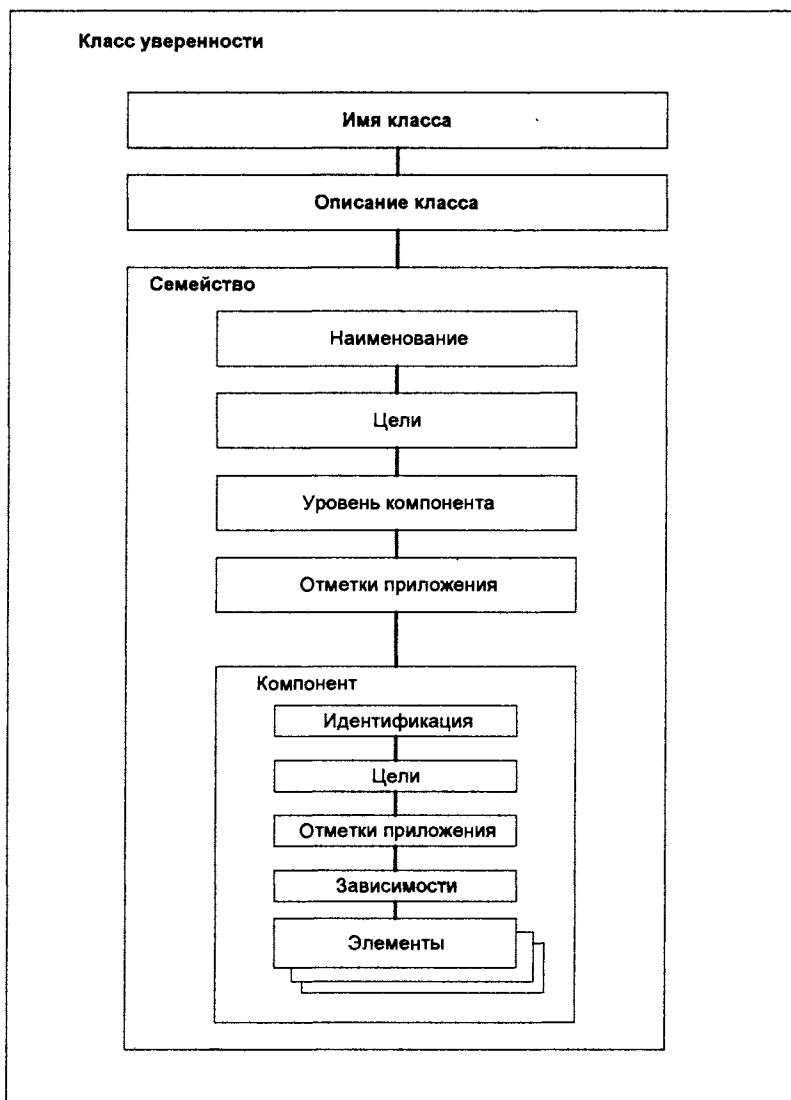


Рис. 6.4. Иерархия класса / семейства / компонента / элемента уверенности

Классы оценок

Отдельно в документе выделено два класса.

Класс АРЕ — Оценка профиля защиты

- ☐ Описание предмета оценки АРЕ_DES.
- ☐ Среда безопасности АРЕ_ENV.

- ☐ Описание профиля защиты APE_INT.
- ☐ Цели безопасности APE_OBJ.
- ☐ Требования информационной безопасности APE_REQ.
- ☐ Подробно описанные требования информационной безопасности APE_SRE.

Класс ASE — Оценка предмета безопасности

- ☐ Описание предмета оценки ASE_DES.
- ☐ Среда безопасности ASE_ENV.
- ☐ Описание предмета безопасности ASE_INT.
- ☐ Цели безопасности ASE_OBJ.
- ☐ Заявки профиля защиты ASE_PPC (цель этого параметра — определить корректность определения предметом безопасности профиля защиты).
- ☐ Требования информационной безопасности ASE_REQ.
- ☐ Подробно описанные требования информационной безопасности ASE_SRE.
- ☐ Суммарная спецификация предмета оценки ASE_TSS.

Далее определяются уровни уверенности по возрастанию:

- ☐ EAL1 — функционально протестировано;
- ☐ EAL2 — структурно протестировано;
- ☐ EAL3 — методически протестировано и сверено;
- ☐ EAL4 — методически проработано, протестировано и проанализировано;
- ☐ EAL5 — полуформально проработано и протестировано;
- ☐ EAL6 — полуформально проработано и протестировано с подтверждением;
- ☐ EAL7 — формально проработано и протестировано с подтверждением.

В данном случае термины "формально" и "полуформально" носят следующий смысл.

Формально — отражено на языке фиксированного синтаксиса с предопределенной семантикой на основе качественно проработанных математических концепций.

Полуформально — отражено на языке фиксированного синтаксиса с предопределенной семантикой.

Технологические классы

Класс ACM — Управление конфигурацией

- ☐ Автоматизация управления конфигурацией ACM_AUT.
- ☐ Возможности управления конфигурацией ACM_CAP.
- ☐ Область действия управления конфигурацией ACM_SCP.

Класс ADO — Доставка (предмета оценки до пользователя) и операции

- ☐ Доставка ADO_DEL.
- ☐ Установка, преобразование и запуск ADO_IGS.

Класс ADV — Разработка

- ☐ Функциональная спецификация ADV_FSP.
- ☐ Проработка верхнего уровня ADV_HLD.
- ☐ Представление внедрения ADV_IMP.
- ☐ Внутренние составляющие системы ADV_INT.
- ☐ Проработка нижнего уровня ADV_LLD.
- ☐ Соответствие представлений ADV_RCR.
- ☐ Моделирование политики безопасности ADV_SPM.

Класс AGD — Руководства и инструкции

- ☐ Руководство администратора AGD_ADM.
- ☐ Руководство пользователя AGD_USR.

Класс ALC — Поддержка жизненного цикла

- ☐ Безопасность разработки ALC_DVS.
- ☐ Устранение изъянов ALC_FLR.
- ☐ Определение жизненного цикла ALC_LCD.
- ☐ Средства и технологии ALC_TAT.

Класс ATE — Испытания

- ☐ Покрытие (полнота тестирования разработчиком) ATE_COV.
- ☐ Глубина (уровень подробности тестирования) ATE_DPT.
- ☐ Функциональное тестирование ATE_FUN.
- ☐ Независимое тестирование ATE_IND.

Класс AVA — Оценка уязвимости

- ☐ Анализ скрытых каналов AVA_CCA.
- ☐ Злоупотребление AVA_MSU.

- ☐ Прочность функций безопасности AVA_SOF.
- ☐ Анализ уязвимостей AVA_VLA.

Класс поддержания уверенности

Кроме перечисленных, выделен отдельный класс — поддержания уверенности, зависящий от жизненного цикла состояния уверенности, состоящий из следующих этапов: принятие предмета оценки, мониторинг предмета оценки, переоценка.

Класс АМА — Поддержание уверенности

- ☐ План поддержания уверенности AMA_AMP.
- ☐ Отчет по категоризации компонент предмета оценки AMA_CAT.
- ☐ Факты (свидетельство) поддержания уверенности AMA_EVD.
- ☐ Анализ воздействий на безопасность AMA_SIA.

Надеемся, у читателя сложилось представление о процессе классификации информационных систем и объеме требуемых действий. По крайней мере можно представить себе, по каким параметрам производится оценка систем, и, соответственно, на что следует обращать внимание при их проектировании, использовании или модернизации.

Пример требований к классу: класс FCS — поддержка криптографии



Данный класс используется, когда предмет оценки выполняет криптографические функции. Состоит из двух семейств — управление криптографическими ключами (FCS_CKM) и криптографические операции (FCS_COP).

Семейство управления криптографическими ключами (FCS_CKM)

Режим функционирования семейства

Криптографические ключи должны быть управляемы (находиться под контролем) в течение всего жизненного цикла. Это семейство предназначено для поддержки своего жизненного цикла и последовательно определяет требования для следующих действий: генерация ключей, распространение ключей, доступ к ключам и уничтожение ключей.

Как минимум криптографические ключи должны проходить следующие стадии: создание, хранение и уничтожение. Включение других стадий зависит от используемой стратегии управления ключами, например, ТОВ может только генерировать и распространять ключи.

Распределение компонент по уровням

FCS_CKM.1 Генерация ключей требует, чтобы ключи создавались в соответствии с обусловленным алгоритмом и с размером (длиной ключа), которые основываются на predetermined стандарте.

FCS_CKM.2 Распространение ключей требует, чтобы ключи распространялись в соответствии с обусловленным способом распространения, который основывается на predetermined стандарте.

FCS_CKM.3 Доступ к ключам требует, чтобы работа с ключами производилась в соответствии с обусловленным методом доступа, который основывается на predetermined стандарте.

FCS_CKM.4 Уничтожение ключей требует, чтобы ключи уничтожались в соответствии с обусловленным способом разрушения ключей, который основывается на predetermined стандарте.

Управление: **FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.4**

Следующее действие должно быть учтено в функциях управления класса FMT:

- ☐ Управление изменениями в атрибутах ключей. Примеры атрибутов ключей включают пользователя, тип ключа (публичный, приватный, секретный), период действия и использование (для цифровых подписей, шифрации ключей, согласования ключей, шифрации данных)

Аудит: **FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.4**

Следующие действия должны подлежать аудиту, если FAU_GEN включен в профиль защиты и предмет безопасности:

- ☐ минимально: успешное или неуспешное действие;
- ☐ базовое: атрибуты объекта и значения (данные) объекта за исключением критичной информации (скажем, секретных или приватных ключей).

FCS_CKM.1 Генерация ключей

Дальнейшего иерархического разделения нет.

Связан с уровнями:

- ☐ **FCS_CKM.2** Распространение ключей или **FCS_COP.1** Криптографические операции;
- ☐ **FCS_CKM.4** Уничтожение ключей;
- ☐ **FMT_MSA.2** Безопасность атрибутов безопасности.

Для генерации ключей обычно используются случайные числа. В этом случае должен быть использован **FCS_CKM.1** вместо компонента **FIA_SOS.2** Создание секретных величин. В случаях, когда генерация случайных чисел используется для других целей, следует использовать компонент "FIA_SOS.2 Создание секретных величин".

FCS_CKM.2 Распределение ключей

Дальнейшего иерархического разделения нет.

Связан с уровнями:

- ☐ FDP_ITC.1 Импорт данных пользователя без атрибутов безопасности или FCS_CKM.1 Генерация ключей;
- ☐ FCS_CKM.4 Уничтожение ключей;
- ☐ FMT_MSA.2 Безопасность атрибутов безопасности.

FCS_CKM.3 Доступ к ключам

Дальнейшего иерархического разделения нет.

Связан с уровнями:

- ☐ FDP_ITC.1 Импорт данных пользователя без атрибутов безопасности или FCS_CKM.1 Генерация ключей;
- ☐ FCS_CKM.4 Уничтожение ключей;
- ☐ FMT_MSA.2 Безопасность атрибутов безопасности.

Примеры типов доступа к криптографическим ключам включают резервное копирование ключей, архивацию ключей, доверенное хранение ключей и восстановлением ключей и др.

FCS_CKM.4 Уничтожение ключей

Дальнейшего иерархического разделения нет.

Связан с уровнями:

- ☐ FDP_ITC.1 Импорт данных пользователя без атрибутов безопасности или FCS_CKM.1 Генерация ключей;
- ☐ FMT_MSA.2 Безопасность атрибутов безопасности.

Семейство криптографических операций (FCS_COP)

Режим функционирования семейства

Типовые криптографические операции включают шифрование и/или дешифрование данных, генерацию и/или верификацию цифровых подписей, генерацию и/или верификацию контрольных сумм для обеспечения целостности, хэш-функции, шифрование и/или дешифрование криптографических ключей, согласование ключей.

Распределение компонент по уровням

FCS_COP.1 Криптографические операции требуют, чтобы операции производились в соответствии с обусловленным алгоритмом и с размером ключей (длиной ключа), которые основываются на predetermined стандарте.

Управление: FCS_COP.1

Для этих компонент не предполагается действий по управлению.

Аудит: FCS_COP.1

Следующие действия должны подлежать аудиту, если FAU_GEN включен в профиль защиты и предмет безопасности.

- ☐ Минимально: успех или неудача и тип криптографической операции.
- ☐ Рекомендуется: любые применяемые виды (режимы) операций, атрибуты объекта и субъекта.

Примерами режимов криптографических операций, в частности, в отношении алгоритмов создания цепочек, являются CBC — cipher block chaining (режим цепочек в блочных шифрах), OFB — output feedback mode (режим с обратной связью по выходу), ECB — electronic code book (режим электронной криптокнижки) и CFB — cipher feedback mode (режим с обратной связью по шифротексту). *(Подробнее об этом можно прочитать в главе 20).*

FCS_COP.1 Криптографические операции

Дальнейшего иерархического разделения нет.

Связан с уровнями:

- ☐ FDP_ITC.1 Импорт данных пользователя без атрибутов безопасности или FCS_CKM.1 Генерация ключей;
- ☐ FCS_CKM.4 Уничтожение ключей;
- ☐ FMT_MSA.2 Безопасность атрибутов безопасности.

Контроль классификации

Еще раз повторим, что необходимо различать понятия классификации самой информации и классификации средств работы с информацией.

Контроль классификации информации



Классификация информации представляет трудность в первую очередь спецификой работы самого предприятия. Поэтому, если классификация информации на предприятии присутствует, необходимо проверить порядок проведения процесса такой классификации. Если классификация производилась на основании умозаключений только представителей службы безопасности, то она может не отражать реальной потребности в защите информационных объектов. То, что кажется маловажным для аналитиков службы безопасности, может показаться критически важным работникам бизнес-подразделений, разработчикам стратегий развития предприятия и другим службам.

Если формальная (задокументированная) классификация отсутствует, то скорее всего присутствует некая сложившаяся неформальная классификация и с ней необходимо ознакомиться. В этом случае не следует принимать во внимание разговоры о том, что все информационные объекты защищаются одинаково хорошо. Такое еще возможно в некой одноранговой сети, где каждый пользователь решает, какую информацию предоставить кому из коллег. При совместной же работе двух и более пользователей с одним объектом рано или поздно появляется проблема отделения одной части информации от другой и разграничения доступа к этим частям. А значит, возникают и разные уровни чувствительности информации хотя бы по параметру конфиденциальность.

Контроль классификации средств работы с информацией



Что касается классификации средств работы с информацией, то необходимо различать классификацию, проведенную силами самого предприятия (его службой информационной безопасности) и классификацию, произведенную третьей стороной (например, производителем системы). При этом вовсе не обязательно, что первый вариант хуже второго. Широко известные, сертифицированные на определенную категорию, системы обычно (и это указывается в сертификационном документе) категоризируются в определенной конфигурации, на определенном оборудовании, в определенной среде и при других определенных условиях. А это значит, что система, сертифицированная по пресловутому классу C2, вовсе не соответствует данной категории на конкретном предприятии по различным причинам (другое оборудование, особенности или ошибки конфигурации и пр.). В этом случае, с одной стороны, собственная категоризация (или перекатегоризация), возможно, окажется более точной. С другой стороны, собственная категоризация может не быть такой подробной, как [CC], например, о которых было рассказано в разделе для специалистов.

В любом случае необходимо убедиться в том, как оценивались надежность механизмов безопасности (аутентификация, криптография и др.), возможность управления этими механизмами, защита данных и разграничение доступа к ним, устойчивость системы к сбоям, качество документации к системе, возможные уязвимости системы и остальные параметры, которые подлежат оценке.

Проверка соответствия информации средствам работы с ней



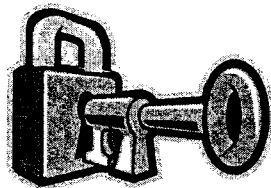
Когда получены категории информации и средств работы с ней или аналоги этих категорий, необходимо убедиться, что работа с информацией данного уровня чувствительности действительно производится на данном средстве с

возможностью поддержания заданного уровня. Такая проверка производится обычно одним из следующих способов:

- ☐ изучением описания работы системы;
- ☐ опросом администраторов и пользователей системы;
- ☐ тестовой работой проверяющего в системе, чаще всего в сопровождении администратора системы.

Если подтверждение соответствия уровней информация/средство работы получено, можно утверждать, что классификация произведена правильно и работа производится в соответствии с классификацией. Если же обнаружены расхождения, необходима дополнительная проверка (в основном опросы уполномоченных сотрудников) для определения причин расхождений и способов их устранения, вплоть до полной переклассификации всех систем.

Глава 7



Роли и ответственность субъектов

Субъекты информационного пространства

Под *субъектом информационного пространства* будем понимать людей, процессы или средства работы с информацией, выполняющие в информационном пространстве некие активные действия в отношении информационной системы. Процессы и средства как безличные сущности могут выполнять свои действия:

- либо от имени пользователей, управляющих ими в данный момент времени, — в этом случае они будут рассматриваться неотделимо от человека и его профиля прав и ответственности;
- либо как независимые субъекты системы с собственными профилями прав доступа.

Следует различать классификацию субъектов в рамках всего информационного пространства предприятия и в рамках конкретной информационной системы. Хотя эти классификации могут и совпадать, но обычно они различаются, так как классификация в конкретной информационной системе носит более производственно-ориентированный характер, чем в первом случае. Например, с точки зрения конкретной системы пользователи, выполняющие транзакции вида 1, и пользователи, выполняющие транзакции вида 2, — это две различные группы пользователей, в то время как для информационного пространства в целом обе эти группы — это субъекты, работающие с одним объектом, база данных транзакций.

По-другому это можно пояснить с точки зрения операционной системы (аналог пространства) и конкретного приложения (аналог информационной системы). В рамках операционной системы два пользователя могут входить в группу users с равными правами на запуск приложения. Однако, работая в конкретной программе, эти пользователи относятся к разным группам: один — бизнес-администратор, а другой — оператор по вводу данных.

В этой главе нам хотелось бы в первом разделе предложить более простую модель классификации субъектов, во втором — описать одну из западных, более сложных моделей, а в третьем разделе — привести примерный вид распределения обязанностей между субъектами.

Принципы распределения прав и ответственности

Прежде чем перейти к конкретным вопросам, заострим внимание на двух фундаментальных принципах, на которых строится распределение ролей и ответственности.

1. *Распределение ответственности* (англ. *separation of duties*). Этот принцип означает, что в системе (пространстве) не должно быть субъекта, который мог бы самостоятельно от начала до конца выполнить операцию, которая может нанести ущерб безопасности. Например, если пользователь А выполняет некую транзакцию, имеющую существенное значение для системы, то она должна вступить в силу только после того, как пользователь Б проверит и подтвердит ее корректность. Таким образом, ни А, ни Б не могут по отдельности произвести данную транзакцию, один может только формировать данные для нее, другой — только авторизовать ее (то есть в данном случае отправить на выполнение), — без этих двух процедур транзакция не будет иметь результата.

Если конкретная операция в системе построена таким образом, что она не может быть разделена между двумя пользователями, значит, при ее критической важности, она должна быть выполнена одним пользователем только в присутствии другого. Это может быть достигнуто, например, разделением пароля на аутентификацию пользователя, от имени которого запускается функция, на две (или более) части. Иногда можно встретить другое определение такого принципа — *"в четыре глаза"* (англ. *four eyes*).

Если же объективно ряд значимых операций должен выполняться только одним человеком, то результаты его действий должны регистрироваться в журналах, к которым он не имеет доступа на корректировку. Эти журналы должны регулярно анализироваться другим человеком.

Набор прав и обязанностей, который можно применить к тому или иному сотруднику в зависимости от его функций, называется *ролью*. Цель политики ролей на предприятии — унифицировать и упростить делегирование прав сотрудникам. После разработки подобной схемы исчезает необходимость составлять ради каждого вновь прибывшего работника набор его прав и обязанностей и настраивать сотни учетных записей. Достаточно указать принадлежность сотрудника той или иной роли, и, возможно, произвести минимальное редактирование профиля.

2. *Минимизация привилегий* (англ. *least privelegies*). Этот принцип означает, что пользователю предоставляется ровно столько прав, сколько ему необходимо для выполнения работы, т. е. если в обязанности пользователя входит формирование отчетов по всей базе данных, то он должен иметь доступ ко всем данным, но только на чтение. Если пользователь должен выполнять транзакции, изменяющие состояние двух групп объектов, то он должен иметь доступ на модификацию (в зависимости от конкретной задачи, возможно, на создание и/или удаление) только к этим двум объектам.

Можно возразить, что объективно в информационном пространстве должен присутствовать субъект, имеющий полный доступ ко всем ресурсам, хотя бы для того, чтобы добавлять новых пользователей и наделять их правами на доступ (а значит, и имеющий возможность установить себе полный доступ ко всем информационным объектам) — например, администратор локальной сети. Возможно, это будет верно для небольших предприятий, однако там, где есть возможность содержать в штате более одного администратора, необходимо разделять их права, например, с помощью выделения доменов ответственности, на ресурсном принципе или по какой-либо другой схеме.

Упрощенная модель классификации субъектов



Рассмотрим упрощенную, или обобщенную, модель классификации субъектов. В информационном пространстве предприятия существуют следующие субъекты:

- ☐ создающие объекты;
- ☐ использующие объекты;
- ☐ администрирующие объекты (то есть обеспечивающие среду работы с объектами других субъектов);
- ☐ контролирующие использование объектов субъектами.

Каждый конкретный субъект может совмещать в себе несколько или все указанные классификационные сущности.

Определим следующие группы по способам работы с информацией.

- ☐ Группа А. Один субъект владеет информацией, самостоятельно обрабатывает ее без передачи другим субъектам. В этом случае он сам контролирует все способы работы и классификацию информации.
- ☐ Группа Б. Один субъект владеет информацией и передает ее для использования другому субъекту или группе субъектов. При этом он должен произвести классификацию информации, определить правила ее использования и ознакомить с ними пользователей, либо уполномочить другого субъекта (или нескольких субъектов) на выполнение этих действий.

- ❑ Группа В. Группа субъектов использует один и тот же информационный объект (в целом или различные его части) или совокупность объектов без явно выраженного права владения одним субъектом. В этом случае необходимо произвести разделение субъектов на пользователей, администраторов и контролеров.
- ❑ Группа Г. Субъект или группа субъектов использует информацию, владелец которой находится вне организации. В этом случае субъекты следуют правилам использования информации, определенной владельцем в рамках законодательного пространства.
- ❑ Группа Д. Группа субъектов использует информационные объекты широкого или неопределенного доступа. В этом случае работа с объектами производится без ограничений, в рамках законодательного пространства.

Наибольший интерес для нас представляет группа В. В соответствии с описанными группами будем различать следующие обязанности субъектов по жизненным стадиям информации.

Субъекты-администраторы (далее просто администраторы) должны обеспечить наличие условий для создания, использования, передачи и хранения объектов в соответствии с задачами, определяемыми уполномоченными *субъектами-пользователями* (далее пользователями), а также создать условия для осуществления контроля со стороны *субъектов-контролеров* (далее контролеров) за действиями как пользователей, так и администраторов. В своей работе администраторы должны следовать разработанным правилам. Контролеры должны разработать (и при необходимости дорабатывать) правила, по которым пользователи будут работать с объектами. Кроме того, они должны произвести классификацию информационных объектов, обеспечить ознакомление пользователей с правилами и классификацией, а также контролировать соблюдение этих правил. Пользователи должны работать с объектами в соответствии с бизнес-задачей в рамках разработанных правил.

Полная модель классификации субъектов



Рассмотрим более сложную западную модель, предложенную в книге "Handbook of Information Security Management" ("Руководство по управлению информационной безопасностью").

Владелец информации — бизнес-менеджер, который ответственен за информационные активы предприятия. Обязанности следующие:

- ❑ устанавливать первичную классификацию информации и периодически проверять, что эта классификация отвечает производственным задачам;
- ❑ определять работу механизмов безопасности в соответствии с классификацией;

- ☐ анализировать актуальность прав доступа к информационным активам;
 - определение требований безопасности, резервного копирования и критериев доступа к информационным активам;
- ☐ выполнять или назначать исполнителей для следующих операций:
 - санкционирование запросов на доступ от других бизнес-подразделений;
 - резервное копирование;
 - восстановление данных;
- ☐ санкционировать различные действия по фактам нарушения безопасности.

Хранитель информации — обычно специалист по информационным технологиям, основная задача которого — резервное копирование и восстановление данных. Обязанности следующие:

- ☐ производить резервное копирование в соответствии с требованиями, установленными владельцем информации;
- ☐ при необходимости восстанавливать потерянные или поврежденные данные;
- ☐ производить необходимые мероприятия по обеспечению сохранности и доступности данных из резервных копий;
- ☐ обеспечивать учет хранения в соответствии с требованиями владельца информации.

Владелец приложения — руководитель бизнес-подразделения, который полностью ответственен за выполнение производственных или иных функций, обслуживаемых приложением. Имеет следующие обязанности:

- ☐ устанавливать критерии доступа пользователей и требования к доступности для приложения;
- ☐ контролировать адекватность использования механизмов безопасности приложения (то есть, если информация, скажем, "совершенно секретная", то и механизм безопасности не может быть таким же, как для просто "секретной" информации);
- ☐ исполнять или поручать исполнение следующего:
 - ежедневное администрирование безопасности;
 - рассмотрение отдельных запросов на доступ;
 - анализ случаев нарушения безопасности;
 - рассмотрение и утверждение всех изменений к приложению до их установки на реальную систему;
 - подтверждение актуальности прав доступа пользователей в рамках приложения.

Администратор пользователей — непосредственный руководитель сотрудников. В его полной ответственности — учетные данные пользователей и информационные ресурсы работников предприятия (то есть системы, приложения, данные и т. п.). В случаях, когда для выполнения операций привлечены не работники предприятия (контрактники, консультанты и т. д.), в его ведении находится их деятельность и информационные ресурсы, используемые ими. Администратор пользователей обычно отвечает за привлечение сторонних организаций. Обязанности:

- ❑ информировать администратора безопасности об увольнении пользователя для удаления учетных записей пользователя, их отключения или временного блокирования;
- ❑ информировать администратора безопасности о служебных перемещениях пользователей, если это влечет изменение форм или прав доступа;
- ❑ докладывать в службу информационной безопасности обо всех происшествиях по безопасности или подозрении на такие происшествия;
- ❑ контролировать актуальность пользовательской учетной информации;
- ❑ формировать и предоставлять первичные пароли для новых пользователей;
- ❑ проводить обучение пользователей в вопросах политики безопасности, ее процедурам, стандартам и т. п.

Администратор безопасности — сотрудник предприятия, который имеет соответствующие полномочия в системе управления доступом. Он устанавливает механизмы безопасности, администрирует учетные записи пользователей и права доступа к информационным ресурсам. Он подотчетен либо бизнес-подразделению, либо службе информационной безопасности внутри подразделения информационных технологий. Имеет следующие обязанности:

- ❑ разбираться в различных средах обработки данных и в результатах предоставления доступа к ним;
- ❑ контролировать тот факт, что запросы на доступ соответствуют общей линии использования информации и правилам безопасности;
- ❑ администрировать права доступа в соответствии с критериями, установленными владельцем информации;
- ❑ создавать и удалять учетные записи пользователей, установленные администратором пользователей;
- ❑ администрировать систему в рамках своей работы и функциональных обязанностей;

- ☐ расследовать отчеты о нарушениях безопасности;
- ☐ направлять начальные пароли новых пользователей только непосредственным начальникам этих пользователей.

Аналитик безопасности — сотрудник, ответственный за определение развития безопасности данных (стратегий, процедур, правил) для обеспечения уверенности в том, что контроль и защита информации основаны на значимости информации, риске потери или компрометации и легкости восстановления. Обязанности:

- ☐ предоставлять руководства по безопасности для процедур управления информацией;
- ☐ обеспечивать понимание основных принципов работы с информацией, для того чтобы убедиться, что используются соответствующие механизмы контроля;
- ☐ обеспечивать участие, консультации и анализ в разработке систем защиты данных.

Аналитик контроля модификаций — сотрудник, ответственный за анализ запросов на модификацию инфраструктуры информационных технологий и определение влияния этих изменений на работу приложений.

Аналитик данных — этот сотрудник анализирует бизнес-требования к разработке структур данных, рекомендует определение стандартов данных и физические платформы для них. Он ответственен за применение соответствующих стандартов управления данными. Обязанности:

- ☐ разрабатывать структуру данных для соответствия потребностям бизнеса;
- ☐ разрабатывать физическую структуру баз данных;
- ☐ создавать и поддерживать логические модели данных на основе бизнес-требований;
- ☐ обеспечивать техническую поддержку владельцу информации в разработке архитектуры данных;
- ☐ записывать метаданные (сведения о хранении данных) в библиотеку данных;
- ☐ создавать, поддерживать и использовать метаданные для эффективного управления распределением данных.

Провайдер (поставщик) решений — сотрудник, участвующий в разработке решений (приложений) и процессе разворачивания бизнес-решений. В различных информационных системах называется *интегратором, разработчиком приложений, провайдером информационных технологий*. Обязанности:

- ☐ работать с аналитиком данных для обеспечения уверенности, что приложение и данные будут работать совместно в соответствии с бизнес-требованиями;

- ❑ передавать технические требования аналитику данных для обеспечения соответствия требований производительности и отчетности.

Конечный пользователь — любой сотрудник, контрактник или поставщик предприятия, использующий информационные системы и ресурсы в рамках своей работы. Обязанности:

- ❑ сохранять в тайне пароли на доступ;
- ❑ осознавать, что безопасность информации — это в том числе и его забота;
- ❑ использовать бизнес-активы и информационные ресурсы предприятия только для целей, определенных руководством;
- ❑ соблюдать все аспекты политики безопасности, процедур, стандартов и руководств по информационной безопасности;
- ❑ по запросу предоставлять руководству отчеты по событиям, связанным с информационной безопасностью.

Владелец процесса — сотрудник, ответственный за внедрение, управление и постоянное улучшение процесса, соответствующего определенной потребности производства. Обязанности:

- ❑ контролировать требования к данным, которые должны быть направлены на поддержку бизнес-процесса;
- ❑ понимать, как доступность и качество влияют на эффективность процесса;
- ❑ работать совместно с владельцем информации для определения и отстаивания программы качества данных внутри процесса;
- ❑ разрешать проблемы с данными в рамках приложения внутри процесса.

Администратор продукта — сотрудник, ответственный за понимание бизнес-требований и формулирование их в виде требований к продукту, за работу с поставщиком и пользователями для обеспечения соответствия продукта этим требованиям, за отслеживание новых версий и за контакты с ключевыми фигурами по вопросам приобретения новых версий. Обязанности:

- ❑ обеспечивать оценку новых версий и обновлений, планируемых для внедрения, и качественное их внедрение;
- ❑ обеспечивать соответствие программного обеспечения лицензионным соглашениям;
- ❑ отслеживать производительность продукта в соответствии с требованиями производства;
- ❑ анализировать использование, тенденции и возможности продукта для определения действий, обеспечивающих соответствие требованиям продукта по продукту.

Очевидно, что не все из перечисленных ролей и обязанностей могут быть однозначно применены, особенно на малых и средних предприятиях, однако общие принципы распределения обязанностей и ответственности сотрудников весьма полезны для осознания разделения сфер влияния и контроля в информационном пространстве с точки зрения информационной безопасности.

Сопоставление ролей функциональным обязанностям



На рис. 7.1 приведен примерный вид матрицы распределения доступа по узкому кругу функциональных обязанностей для сотрудников типовой организации.

Объекты Субъекты	Права и профили	Электронный документ	Флаг "ОК" (ЭЦП)	Регистрационный журнал
Система (ПО)	<i>Полный доступ</i>	<i>Полный доступ</i>	<i>Полный доступ</i>	<i>Полный доступ</i>
Администратор системы	<i>Полный доступ</i>	<i>Нет доступа</i>	<i>Нет доступа</i>	<i>Чтение Удаление</i>
Администратор безопасности	<i>Чтение</i>	<i>Чтение</i>	<i>Чтение</i>	<i>Чтение</i>
Пользователь 1	<i>Нет доступа</i>	<i>Создание</i>	<i>Чтение</i>	<i>Нет доступа</i>
Пользователь 2	<i>Нет доступа</i>	<i>Чтение</i>	<i>Создание</i>	<i>Нет доступа</i>
Начальник отдела	<i>Нет доступа</i>	<i>Чтение</i>	<i>Чтение</i>	<i>Нет доступа</i>

Рис. 7.1. Пример матрицы распределения доступа к объектам

Распределение ролей, представленное здесь, конечно же, несколько условно. Скажем, очевидно, что рядовой пользователь (или процесс от его имени) должен иметь доступ к регистрационному журналу, с тем чтобы регистрировать события, связанные с этим пользователем.

Рассмотрим функциональные обязанности субъектов по схеме.

- ☐ Система обеспечивает функционирование всей схемы.
- ☐ Администратор системы настраивает права доступа остальных субъектов. Он имеет доступ к регистрационному журналу для трассировки значимых событий. При определенных условиях (например, переполнение пространства хранения) администратор системы может удалить журнал,

но это будет сигналом об опасном событии для администратора безопасности. Формально администратор системы может дать себе права и на документ и т. п., но это также контролируется администратором безопасности путем просмотра настроек прав и регистрационного журнала.

- ❑ *Администратор безопасности (Контролер)* имеет доступ ко всем параметрам — но только на чтение. Таким образом, с одной стороны, он может контролировать все изменения на предмет их авторизованности (в том числе и в документах — с точки зрения бизнес-логики), но, с другой стороны, не может вмешаться в функционирование системы.
- ❑ *Пользователь 1-го уровня доступа* может создать электронный документ, но не может придать ему статуса актуальности, путем выставления флага "ОК" (или придать юридическую значимость установкой электронно-цифровой подписи — ЭЦП). При этом он может просмотреть, какие из созданных им документов уже актуализированы.
- ❑ *Пользователь 2-го уровня доступа* не может сам создать документ, однако уполномочен придать документу статус актуальности после его просмотра и проверки корректности формирования.
- ❑ *Начальник отдела* контролирует оба уровня пользователей (корректность формирования документа и своевременность его актуализации) и использует готовый документ в бизнес-целях (например, отправляет его руководству или партнерам).

Таким образом в данном примере — работе с электронным документом — достигнут высокий уровень безопасности, и при этом все значимые действия в системе находятся под децентрализованным контролем.

Сопоставление ролей классам обрабатываемой информации



С тем чтобы как специалисту, выполняющему классификацию системы, так и проверяющему лицу было проще понять, как выглядит распределение ролей и ответственности в привязке к классификации объектов, рассмотрим пример такой классификации. При этом необходимо будет вспомнить принятые в *главе 6* сокращения по категориям информационных объектов.

- ❑ Для объектов уровня Д0.

Администраторы. Обязательное географически распределенное хранение объекта (кластеризация, зеркалирование и т. п.), а также ежедневное резервное копирование на внешний носитель. Санкционированную возможность управления доступом к объекту должны иметь не более двух администраторов, при этом следует обеспечить возможность

аварийного администрирования объекта в случае отсутствия администраторов. Администраторы должны иметь широко известные средства коммуникационного контакта и возможность оперативного оповещения в случае сбоя в работе объекта. Доступ к внешнему носителю с резервной копии должен быть физически ограничен. Число субъектов, имеющих право доступа, не может превышать пяти. Факты использования резервной копии подлежат регистрации, так же как и все действия по управлению объектом.

Контролеры. Необходимо определить строгие и однозначные правила работы с объектом, процедуры для осуществления контроля и способы реакции на нарушения в работе объекта. Контроль действий должен производиться не реже одного раза в рабочий день. Предусматривается создание аварийных планов на случай временной или полной недоступности объекта.

Пользователи. Должны строго придерживаться правил работы с объектом, не пытаясь получать доступ к другим объектам, кроме предназначенного для них в соответствии с их функциональными обязанностями. При невозможности получения доступа к объекту им следует немедленно обратиться к администратору.

□ Для объектов уровня Д1.

Администраторы. Обязательное резервное копирование объекта на жесткий диск не реже трех раз в день и ежедневное копирование его на внешний носитель. Наличие холодного резервирования средств хранения и обработки информации. Санкционированную возможность управления доступом к объекту должны иметь не более двух администраторов, при этом должна быть обеспечена возможность аварийного администрирования объекта в случае отсутствия администраторов. Администраторам надлежит иметь широко известные средства коммуникационного контакта, а пользователям — возможность оперативного оповещения их в случае сбоя в работе объекта. Доступ к внешнему носителю с резервной копией должен быть физически контролируемым, случаи использования резервной копии следует регистрировать. Все действия по управлению объектом также должны регистрироваться.

Контролеры. Необходимо разработать строгие и однозначные правила работы с объектом, процедуры для осуществления контроля и способы реакции на нарушения в действиях объекта. Контроль действий производится не реже одного раза в рабочий день. Должны быть разработаны аварийные планы на случай временной или полной недоступности объекта.

Пользователи. Обязаны строго придерживаться правил работы с объектом, не пытаясь получать доступ к объектам помимо прав, определенных для них в соответствии с функциональными обязанностями. При невозможности получения доступа к объекту немедленно обратиться к администратору.

□ Для объектов уровня Д2.

Администраторы. Обязательное ежедневное резервное копирование на внешний носитель. Санкционированную возможность управления доступом к объекту должны иметь не более трех пользователей. Необходимо определить одного администратора. Следует обеспечить возможность аварийного администрирования. Доступ к внешнему носителю с резервной копией должен контролироваться. Действия по изменению размещения объекта или правил доступа к объекту подлежат регистрации.

Контролеры. Предусмотрены процедуры для осуществления контроля. Контроль действий должен производиться не реже одного раза в неделю.

Пользователи. Должны придерживаться правил работы с объектом, не пытаясь получать доступ к другому объекту, помимо определенного для них в соответствии с их функциональными обязанностями. При невозможности получения доступа к объекту следует обратиться к администратору или пользователям, управляющим доступом.

□ Для объектов уровня Д3.

Работа с полезной информацией оставляется на усмотрение пользователей информации.

□ Для объектов уровня Д4.

Работа с несущественной информацией не регламентируется.

□ Для объектов уровня Д5.

Администраторы. Необходимо обеспечить доступными средствами ограничение попадания информации данного класса пользователям.

Контролеры. Необходимо наличие регламента использования такой информации и процедур контроля за расходом пользователями ресурсов на информацию данного класса.

Пользователи. При обнаружении подобной информации пользователь должен принять меры к ее уничтожению и сообщить об этом администратору или контролеру.

□ Для объектов уровня Ц0 и Ц1.

Администраторы. Необходимо обеспечить право на модификацию только для авторизованных, идентифицируемых пользователей по предъявлению пароля. Предусматривается разделение прав на модификацию данных (внесение изменения/авторизация изменения) и ведение регистрационного журнала изменений. Для хранения и обработки объектов класса Ц0 предпочтительно использовать средства хранения, не включенные в общую компьютерную сеть.

Контролеры. Следует разработать строгие и однозначные правила модификации объекта, процедуры для осуществления контроля и способы реакции на несанкционированные модификации объекта. Проверка регистрационного журнала изменений должна осуществляться не реже одного раза в рабочий день.

Пользователи. Обязаны, с одной стороны, придерживаться правил работы с объектом, не пытаясь модифицировать объект вне рамок своих функциональных обязанностей, сохранять в тайне пароль на модификацию и не пытаться получить доступ к чужим паролям. С другой стороны, они должны осуществлять санкционированные модификации в соответствии с временным графиком и в рамках своих полномочий. На каждую модификацию информации данной категории должен быть документ в виде твердой копии соответствующей формы. При обнаружении модификации объекта, подозрительно похожей на несанкционированную, обязаны немедленно сообщить контролеру.

□ Для объектов уровня Ц2.

Администраторы. Должны обеспечить права на модификацию только для авторизованных, идентифицируемых пользователей по предъявлению пароля. Заполнять регистрационный журнал изменений.

Контролеры. Следует разработать правила модификации объекта, процедуры для осуществления контроля и способы реакции на несанкционированные модификации объекта. Проверка регистрационного журнала изменений должна осуществляться не реже двух раз в неделю.

Пользователи. Должны, с одной стороны, строго придерживаться правил работы с объектом, не пытаясь модифицировать объект вне рамок своих функциональных обязанностей, сохранять в тайне пароль на модификацию и не пытаться получить доступ к чужим паролям. С другой стороны, пользователи должны осуществлять санкционированные модификации строго в соответствии с временным графиком и в рамках своих полномочий. При обнаружении модификации объекта, подозрительно похожей на несанкционированную, немедленно сообщить контролеру.

□ Для объектов уровня Ц3.

Определение способов работы с объектами данного класса оставлено на усмотрение пользователей объекта.

□ Для объектов уровня Ц4.

Определение способов работы с объектами данного класса не регламентируется.

□ Для объектов уровня K0.

Запрещено хранение объектов данного типа в электронном виде. Необходимо обеспечить подписание пользователями объекта обязательства о неразглашении информации.

□ Для объектов уровня K1.

Работа с объектами данного типа возможна только на отдельно стоящих компьютерах или гарантированно защищенных от удаленного доступа. В этом случае пользователь информации является одновременно ее администратором и контролером, предоставляя ее для использования другим пользователям только в личном присутствии и под личным контролем. Необходимо обеспечить подписание пользователями обязательства о неразглашении информации.

□ Для объектов уровня K2:

Администраторы. Право на ознакомление с информацией предоставляется только авторизованным, идентифицируемым пользователям по предъявлению пароля и/или ключа на внешнем носителе. Необходимо заполнять регистрационный журнал доступа.

Контролеры. Следует разработать правила использования объекта, процедуры для осуществления контроля. Проверка регистрационного журнала изменений доступа должна осуществляться не реже одного раза в день. Необходимо обеспечить подписание пользователями обязательства о неразглашении информации.

Пользователи. Должны строго придерживаться правил работы с объектом, не пытаясь использовать объект вне рамок своих функциональных обязанностей, сохранять в тайне пароль на доступ и не пытаться получить доступ к чужим паролям.

□ Для объектов уровня K3.

В зависимости от специфики конкретной информации используются те же требования, что и для объектов уровня K2 или K4.

□ Для объектов уровня K4.

Определение способов работы с объектами данного класса возложено на пользователей объекта.

□ Для объектов уровня K5.

Определение способов работы с объектами данного класса не регламентируется.

Результаты классификации информационных систем

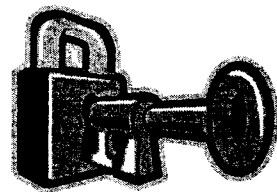
В результате произведенной классификации информационной системы в службе безопасности предприятия должен появиться документ "Классификатор информационной безопасности", содержащий следующие разделы:

- ☐ классификатор объектов;
- ☐ классификатор средств;
- ☐ классификатор субъектов;
- ☐ матрица разрешений, производящая сопоставление классов объектов с классами средств их обработки и классами субъектов, выполняющих эту обработку.

Весьма желательным дополнением к классификатору является наличие полного и актуального *списка кадрового состава* предприятия, к каждой строке которого добавлено поле "класс субъекта".

Первый раздел классификатора (посвященный информационным объектам) подвергается обновлению с появлением каждого нового типа обрабатываемой информации. Третий раздел модифицируется в случае введения качественно новых (в плане работы с информацией) должностей на предприятии. Второй и особенно четвертый разделы при правильном построении классификатора практически не должны претерпевать изменений. Как уже было отмечено, список кадрового состава должен обновляться незамедлительно по факту реальных перестановок в персонале и в идеальном случае стать образующим компонентом централизованной системы распределения ключей на предприятии.

Утверждение или хотя бы согласование "Классификатора информационной безопасности" должно производиться на уровне руководства предприятия.



Глава 8

Модель информационных потоков

В предыдущих главах, обращаясь к вопросам анализа информационных систем — обследования и классификации, мы старались говорить об информационном пространстве, в котором функционируют информационные системы, однако специально не концентрировали внимание на необходимости рассмотрения вопроса безопасности для всех систем пространства в целом. Из-за этого может сложиться впечатление, что общая безопасность пространства складывается из безопасности отдельных информационных систем — иногда говорят, что безопасность пространства в данном случае равна безопасности самой слабой из систем. Однако это верно только в том случае, когда среда передачи данных между отдельными системами рассматривается также как полноценная информационная система со своими свойствами и уязвимостями.

В ряде случаев, анализируя вопросы защищенности систем, упускаются из виду проблемы обмена данными между системами. Поэтому вопросу взаимодействия систем посвящена отдельная глава.

Направления защиты процессов обмена информацией



Те, кто изучал или принимал участие в процессе внедрения систем, возможно, сталкивался с ситуацией, когда система с тщательно продуманной моделью безопасности, с устойчивыми механизмами идентификации и аутентификации, с тонко настраиваемым доступом и правами, с хранением данных, защищенных сильными криптографическими средствами, допускает экспорт практически всех значимых данных (например, для некоторого круга пользователей, у которых есть право на подобный экспорт) в виде абсолютно незащищенного набора данных (обычного текстового файла). С точки зрения бизнес-логики в этом нет ничего противоестественного, система должна обмениваться данными с другими системами, которые могут ничего не знать о форматах данных исходной системы, о ее криптографических алгоритмах и тем более о способах обмена ключами шифрования

и т. д. С точки зрения самой системы также все в порядке — она надежно защищала данные внутри себя, предоставила право на экспорт только уполномоченному лицу. Однако с точки зрения комплексной безопасности всего информационного пространства возникает проблема — система освободила себя от забот по обеспечению безопасности! Кто теперь берет на себя эти обязанности?

Экспорт данных. В решении этой проблемы существуют два пути: *проактивный* и *реактивный*. Первый вариант возможен только тогда, когда соответствующие специалисты задумались о проблеме безопасности обмена данных до того, как была приобретена или разработана информационная система, которая должна обмениваться данными с другими системами и в которой отсутствуют механизмы обеспечения защищенного обмена данными. В этом случае есть возможность определить и выставить требование поддержки защищенного обмена данными для каждой устанавливаемой в информационном пространстве системы. При этих условиях исходная информационная система при экспорте (в этом случае некорректно говорить "экспорт" — речь идет об отправке данных в систему-получатель) не перестает обеспечивать защиту данных до тех пор, пока эту роль не возьмет на себя система-получатель. Таким образом, появляется возможность планировать общий уровень безопасности пространства при внедрении новых систем. Но возможно это только в том случае, когда служба информационной безопасности принимает участие в вопросах проектирования и внедрения систем на самом раннем этапе.

Второй вариант распространен, когда уже внедрено определенное число информационных систем, ничего не знающих друг о друге, и встает вопрос об обеспечении безопасности обмена данными между ними. Скорее всего красиво это сделать уже не удастся. Наилучший из возможных способов, когда система имеет встроенные возможности для дополнительной доработки — прикладные программные интерфейсы (англ. Application Programming Interface — API) или интерфейсы, ориентированные специально на обеспечение безопасности (англ. Security Services Provider Interface, в некоторых источниках — Security Support Provider Interface, — SSPI). Но такую возможность должна иметь каждая пара систем, обменивающихся данными. В этом случае с помощью своих разработчиков либо по дополнительному соглашению с поставщиком системы можно провести соответствующую модификацию. В противном случае придется разрабатывать дополнительные процедуры, в которых необходимо будет использовать возможности операционной системы, устройств передачи данных и каналов связи, дополнительные организационные мероприятия.

Это наиболее часто встречающаяся проблема, однако существуют и другие ответвления информационных потоков, которые не очевидны на первый взгляд и о которых не всегда задумываются. Если мы говорим именно об информационной безопасности, а не только о компьютерной, необходимо

прослеживать весь путь и жизненный цикл информации от того момента, как она покинула свой источник создания (например, человеческий мозг), до того, как она перестала существовать (например, потеряла актуальность в архиве). Проще всего это сделать, построив соответствующие схемы, тогда можно проследить, где информация оседает в виде распечаток, на дискетах и т. п., в местах, которые не были очевидны с первого взгляда.

Защита сетевых процессов обмена данными



Те, кто прочитал *раздел "Классификация средств обработки информации: стандарт CCITSE" главы 6*, возможно, обратили внимание на классы FTP и FCS, которые составляют для системы основу безопасного обмена данными с другими системами. Это как раз те механизмы обеспечения защиты обмена данными, когда информация не остается незащищенной ни на каком из этапов перемещения между системами. В основном, работа таких механизмов основана, конечно, на работе криптографических средств.

Если информационное пространство предприятия ориентировано на стек TCP/IP и есть возможность в случае необходимости дорабатывать системы, то в настоящее время с появлением новых механизмов и протоколов защиты обеспечивать безопасность такого пространства становится проще. В *главе 22*, посвященной сетевым протоколам безопасности, будут затронуты такие средства, как IPSec, ISAKMP, IKE, SSL и пр., которые делают обмен данными защищенным и работают одинаково вне зависимости от того, располагаются ли информационные системы в одной локальной сети или используют в качестве канала связи Интернет.

При этом независимо от того, используется ли в качестве сетевого протокола TCP/IP или другой стек протоколов, следует помнить о классической семиуровневой сетевой модели (OSI — Open System Interconnection). Это означает, что построение механизмов защиты должно также носить многоуровневый характер. Проще пояснить это на примерах. Если нас интересует только защита конфиденциальности и целостности данных в приложениях, то, обычно, этот вопрос можно решить на верхних уровнях (приложения или представления данных). Если встает вопрос об обеспечении надежной доставки — акценты смещаются к транспортному уровню. Если обмен информацией между системами должен скрывать внутреннюю сетевую структуру систем, то речь идет о сетевом уровне. Если необходимо учесть опасность широкоэвещательных сообщений (например, угрозы использования пассивного прослушивания сегмента), следует говорить о канальном уровне. Физический уровень обычно затрагивается, когда речь идет о защите от побочных электромагнитных излучениях или возможности физического внедрения злоумышленника в канал связи.

Когда в *главе 5* рассматривалась проблема инвентаризации информационных систем, одними из рекомендуемых ко включению в обследование вопросов были источники получения и цели отправления информации. Если это обследование было произведено качественно, то становится легко составить схему информационных потоков (причем не виртуально, а именно нарисовать на бумаге). Возможно, части этой схемы следует показать обследуемым и опрашиваемым пользователям для уточнения. Теперь, имея подобную схему, можно проанализировать тонкие места построенной или планируемой системы защиты данных.

Если обе системы, между которыми происходит обмен данными, поддерживают одни и те же протоколы безопасности, то при соответствующей настройке они смогут обеспечить требуемый уровень безопасности. А если одна из систем представляет собой "черный ящик" без возможности модификаций и не может импортировать/экспортировать никакие другие данные, кроме как в обычном текстовом или другом незащищенном формате?

Здесь необходимо подойти к вопросу оценки необходимости защиты с точки зрения классификации информации, используемой в системе, т. е. определить, к какой категории относится данная информация, какая составляющая для нее наиболее критична: конфиденциальность, целостность, доступность и т. д. Далее необходимо проанализировать, какие альтернативные средства защиты, предоставляемые системой, можно использовать. Возможно, система может рассчитывать некую контрольную сумму экспортируемых/импортируемых данных (аналог электронно-цифровой подписи), либо она ведет расширенный регистрационный журнал загруженных/выгруженных данных и т. п. Необходимо рассмотреть, как имеющиеся средства могут быть использованы второй системой.

Защита экспорта файлов



Если обмен данными ведется по самому примитивному варианту (одна система записывает текстовый файл в каталог на сервере, другая считывает его оттуда), возможно использовать дополнительные варианты защиты:

- *средствами операционной системы* (при условии, конечно, что сама операционная система поддерживает необходимые механизмы безопасности). Установить доступ на каталог импорта/экспорта только для специальных пользователей, от имени которых работают системы. Снизить до минимума или исключить возможность вмешательства администратора приложения или файлового сервера в работу систем. Возможно, предусмотреть дополнительный сервис, который бы брал на себя контроль за состоянием файла после его создания первой системой и уничтожение его после импорта второй системой. Создание средствами операционной системы или коммуникационными устройствами шифрованного или VLAN-канала между серверами приложения взаимодействующих систем и файловым сервером импорта/экспорта;

- *организационными мероприятиями.* Принудительное отключение пользователей от работы в сети на момент экспорта/импорта, запрет на использование анализаторов сетевого трафика даже для администраторов (в установленное время) и пр. Эти мероприятия могут несколько укрепить общую безопасность работы, но в силу своего "заплаточного" характера не являются надежными.

Вопросы построения полной и функциональной схемы информационных потоков



При построении схемы информационных потоков следует задуматься о возможности миграции информации по категориям классификации в зависимости от того, на какой стадии информационного потока она находится. Наиболее простой пример таков. Ряд пользователей или систем, каждый сам по себе, обладает частью информации, не представляющей особой важности и находящейся в низкой категории классификации. Та же информация, собранная в одной системе вместе, составляет критически важные данные, подлежащие особой защите, требующие размещения в верхней части таблицы классификации. Обратный пример — часть критически важной по конфиденциальности информации, может быть разглашена на определенном этапе, без ущерба для общего набора информации.

Таким образом, необходимо проводить оценку безопасности информации с точки зрения ее жизненных циклов и других условий. При построении информационного потока следует учитывать возможные ответвления от него, не являющиеся очевидными на первый взгляд. Приведем два примера.

- Информация следует от системы А через Б к В, на первый взгляд вся цепочка надежно защищена. Однако выясняется, что в системе Б пользователь распечатывает часть данных и передает другому пользователю для анализа. Поскольку данное ответвление находится вне электронного взаимодействия, оно может быть упущено, между тем если информация критична по конфиденциальности, необходимо применение соответствующих дополнительных мер.
- Информация также следует от системы А через Б к В. На этапе Б, перед тем как перейти к В, информация может быть изменена пользователем на основе других данных (например, на основе информации, поступившей от системы Б2). Как информационный поток защищен от ошибок в системе Б2, если система Б2 — это производственное совещание?

Возможно, примеры несколько надуманы, но наша цель — показать возможность участия неучтенных внешних факторов в потоке.

Средства создания схем информационных потоков



Теперь несколько слов собственно о создании схем информационных потоков. Здесь затруднительно предложить общую методику по причине специфики информационных систем и в значительной степени бизнес-процессов (и бизнес-потоков) в конкретном предприятии. Вполне возможно использование уже имеющегося механизма составления таких схем — универсального языка моделирования UML (Unified Modeling Language). Это достаточно сложный инструмент, для серьезной работы с ним может потребоваться квалифицированный специалист. Однако для профессионалов, знакомых с объектно-ориентированной концепцией, не составит труда использовать элементы UML при составлении схем.

При использовании UML или любого другого инструмента построения моделей следует помнить, что результирующая, рабочая схема должна описывать не абстрактную "безопасность", а конкретные ее категории. Соответственно, оперируя параметрами и функциями объектов схемы, необходимо разделять их на отвечающие за конфиденциальность, целостность и т. д. Если вспомнить о необходимости многоуровневого анализа построения защиты, о котором упоминалось в разделе *"Защита сетевых процессов обмена данными"* этой главы, то становится ясно, что построение качественной схемы — задача отнюдь не тривиальная.

Контроль построения модели информационных потоков



Убедиться, что все информационные потоки определены правильно и информация во время всего следования защищена надежно — задача, сравнимая по сложности с самим построением схемы информационных потоков. Если все информационные системы обладают возможностью защиты передаваемых данных, то убедиться в правильности построения схемы защиты для неспециалиста будет затруднительно: для этого необходимо анализировать техническую документацию к системам и конфигурацию работающих систем.

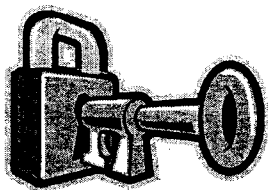
Однако, если одна из систем была спроектирована до 1990 года или в начале 90-х годов, и после этого модификаций ядра или модели безопасности не проводилось — высока вероятность того, что система может воспринимать и выдавать только простые текстовые файлы. В этом случае необходимо дополнительно выяснить, как производится защита этого файла в промежуток времени после выгрузки из одной системы до загрузки в другую.

Если никаких дополнительных мероприятий не предусмотрено, то следует задать следующие вопросы.

- ☐ Как защищен этот файл от постороннего прочтения?
- ☐ Как производится проверка того, что данные не были изменены непосредственно перед загрузкой в систему?
- ☐ Что происходит, если выгруженный файл был испорчен или удален?

В зависимости от ответов на них, можно представить картину мероприятий по организации защиты данных при перемещении от одной системы к другой.

В качестве тестового испытания полноты созданной модели, наиболее простого для выполнения со стороны менеджера проекта или руководителя отдела, можно порекомендовать следующую "проверку". Соберите все штатные (не тестовые) распечатки обрабатываемой системой информации, которые случайно найдутся у вас на столе или в сейфе. Затем исследуйте модель потоков, созданную специалистами по информационной безопасности, и проверьте, все ли виды распечаток, имеющиеся у вас, отражены в ней как ответвления информационных потоков.



Часть III

От чего защищать

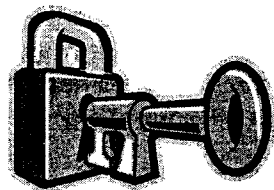
Глава 9. Типовая модель нападения

Глава 10. Локальные атаки

Глава 11. Удаленные атаки

Глава 12. Атаки на поток данных

Глава 9



Типовая модель нападения

Классификация атак

Локальной атакой будем называть случай, когда злоумышленник оказался непосредственно перед клавиатурой (дисководом, CD-ROM и т. п.) данного компьютера.

Удаленная атака — это вариант атаки, когда злоумышленник не видит (и возможно никогда не увидит) ту рабочую станцию (или сервер), с которой он работает. При этом сам атакуемый компьютер, возможно, не проявляет никакой сетевой активности.

Атака на поток данных — инцидент, когда атакуемый компьютер (компьютеры) активно отправляет, принимает или обменивается данными с другими компьютерами сети, локальной или глобальной, а местом приложения атакующего воздействия является сегмент сети или сетевой узел между этими системами.

Схематически три вида атак представлены на рис. 9.1.

Немного о хакерах

Возможно или нет взломать данную конкретную компьютерную систему? Большой вопрос для многих специалистов и руководителей. Среди профессионалов есть мнение, что если два компьютера соединены между собой кабелем, значит, с одного из них можно взломать другой. Но это теоретически. Практически все зависит от множества факторов.

А кто этот таинственный взломщик, которого все боятся?

Многие книги по информационной безопасности посвящают одну из первых глав разъяснению различий между словами *хакер* и *кракер* (еще, быть может, *фрикер*). Терминология различается приблизительно следующим образом: *хакер* — идейный борец за свободу информации, вторгающийся в чужие системы в основном из интереса, без прямой материальной заинтересованности; *кракер* — как раз тот, кто взламывает чужие системы, преследуя собственный финансовый интерес. Таким образом, все те, кого мы привыкли называть хакерами — на самом деле кракеры. *Фрикеры* — злоумышлен-

ники, использующие в собственных интересах уязвимости в телефонных системах. Тот, кому когда-либо удалось бесплатно позвонить по междугородному телефону из-за сбоя на АТС, — фрикер.

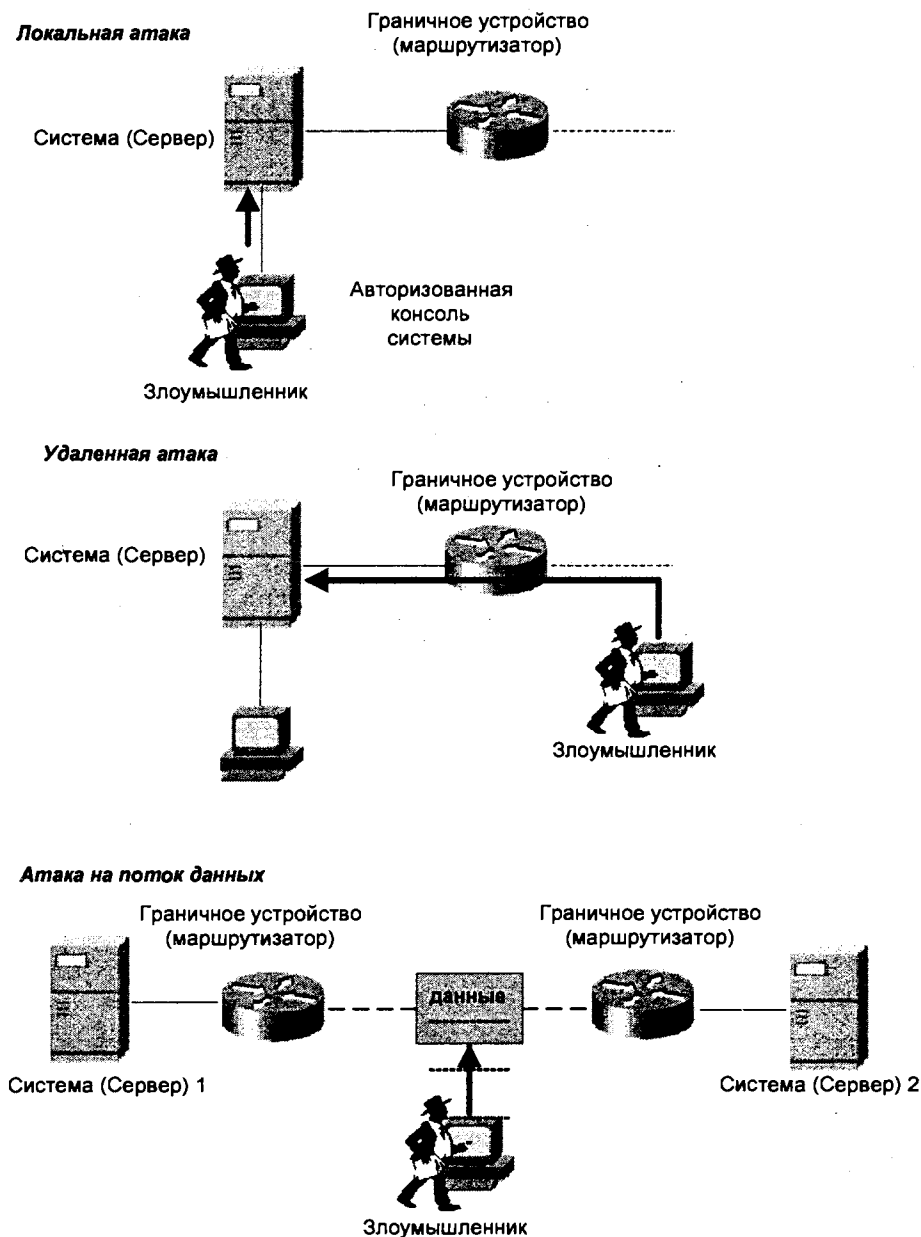


Рис. 9.1. Классификация атак

На наш взгляд, необходимо провести несколько иную классификацию. В нее попадут только те, кто наносит своими действиями ущерб (не важно, материальный или моральный), вне зависимости от того, какие цели он преследует при этом.

В первую категорию попадут серьезные исследователи компьютерных систем, изучающие операционные системы и прикладные программы на нижнем уровне с отладчиком или дизассемблером, создающие системные драйверы, способные самостоятельно переписать большую часть ядра операционной системы и умеющие много других сложных вещей. Именно специалисты такого уровня пополняют архивы программного обеспечения для взлома на "хакерских" сайтах и находят большую часть "дыр" в системах. При этом необходимо признать, что эти люди в подавляющем большинстве не преследуют материальную выгоду.

Во вторую категорию попадут те, кто имеет представление о работе информационных систем в объеме, достаточном лишь для того, чтобы использовать имеющиеся знания в своих корыстных интересах. Они применяют на практике программное обеспечение с "хакерских" сайтов, используют найденные другими уязвимости в системах и, чаще всего, преследуют при этом личную выгоду. Иногда для их обозначения используется термин *script kiddie* (вольный перевод с англ. — "ребенок с пистолетом").

Очевидно, что наибольшую опасность представляет первая категория, поскольку они используют уязвимости и средства, о которых, может быть, еще не известно специалистам по безопасности. Единственное утешение в том, что их немного и их интересы нацелены преимущественно на крупные учреждения, где получение выгоды может быть максимальным. Основной защитой от подобных злоумышленников может служить только комплексный подход, т. е. бороться не с конкретной уязвимостью системы, о которой вы узнали, а как минимум с классом уязвимостей. А еще правильнее — строить эшелонированную, модульную оборону, безотносительно атак, а известные уязвимости использовать для отслеживания тенденций появления новых типов атак.

Бороться со вторым типом проще, так как подобные злоумышленники используют уже известные уязвимости и средства. Сложность в том, что таких специалистов гораздо больше. Парочка из них может даже находиться среди наиболее продвинутых пользователей вашей собственной организации. Находятся иногда и любители проверить устойчивость собственной системы безопасности. Подобные исследования должны быть блокированы внутренними нормативными документами организации.

Вообще представление о том, что хакер — это худой длинноволосый молодой человек в очках, проводящий ночи у монитора в клубах сигаретного дыма и в окружении пивных бутылок, не всегда соответствует действительности. В одном из пресс-релизов спецслужб говорилось, что наиболее опасным хакером может стать соседский школьник или бабушка-пенсионерка из дома напротив (второе, конечно, более характерно для Запада). Словом,

это любой человек, у которого есть необходимое оборудование (компьютер и модем) и достаточно свободного времени.

Типовая атака на систему

Рассмотрим, как происходит типовая, абстрактная хакерская атака. Не будем пока говорить о серьезных профессиональных методах с засылкой шпионов в организацию, анализе содержания мусорных корзин компании или методах социальной инженерии. Просто один любопытствующий субъект с одной стороны земного шара хочет узнать, что есть интересного на серверах компании за океаном.

Первое, о чем должен задуматься хакер — об отсутствии обратной связи, т. е. о необходимости обеспечить себе условия, когда владелец атакуемой им системы (или тот, кто будет действовать в его интересах) не сможет узнать ни адреса, ни тем более личности того, кто произвел атаку. Простой периодической сменой IP-адреса тут не обойдешься. Утилиты обратного отслеживания укажут путь минимум до провайдера атакующего, а там, используя регистрационные журналы сервиса, можно довольно быстро разыскать злоумышленника. Необходимы другие методы. Первый — физический, т. е. периодически менять физическое место атаки, например, переходя из одного интернет-кафе в другое, оставаясь анонимным клиентом. Но этот метод хорош только для очень краткосрочной работы (несколько дней, а иногда и часов), так как довольно быстро можно определить страну и город хакера, а дальше в дело вступят оперативные мероприятия соответствующих служб — переезжать же постоянно из города в город довольно дорого.

Второй метод — логический. На безбрежных просторах Интернета существуют сервера (так называемые *прокси-службы*, или *анонимайзеры*), предоставляющие свои адреса для анонимной работы в Интернете. Для работы с таким сервером достаточно знать его адрес и, атакуя удаленную систему, указать специальным образом ее адрес в сочетании с адресом прокси-сервиса. Анонимайзер скроет настоящий адрес хакера, поместив в соответствующее поле свой адрес. Таким образом, в атакуемой системе возможно будет увидеть только адрес прокси-службы, реально находящейся от хакера в тысячах километрах. Используя в атакующем запросе последовательно несколько анонимайзеров можно хорошо спрятать свой собственный адрес. Единственным утешением является то, что использование анонимайзеров возможно не для всех сетевых протоколов.

Однако, если хакер узнал адрес прокси-службы, значит, узнать его может и специалист по безопасности. Периодически внося подобные сервера в "черный" список своего межсетевого экрана или маршрутизатора, можно отсеять все возможные атаки с указанного адреса.

Кроме того, правоохранительные органы и государственные службы в состоянии отслеживать использование анонимайзеров (а также многих других сервисов, предоставляемых пользователям услуги Интернета удаленно) и кардинально другим способом (вопрос только в затратах на оборудование и желании). Дело в том, что при всем многообразии самостоятельных и псевдосамостоятельных провайдеров, топология сети Интернет представляет собой паутину только относительно нескольких центральных узлов. Все периферийное виртуальное пространство имеет древовидную структуру, "расползающуюся" от центра, т. е. достаточно установить оборудование мониторинга всего в нескольких коммутационных узлах, чтобы получить возможность исследовать подавляющий объем трафика сети. Для России подобным объектом является московский узел обмена трафиком между провайдерами "М9"¹.

Естественно, объем кросс-коммутируемого трафика даже на российском узле представляет огромные значения, и на самом деле возникает вопрос о технической возможности процесса фильтрации и логирования трафика, параллельного основной маршрутизации. Однако поддержка журналов, состоящих только из IP-адресов/номеров портов источника и получателя трафика, времени установки и завершения сессии, вполне реальна. Сетевые стандарты для журналов подобного типа разработаны уже давно (например, протокол NetFlow от Cisco Systems). Они спроектированы таким образом, что их создание отнимает очень небольшие вычислительные ресурсы маршрутизатора.

Пусть известно, что компания К была взломана с адреса А с использованием средства С. Если предположить, что имеется доступ ко всему многообразию электронных журналов, можно сделать следующие выборки: кто за последние дни (недели) интересовался сайтом К, кто скачивал (или использовал) средство С и кто использовал прокси-сервис А. Сравнение результатов может существенно снизить область поисков.

Однако допустим, что хакер достаточно смел, чтобы все-таки попробовать начать атаку. Что еще ему нужно для начала? Адрес веб-сервера или шлюза организации? Возможно, но скорее всего там будут сконцентрированы наибольшие средства защиты компании. Быть может, пойти немного другим путем?

Если известен список телефонов организации (или хотя бы диапазон номеров), можно попробовать узнать, не подключен ли кто-нибудь из них к модему. Для этого необходима программа типа war-dialer, которая будет методически обзванивать телефоны по указанному списку, отмечая номера, где ответит модем. Запустив программу на ночь, утром хакер получит список номеров модемов организации. К сожалению, именно модемам служба

¹ "М9" — сокращение было порождено от ММТС-9 (Московская международная телефонная станция), которое применяется и телефонистами. Затем название М-9 было присвоено узлу обмена интернет-трафиком М9.

безопасности часто уделяет недостаточно внимания, оставляя их включенными во вне рабочее время, в режиме автоответа и т. д.

Возможно обойтись и без сканирования коммутируемых каналов, если известен конкретный адрес, на котором обнаружена уязвимость. Второй подход — прямой перебор диапазона IP-адресов, третий — выяснение соответствия конкретных адресов конкретным организациям с использованием *поисковых систем* (англ. *search engine*), а также известных сетевых служб и утилит (whois, nslookup)

Необходимым шагом любой неспонтанной атаки является выяснение топологии атакуемой сети. Помогут в этом злоумышленнику утилиты трассировки сетевых маршрутов, а также программы генерации фальшивых пакетов. Первые смогут выявить расположение маршрутизаторов в сетях, не защищенных межсетевыми экранами, и даже в сетях за МСЭ, если в их настройках разрешен протокол ICMP. Вторые могут проходить через межсетевые экраны и даже NAT-узлы при неправильной их конфигурации.

Очень популярным средством удаленного изучения сетей является протокол SNMP. Многие администраторы, ответственные за настройку сетевого взаимодействия в системе, довольно легкомысленно относятся к его активности. Результатом этого становится огромный объем информации, собранный потенциальным злоумышленником о сети удаленно, без каких-либо усилий. Подробнее протокол SNMP описан в *главе 11*.

Следующим шагом исследований злоумышленника скорее всего станет сканирование портов. Обычно оно применимо к сетям, работающим на основе TCP/IP. Как известно, сокет (IP-адрес + номер порта) определяет службу, которая запущена на данном компьютере, т. е. наличие ответа на попытку TCP соединения по данному порту означает функционирование сервиса или демона, запущенного на атакуемом узле. Таким образом, можно собрать данные о службах, запущенных на узле-жертве и, возможно, об операционной системе и ее версии. Для сканирования портов существует специальное программное обеспечение, которое работает без участия человека (например, ночью) и может быть легко загружено с известных хакерских сайтов.

Допустим, модем работает всего лишь в режиме приема-передачи файлов на уровне, ниже сетевого, не используя схему TCP/IP. Можно ли из этого сделать вывод, что его работа не представляет опасности для организации? Следует просто задуматься о том, что некое устройство (в данном случае модем с соответствующим программным обеспечением), основываясь на своих алгоритмах (известных в том числе и злоумышленникам), не всегда учитывая вопросы безопасности, складывает на компьютер в информационной сети предприятия (а возможно и отправляет во вне) некую информацию. Это происходит автоматически, без участия человека. Существует ли стопроцентная уверенность, что программное обеспечение для модема разработано без уязвимостей?

Авторская ремарка

У специалистов всегда вызывают улыбку эпизоды, когда в художественном фильме хакер, садясь за чужой компьютер, начинает взламывать систему, усиленно "колотя" по клавиатуре. Не будем говорить о том, что компьютер, хранящий секреты, должен быть как минимум защищен от атаки перебором пароля (хотя бы временным блокированием счета). Сценаристам следует заставить хакера хотя бы вставить дискету или компакт-диск. Реальный взлом системы — это чаще всего не борьба человека с компьютером, а противостояние одной программы другой (причем безо всяких графических этюдов на мониторе). Правда, на широком экране это выглядело бы не так эффектно.

Теперь, когда атакующий знает, что и как работает в атакуемой системе, начинается наиболее интеллектуальная часть работы хакера. Многие сервисы и операционные системы имеют известные уязвимости и, если администратор или специалист по безопасности атакуемой сети не позаботился об их устранении, то атакующий сможет ими воспользоваться.

Например, большинство операционных систем имеет установленные производителем административные счета с большими правами, с известными паролями по умолчанию. Если никто не позаботился об их удалении или отключении (или хотя бы изменении пароля), ими можно воспользоваться. Другой широко известный пример — установка полного доступа для всех на каждый новый *разделяемый* (англ. *shared*) каталог в сети в Windows 9x и NT-системах.

Кроме этого, прикладные программы могут иметь известные уязвимости. Примеры можно приводить бесконечно, начиная от уязвимости программы sendmail, использованной еще червем Морриса, до спецсимволов в URL для некоторых интернет-браузеров.

Если хакеру стало известно *имя пользователя* (англ. *username*), он может предпринять атаку перебором пароля. При отсутствии мер защиты от таких атак соответствующая программа может также без участия человека, например за ночь, произвести подбор пароля, т. е. пресловутый взлом. Известны западные статистические исследования, которые подтверждают, что в группе из 500 пользователей найдется минимум один, использовавший в качестве пароля слова "suzi", "qwerty" или "123".

Когда атакующий получил доступ на данный компьютер хотя бы с минимальными правами, он будет пытаться расширить свои права до максимальных. Здесь будут использованы и программы получения привилегий администратора из прав обычного пользователя. В некоторых системах существует возможность запоминания паролей и, если когда-либо на данном компьютере работал администратор (например, устранял проблемы в работе пользователя), и не позаботился о том, чтобы убрать за собой файл пароля, тогда он окажется в распоряжении хакера. Можно попытаться скопировать и базу паролей пользователей сети (если она не защищена). Хотя пароли и хранятся в зашифрованном виде, тем не менее в некоторых системах с по-

мощью специальных программ при наличии времени и вычислительных мощностей можно восстановить пароли в исходном виде.

Когда атакующий получил необходимый доступ, он может не только скопировать себе интересующие его данные, но и перестроить существующую сеть по своему желанию. Например, отключить сервисы безопасности, создать дополнительные счета пользователей с широким спектром прав, разместить троянскую программу и пр., вплоть до полного разрушения данных программ. В тех случаях, когда захваченная машина является всего лишь шагом на пути к другой более заманчивой цели, очень высока вероятность установки злоумышленником на машине *программы-сниффера* (англ. *sniffer* — "вынюхиватель"). Выполненный с ее помощью перехват и анализ трафика локальной сети — один из самых интересных для атакующего каналов сведений об атакуемой системе. И самое главное — злоумышленник с высокой степенью вероятности позаботится об уничтожении своих следов, т. е. сотрет соответствующие записи о вторжении в регистрационных журналах.

После такой деятельности администраторам сети останется только заново переустановить не только основные сервера, но и все пользовательские компьютеры, так как различные программы-шпионы и логические бомбы могут быть скрыты где угодно.

Надеемся, что все, о чем шла речь, будет воспринято не как инструкция к взлому, а как пособие специалистам по безопасности о неких минимальных мерах, которые должны быть предприняты в "родной" сети. Краткий свод мер, необходимых для безопасности сети:

- ☐ периодически проверять телефонные номера своей компании на предмет включенных модемов;
- ☐ устанавливать модемы, сконфигурированные максимально безопасно;
- ☐ производить анализ присутствия процесса сканирования портов;
- ☐ удалять или блокировать счета, существующие по умолчанию;
- ☐ устранять уязвимости систем и программ имеющимися заплатками и обновлениями;
- ☐ внедрить парольную политику с требованиями к сложности паролей и ограничениями к попыткам ввода некорректных паролей;
- ☐ наделять пользователей правами не большими, чем им нужно для работы;
- ☐ с осторожностью относиться к использованию административного счета (вплоть до смены пароля после каждого использования);
- ☐ защитить базу паролей пользователей от копирования ее рядовыми пользователями;
- ☐ вести учет программного обеспечения (системного и прикладного) и его конфигураций на компьютерах пользователей;

- постоянно делать резервные копии данных и сохранять инсталляционные копии программ;
- обеспечить сохранение регистрационных журналов в режиме, исключающем возможность их корректировки.

Дополнительные возможности атак "изнутри"

Рассмотренная типовая атака была инициирована извне предприятия. Между тем, как известно, большинство взломов происходит при участии самих сотрудников предприятия. Типовая атака в этом случае будет происходить не так увлекательно и, скорее всего, сведется к попыткам раскрытия пароля пользователя, имеющего соответствующий доступ к интересующей системе.

Технические средства, описанные в этой главе в *разд. "Типовая атака на систему"*, вроде программ перебора паролей, или снифферов, могут быть использованы и здесь. Однако атаки "изнутри" характеризуются одним очень серьезным фактором, который отсутствует в варианте удаленной атаки, а именно — возможностью воздействия на самого пользователя. Атака, при которой злоумышленник сам активно воздействует на пользователя с целью узнать его пароль, будет рассмотрена в *главе 10 в разд. "Социальная инженерия"*. В нашем случае наиболее интересен и, к сожалению, весьма распространен вариант, когда пользователь сам предлагает свой пароль, прямо или косвенно.

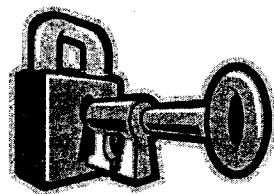
"Косвенно" — это когда пользователь вроде и скрывает свой пароль, но делает это довольно неумело — например, записывает в блокнот (который затем забывает на столе) или консультируется с коллегами при выборе пароля ("а могу я использовать дату рождения?").

Ну а "прямо" — это когда пользователь сообщает свой пароль другому сотруднику по различным причинам. Например, в ситуации, когда сотрудник должен срочно покинуть рабочее место и хочет, чтобы какую-либо операцию в системе сделали от его имени. Или он заболел, а начальник, звоня по телефону, требует пароль, чтобы срочно извлечь информацию, доступную только данному пользователю.

Поскольку в основе защиты на доступе по паролю лежит принцип самостоятельного сохранения пароля в секрете самим пользователем, то бороться с проявлениями таких атак можно только обучением пользователей и постоянным разъяснением им важности сохранения пароля в тайне. В тех же случаях, когда удастся установить факт передачи пароля одного пользователя другому, необходимо предпринимать административные меры с доведением информации об инциденте до остальных пользователей сети.

В реальной жизни вероятнее всего атаки (быть может, только за исключением атак на отказ в обслуживании) будут совмещать оба описанных сценария — что-то удастся выяснить локально, что-то сделать удаленно.

Глава 10



Локальные атаки

Для определения наиболее вероятных (или наиболее часто реализуемых) атак на информационную безопасность необходимо установить, на каких, не подкрепленных реальными средствами теоретических принципах, построена модель безопасности. Например, если одним из ключевых положений является то, что к компьютеру сможет физически приблизиться только уполномоченный человек, и при этом не установлено никаких систем ограничения физического доступа — значит, наиболее вероятны атаки именно на физическую безопасность.

Социальная инженерия

Атака под условным названием *социальная инженерия* отнесена нами к разряду локальных (хотя она может быть использована и удаленно), поскольку злоумышленнику все же необходим некий минимальный контакт с представителем атакуемой системы, а именно — пользователем.

Термином *социальная инженерия* обычно называют набор мероприятий по сбору сведений об информационной системе, напрямую не связанный с техническими подробностями реализации системы, а основанный на человеческом факторе. Наиболее простым, даже утрированным примером можно назвать случай, когда злоумышленник, представившись пользователю уполномоченным лицом (администратором сети или большим начальником), вынуждает пользователя сообщить свой пароль. Если это удастся, злоумышленник получает возможность доступа к системе без особых знаний о принципах ее устройства и без применения специальных технических средств.

Общий смысл методов социальной инженерии основан на психологических методах с использованием таких качеств человека, как необоснованное доверие (к не идентифицированному надежно человеку), лень (перепроверить полученные данные), невнимательность (к явным признакам злого умысла) и прочих слабостей.

Без сомнения, все пользователи сети должны быть предупреждены о возможности применения к ним методов социальной инженерии, но это не всегда достаточно. В практике авторов известно множество случаев, когда,

только что прочитав инструкцию и подписав обязательство о неразглашении пароля с серьезными угрозами за нарушение, пользователь тем не менее в ответ на просьбу администратора зарегистрировать пароль, вместо того чтобы ввести его с клавиатуры, произносит его громко вслух в присутствии нескольких людей, некоторых из которых он видит впервые в жизни. Методы борьбы с этим могут быть разными, например материальное наказание за известный случай разглашения пароля (даже если это не повлекло серьезных последствий) с оповещением о данном эпизоде остальных пользователей.

Существует тем не менее целый ряд ситуаций, потенциально опасных для случаев применения таких методов. Например, следующая ситуация: пользователи сети географически распределены по разным территориальным участкам (разные здания в пределах города), а администрирование, в частности управление учетными записями пользователей, осуществляется централизованно. Пользователи периодически обращаются по телефону к администраторам с вопросами смены пароля (забыл, заблокировал и т. п.). Как поступать администратору, когда он не видит пользователя и не может однозначно идентифицировать его по голосу? Для решения этой проблемы администратору или специалисту по безопасности необходимо продумать варианты возможных угроз и методов противодействия им.

В данном случае угрозы могут быть следующими.

- ❑ Злоумышленник, представившись пользователем, сообщит, что забыл пароль и попросит установить временный пароль для входа в систему, с тем чтобы после этого изменить временный пароль на новый.
- ❑ Если администратор использует для всех случаев один и тот же временный пароль (например, "123456"), злоумышленник может периодически перебирать учетные записи пользователей с этим паролем, в надежде поймать ситуацию, когда пользователю установлен временный пароль.

Хотя вторая ситуация менее вероятна, ее также не следует упускать из вида, если злоумышленник, скажем, присутствовал при разговоре пользователя с администратором по телефону об установке временного пароля, причем после этого пользователь не изменил пароль сразу же.

Допустимыми мерами защиты в данном случае могут быть следующие:

- ❑ сохранять в учетной записи пользователя информации, которую сможет прочитать администратор и которая известна только пользователю (номер домашнего телефона, девичья фамилия матери и т. п.) — перед сменой пароля администратор должен проверить знание пользователем данной информации;
- ❑ сверять с регистрационным журналом дату и время последнего использования учетной записи (входа в систему) — перед сменой пароля запросить у пользователя указанную дату и время;

- ❑ не использовать один и тот же временный пароль, слегка изменяя его для каждого раза (например, не "123456", а "k123456hh");
- ❑ перезванивать пользователю на его рабочий телефон после его просьбы о необходимости смены пароля и продолжать дальнейший диалог, уже удостоверившись в номере телефона, а, следовательно, и месторасположении;
- ❑ не сообщать пользователю о временном пароле по тому же каналу связи, по которому пришел запрос на смену пароля, т. е., если пользователь обращается по телефону, то временный пароль направить ему по внутренней электронной почте;
- ❑ сообщать временный пароль не самому пользователю, а его непосредственному руководителю, с тем чтобы руководитель передал временный пароль уполномоченному пользователю.

Комбинацией этих и других возможных мероприятий можно существенно снизить риск использования уязвимостей данной ситуации. Однако все эти меры носят лишь вероятностный характер защиты — при хорошем знании злоумышленником порядка дел в организации и наличии специальных технических средств все они могут быть преодолены.

Таким образом, угрозы использования социальной инженерии должны быть учтены наравне с остальными угрозами информационной безопасности, необходимо разрабатывать и внедрять соответствующие методы и средства противодействия, обучать персонал и т. д.

Закладки в аппаратном обеспечении



Большинство информационных систем (возможно, за исключением учреждений с повышенным уровнем секретности) функционируют исходя из того, что *аппаратное обеспечение* не представляет собой угрозу. При этом не производится даже начальной, не говоря уже о регулярной, проверки на наличие аппаратных закладок. Напомним, что *закладкой* называется логическое, в данном случае аппаратное, устройство, которое выполняет некоторые недокументированные или недеklarированные функции обычно в ущерб пользователю данной информационной системы. В рассматриваемой ситуации такая закладка может накапливать и передавать соответствующему субъекту информацию о системе, в которой функционирует: от статистических данных о работе системы до паролей пользователей или собственно данных системы.

Очевидно, что не все организации обладают необходимым штатом специалистов по электронике, способных выявить аппаратные закладки. Для обеспечения первоначальной уверенности в отсутствии таких закладок на новом

приобретенном оборудовании придется полагаться на наличие сертификатов уполномоченных организаций по данному производителю или поставщику, по классу оборудования и по другим параметрам. Для обеспечения более высокого уровня доверия можно пригласить специалистов из соответствующих организаций для выборочной или полной проверки поставляемого оборудования.

Обеспечение регулярной проверки также возможно различными способами, в зависимости от требований к конкретному классу информации или средств работы с ней и от материальных и других ресурсов предприятия. Например, методы проверки могут быть следующими.

- ❑ Периодическая проверка оборудования приглашенными специалистами уполномоченных организаций.
- ❑ Фиксация серийных номеров комплектующих частей оборудования с регулярной проверкой соответствия реально установленных и зафиксированных номеров. Дополнительная работа заключается в перерегистрации данных при проведении ремонта, замены и т. п.
- ❑ Автоматизированная инвентаризация элементов аппаратного обеспечения в информационном пространстве предприятия. Эту функцию обычно выполняет специальное программное обеспечение, поддерживающее базу данных по аппаратному обеспечению, используемому на предприятии.
- ❑ Опечатывание разборных частей корпусов оборудования с регулярной проверкой целостности печатей. В этом случае основной акцент переносится на пользователя данной единицы оборудования, который должен регулярно (обычно ежедневно перед началом работы) проверять целостность печатей.

В более серьезных случаях возможны затраты на оборудование, выполняющее постоянный мониторинг, либо даже фильтрацию/подавление возможных сред передачи информации вовне организации. Традиционно к данным средам относят (по убыванию частоты использования): радиоэфир, сети проводной передачи данных, сети питания вычислительных машин, видимый/инфракрасный диапазон, звуковой фон. Вопрос эффективного обнаружения и активного/пассивного противодействия аппаратным закладкам на самом деле занимает не одну книгу. В решении данной проблемы авторы отсылают заинтересовавшихся читателей к многочисленным русскоязычным изданиям.

Вообще в данном вопросе, как впрочем и в других, важным элементом обеспечения безопасности является обучение пользователей, которые должны сигнализировать в службу информационной безопасности о любом предопределенном или подозрительном событии, в том числе и произошедшем с аппаратным обеспечением. В качестве примера можно привести известную атаку на банкоматы для получения информации о ПИН (персональном

идентификационном номере) для доступа к пластиковой карте. Если заранее не быть информированным о такой атаке, злоумышленник может достичь успеха. Атака заключалась в нанесении злоумышленником тонкого слоя пыли на ПИН-клавиатуру. Когда пользователь набирал свой ПИН, на клавиатуре оставались следы, которые затем анализировались злоумышленником и могли привести к разгадыванию ПИНа пользователя.

Мы рассмотрели только атаки, направленные на получение доступа к информации (на конфиденциальность и целостность), однако говоря об атаках на аппаратное обеспечение нельзя забывать и об атаках на доступность, когда само оборудование может быть похищено или разрушено, прекратив, таким образом, функционирование всей системы или ее части. О защите от таких атак разговор пойдет в *главе 31*.

Преодоление ограничений доступа на уровне *firmware*



Теперь допустим, что надежность работы аппаратного обеспечения достигнута. Рассмотрим возможности встроенного программного обеспечения (англ. *firmware*). Известно, что иногда в разграничении доступа к конкретному компьютеру система безопасности полагается на *пароль на загрузку* самого компьютера (иногда называемого *паролем BIOS*).

На наш взгляд, данное средство можно использовать только как дополнительное усиление в системе контроля доступа, так как данный метод обладает рядом недостатков.

- ❑ Сброс пароля в случае выключения или разрядки батареи постоянного питания компьютера (батареи часов).
- ❑ Хранение некоторыми производителями *firmware* пароля в виде преобразованного набора символов (контрольного значения), причем данное преобразование имеет небольшой фиксированный набор значений. В практике авторов известен случай, когда два различных слова, используемых в качестве пароля, имели одно и то же контрольное значение. Причем первое слово было мастер-паролем технической службы для доступа в настройки BIOS всех пользовательских компьютеров, а второе слово — пароль на загрузку компьютера одного из пользователей. Этот пользователь был приятно удивлен, обнаружив, что его парольное слово не только позволяет загружать все остальные пользовательские компьютеры, но и дает возможность управлять конфигурацией загрузки. Предприятие потратило существенное время на замену мастер-пароля на всех компьютерах пользователей.
- ❑ Наличие у некоторых производителей недокументированного универсального пароля для входа в систему с целью отладки аппаратного обеспечения. Широкий резонанс получил случай с недокументированным

паролем "AWARD_SW" в огромной серии BIOS для IBM PC AT фирмы Award Software, Inc.

Кроме того, при использовании BIOS-паролей следует различать пароль на загрузку самого компьютера и пароль на доступ к конфигурационным данным загрузки. Это должны быть различные пароли, для того чтобы рядовой пользователь данного компьютера (как впрочем и потенциальный злоумышленник, завладевший паролем пользователя) не смог изменить параметры загрузки в свою пользу, как, например, для атак, описанных в следующем разделе.

Получение доступа на этапе загрузки ОС



При нормально функционирующем аппаратном обеспечении в процессе загрузки компьютера наступает момент, когда firmware передает управление компьютером операционной системе. Однако при соответствующей настройке последовательности загрузки на многих ЭВМ возможна передача первоначального обращения не к *стационарному* (жесткий диск, сетевая плата), а к *внешнему носителю* (дискете, CD-ROM диску). Если в этот момент за рабочим местом пользователя находится злоумышленник, он может инициировать загрузку с внешнего носителя операционной системы, в которой он обладает правами, достаточными для того, чтобы получить доступ к информации (в том числе и системной) на жестком диске компьютера пользователя. Проведя необходимые изменения (например, заменив или скорректировав базу данных по счетам пользователей данного компьютера или установив программную закладку — троянскую программу), он сможет при следующей загрузке компьютера иметь полные права на данной рабочей станции (или сервере, если он находился за консолью сервера).

Таким образом, службе безопасности следует рассмотреть вопрос о целесообразности использования возможности загрузки операционной системы с внешнего носителя и, при отрицательном решении, изменить конфигурацию компьютера соответствующим образом.

Вообще говоря, вопрос об использовании устройств считывания информации с внешних носителей (а также с неиспользуемых физических портов компьютера) ставится несколько шире, чем возможность загрузки операционной системы. Перечисленные источники получения информации являются широко известными способами перенесения на компьютер (а также копирования с него) информации, полностью находящимися в распоряжении пользователя, при этом их затруднительно контролировать.

В настоящее время считается хорошей практикой отключение или даже физическое удаление возможности считывания с внешнего носителя у пользо-

вателя. При этом имеет смысл только общее отключение, так как если оставлять такую возможность у одного-двух пользователей на подразделение предприятия (отдел, департамент и т. п.), то рано или поздно эти уполномоченные пользователи будут переписывать информацию для всех остальных, практически не контролируя ее. В таком случае имеет смысл создания отдельной службы информационного шлюза для внешних носителей — одного, двух специалистов, размещенных поблизости от входа на предприятие. При этом все сотрудники и посетители должны быть осведомлены о наличии такой службы, для того чтобы все внешние носители с информацией передавать в указанную службу. Там информация должна проверяться по ряду параметров (например, на отсутствие вирусов для входящих или на ненарушение конфиденциальности исходящих), после чего отправляться адресату — например, по электронной почте пользователю предприятия — для входящих, на внешний носитель — для исходящих. Естественно, что отключение локальных съемных устройств ввода/вывода имеет смысл только при наличии компьютерной сети.

Атаки на средства аутентификации



Управление передано операционной системе. Своевременным требованием к возможности продолжения пользователем работы является его *идентификация* и *аутентификация* (то есть сотрудник должен представиться системе и подтвердить, что действительно тот, кем представился). Обычно не бывает идентификации без аутентификации, поэтому далее будем говорить только о последней процедуре. Ряд вариантов аутентификации может быть отнесен на более ранний этап загрузки компьютера, до активации операционной системы. В разделе они приведены для единообразия классификации.

Обычно аутентификация основывается на одном из следующих параметров или их комбинации:

- ☐ что-то, что пользователь знает (пароль);
- ☐ что-то, что пользователь имеет (токен);
- ☐ что-то, чем пользователь является (биометрические параметры).

Самым распространенным является в настоящее время первый вариант, достаточно надежным считается комбинация первого со вторым (предъявление токена и ввод пароля), возможно, с развитием технологий в конечном итоге использоваться будет третий, как наиболее удобный для пользователя.

Рассмотрим их от более сложного и менее распространенного к более простому и чаще используемому.

Биометрические средства аутентификации

Говоря об этом классе средств, необходимо учитывать их точность работы и анализировать в комплексе со следующими характеристиками.

- ❑ *Уровень ошибочного отказа* — процент случаев, когда корректный предъявленный аутентификатор отвергнут из-за особенностей процесса обработки.
- ❑ *Уровень ошибочного подтверждения* — процент случаев, когда некорректно предъявленный аутентификатор принят из-за особенностей процесса обработки.
- ❑ *Скорость обработки аутентификатора* — анализ считанного изображения может занять долгое время на низкоскоростной технике.
- ❑ *Устойчивость к подмене*. Из видеофильмов (надеемся, никому из читателей не пришлось столкнуться с этим на практике) известны случаи с отрезанием пальцев и других частей тела авторизованных пользователей злоумышленниками для предъявления их аутентифицирующим устройствам. Кроме того, для отпечатка пальца, например, может быть создана резиновая копия.
- ❑ *Требования к хранению данных* — хранение аутентификационных данных в виде многопараметрического вектора или подробного изображения может потребовать значительных ресурсов.
- ❑ *Прочие условия*. Использование многими пользователями одного устройства с прикосновением к нему может быть неприемлемо для людей с иными культурными традициями или с физиологическими/психическими отклонениями.

Наиболее известными методами биометрической аутентификации являются: отпечаток пальца или ладони, геометрия ладони (длина, ширина, вес), параметры голоса, анализ сетчатки или радужной оболочки глаза, динамика подписи, геометрия и рисунок теплового излучения лица.

Токены

Токен (англ. *token*) — это устройство, хранящее некий уникальный параметр, на основе которого выдается корректный ответ на запрос системы об аутентификации.

Различают следующие варианты использования токена.

- ❑ На запрос системы токен предъявляет ей хранимое секретное значение. Это не самый надежный случай, так как, перехватив это значение один раз, злоумышленник может имитировать ответ токена.
- ❑ Токен и система имеют общую, синхронизированную систему генерации одноразовых паролей. На запрос системы токен выдает пароль, действи-

тельный для данного промежутка времени. Синхронизированная система генерирует в это время свой вариант пароля, который и сравнивает с полученным.

- ❑ Токен зарегистрирован в системе, и система знает его секретное значение (либо открытый ключ для варианта асимметричной криптографии — *более подробно об этом рассказывается в главе 18*). Получив в запросе некую случайную величину, сгенерированную системой для данного сеанса аутентификации, токен преобразует ее, используя свой секретный параметр. Таким образом, с одной стороны, информация, предоставляемая для аутентификации (как запрос, так и ответ), каждый раз различна и ее перехват ничего не дает злоумышленнику. С другой стороны, система, зная случайное значение и секретный параметр токена, генерирует свой вариант ожидаемого ответа токена и на основе этого сравнения принимает решение об авторизации. По сравнению с предыдущим методом данный алгоритм стоек к рассинхронизации системы и токена. При случайных или умышленных попытках некорректной авторизации либо даже простейшем сбое питания в методе с синхронной системой счетчики одноразовых паролей могут рассинхронизироваться. Это потребует вмешательства в работу системы специалистов с дополнительными полномочиями.

Варианты применения токена совместно с паролем или ПИНом пользователя следующие:

- ❑ ПИН служит паролем доступа к самому токenu: без введения ПИН токен не действует;
- ❑ токен генерирует аутентификационный ответ на основе своего секретного параметра и ПИН пользователя, т. е. корректность и секретного параметра и ПИНа анализируются системой.
- ❑ ПИН пользователя совместно с параметром токена и случайной величиной системы, или с синхронизированным параметром служат основой для выработки одноразового пароля, который затем сообщается пользователю. Пользователь далее работает с этим паролем, как с обычным, но только в ограниченный промежуток времени.

Надежность подобных систем зависит от реализации работы токена и системы. Например, если между системой и устройством считывания данных, подготовленных токеном, может размещаться средство перехвата информации, и при этом срок жизни пароля из ответа токена достаточно долгий, то, естественно, он может быть использован злоумышленником.

Пароли

Это наиболее распространенный в данное время способ аутентификации, когда система приглашает пользователя-человека ввести известную только данному пользователю (и системе) последовательность символов.

Возможные атаки в данном случае основываются на уязвимости:

- ☐ способа доставки пароля от пользователя к системе;
- ☐ способа хранения пароля в системе;
- ☐ системной политики работы с паролем;
- ☐ самого пароля вследствие его некорректного выбора пользователем.

Способы доставки пароля от пользователя до аутентифицирующей системы определяются протоколами доставки, которые будут рассмотрены в *главах 20 и 22*. На первый взгляд в разговоре о локальных атаках не имеет смысла говорить о способах доставки пароля до системы, однако на самом деле это зависит от способа аутентификации пользователя в системе. Если аутентификация пользователя происходит на той же системе, с консоли которой он работает, то, действительно, в данном случае рассмотрение способов доставки можно опустить. Если же перед работой на локальной станции пользователь должен быть аутентифицирован на отдельном сервере, то процесс доставки пароля до сервера необходимо также учитывать.

Единственное, что можно добавить к рекомендациям по усилению безопасности способа доставки паролей — это своевременно отслеживать появления обновлений и заплаток. Так, после снятия в начале 2000 года правительством США ограничений на экспорт сильных криптографических средств, стало возможным использовать протокол безопасности SSL с ключом длиной 128 бит вместо 40-битного, который был до этого. Снятие такого ограничения, возможно, означает, что соответствующие органы в США теперь обладают ресурсами, позволяющими взламывать 128-битные ключи в этой криптосистеме, но тем не менее при прочих равных условиях лучше использовать более сильные ключи.

Способы хранения паролей в системе зависят от самой системы и обычно не могут быть изменены, например, усилены с точки зрения безопасности. Тем не менее в ряде случаев, например, для Unix-подобных операционных систем, возможны дополнительные мероприятия по усилению защиты файла паролей (изменение прав доступа, shadowing). Для других систем, например, Windows NT, сам способ хранения паролей не может быть усилен на месте использования (только производителем с помощью соответствующих заплат). Однако необходимо учитывать дополнительные уязвимости, связанные с файлом паролей, такие как возможность его сохранения в альтернативном месте на случай аварийного восстановления. Например, если альтернативное место хранения не обеспечивает необходимого уровня защиты, файл паролей может быть скопирован злоумышленником и далее подвергнут обработке на предмет извлечения паролей пользователей.

В любом случае необходимо ознакомиться с документацией системы, посвященной вопросам хранения паролей, а также обязательно изучить накоп-

ленный опыт по вопросам несанкционированного извлечения паролей в данной системе.

Уязвимость *системной политики паролей* зависит от настройки, установленной администратором. Если политика безопасности допускает неограниченное число попыток ввода некорректного пароля без блокировки имени регистрации, если нет задержки между попытками ввода, если возможен пароль размером менее 8—10 символов или вообще пустой пароль, то пароль пользователя не истекает никогда, не поддерживается история паролей — все это несомненные возможности для злоумышленника. Для снижения вероятности реализации такой атаки необходимо установить блокировку счета пользователя (хотя бы временную) после 3—5 ошибок при вводе пароля либо после ввода неверного пароля установить запаздывание 5—7 секунд; запретить вводить пароли маленького размера, установить приемлемое для пользователей время истечения пароля (в среднем 1 месяц), поддерживать историю паролей — т. е. запрещать пользователю использовать свои старые пароли. Продвинутые системы могут производить анализ паролей пользователей при вводе и отбраковывать простые, легко разгадываемые пароли.

Если система ведет регистрационный журнал ввода паролей пользователями, можно использовать дополнительные возможности — расследовать часто повторяющиеся случаи ввода неверного пароля пользователем, блокировать счет, если пользователь не входил в систему более 2 недель — месяца и пр.

Отдельный вопрос, который должен быть рассмотрен в данном разделе, это *имитация системного приглашения* к вводу пароля со стороны посторонней программы (программной закладки). Идея такой атаки состоит в том, чтобы перехватить запрос пользователя на доступ к системе. В момент начала процесса авторизации запускается альтернативная программа, созданная злоумышленником. В тот момент, когда пользователь думает, что вводит пароль в соответствующее окно запроса системы, информация на самом деле передается в окно программной закладки. После этого альтернативная программа либо выдает сообщение об ошибке пароля и передает управление настоящей системе, либо, если позволяют технические возможности, прозрачно для пользователя эмулирует ввод пароля в адрес настоящей системы. Если в системе не предусмотрены дополнительные защитные механизмы (генерирование прерывания комбинацией клавиш <Ctrl>+<Alt>+<Del> в Windows NT), то защита от таких атак — обучение пользователей, которые должны внимательно следить за поведением системы при вводе пароля. Изменение надписи приглашения к вводу, цвета заставки, отклонение от обычной последовательности действий или реакций — все это должно вызывать подозрение.

Атака на слабость паролей — это уже притча во языцех. Известно, что примитивно выбранный пароль можно разгадать даже без использования технических средств, более сложные — с использованием известных программ-взломщиков паролей. Конечно, речь должна идти в первую очередь

об обучении и инструктировании пользователей. Если на предприятии не используется практика, когда хорошие пароли генерируются для пользователей программным образом, то целесообразно придерживаться следующих советов.

- ❑ Не принимать в качестве паролей коротких последовательностей символов, меньше 8—10; максимально использовать предоставляемые системой возможности и вводить парольные последовательности в 15—20 символов.
- ❑ Обязательно использовать в пароле не только буквы и цифры, но и спецсимволы (!"№;%:~*{[<>? и т. п.) и изменение регистра символов (с нажатием клавиши <Shift>). Не следует думать, что достаточно набирать в качестве пароля на клавиатуре, включенной в режим латинского шрифта, фразу на русском языке, ориентируясь на русскую раскладку. К сожалению, существуют уже словари для программ-взломщиков паролей, ориентированные и на такие хитрости.
- ❑ Если у пользователя проблемы с запоминанием паролей, можно рекомендовать в качестве пароля известную ему, запоминающуюся фразу или первые строки литературного произведения, перемежая его спецсимволами, в латинской раскладке, со сменой регистров.

Пример (который, конечно же, следует использовать в качестве методики, а не копировать дословно).

Фраза: "31 декабря 1978 года мы пошли в баню" — цифр в ней и так достаточно, будем набирать ее в латинской раскладке, группируя слова спецсимволами и последние три слова в верхнем регистре. Получаем:

"(31-ltrf,hz)+[1978=ujlf] vs&GJIKB D"<FY>""

что вполне может считаться хорошим паролем.

Дополнительно, если система хранения паролей имеет такую техническую возможность, на предприятии можно внедрить практику, когда уполномоченные администраторы или специалисты по информационной безопасности анализируют пароли пользователей с помощью соответствующего программного обеспечения в течение фиксированного промежутка времени (например, считаются приемлемыми пароли, которые не удалось взломать в течение трех дней).

В этом случае необходимо определить:

- ❑ процедуру, препятствующую злоупотреблению взломанным паролем со стороны проверяющего;
- ❑ ответственность пользователя за некачественно выбранный пароль, который был взломан — ведь выбор такого пароля объективно представляет угрозу для предприятия.

Атаки класса "повышение привилегий"

Другая категория возможных локальных атак заключается в выполнении пользователем, получившим доступ в систему (не важно авторизованным пользователем или злоумышленником, приобретшим доступ от имени пользователя) различных несанкционированных действий. Данный класс атак носит название *повышение привилегий*.

Наиболее примитивный вариант — это поиск информации, когда пользователь начинает исследовать доступные для него возможности в системе, которые напрямую не связаны с его рабочими обязанностями. Если права пользователя в системе настроены в соответствии с принципом минимальных привилегий, то такой поиск не приведет к угрозе для системы. Однако, если сама система не позволяет жестко ограничить пользователя, то он может получить дополнительную информацию, например, о системных настройках, системном "мусоре" или о данных, которые не должны быть ему доступны, и использовать их дальше в своих интересах. В качестве системного "мусора" в данном случае можно рассматривать дампы памяти, файлы с расширением `rw1` и пр., из чего можно извлечь информацию.

В рамках такого поиска пользователь может попытаться изменить системные настройки, если система это позволяет. Например, пользователь может случайно или намеренно сделать доступными для остальных пользователей сети свои локальные информационные ресурсы, предназначенные только для него. Кроме того, при специфической организации доступа к данным пользователь может получить к ним доступ вне системы разграничения прав. Например, если разграничение прав доступа к данным производит приложение, а при этом СУБД, хранящая данные, такого разграничения не производит. В этом случае пользователь, обратившись к данным вне приложения, может получить доступ к информации, для него не предназначенной.

Пользователь может использовать известные ему уязвимости операционной системы, хотя обычно это производится с помощью специального программного обеспечения и будет рассмотрено в *главе 23*.

Если невозможно техническое ограничение прав пользователя на поиск информации, необходимо предпринять соответствующие дополнительные меры, например, запретить такой поиск в приказном порядке. Иногда система, не давая возможности ограничивать доступ, тем не менее позволяет вести регистрацию такого доступа; подобные регистрационные журналы можно использовать для анализа несанкционированной активности пользователей.

Постороннее программное обеспечение



Более распространенный вариант, когда пользователь устанавливает на своем компьютере *постороннее программное обеспечение*. Он может сознательно устанавливать программное обеспечение, которое предназначено для исследования или взлома систем, либо неосознанно установить зловредную программу, результат может быть одинаково угрожающим. В этой ситуации:

- ❑ пользователь должен быть предупрежден о запрете установки стороннего программного обеспечения;
- ❑ служба информационной безопасности должна принимать меры к обнаружению и устранению таких программ.

Зловредные программы, устанавливаемые пользователем по незнанию, представляют собой скорее вариант удаленной атаки, так как пользователь компьютера не является ни автором, ни пользователем этой программы. В данном случае больший интерес представляют программы, устанавливаемые пользователем сознательно, чтобы использовать их в своих интересах.

Собственно, в зависимости от строгости порядков на предприятии, любую постороннюю программу — компьютерную игру, программу, обучающую иностранному языку, энциклопедию — можно считать зловредной. Во-первых, все они отвлекают пользователя в его рабочее время, за которое ему платит работодатель. Во-вторых, они расходуют ресурсы (место на диске, процессорное время и т. п.) системы, предоставленной пользователю для выполнения рабочих обязанностей.

Однако рассмотрим непосредственно так называемые "хакерские" программы. К ним следует отнести программы, предназначенные для повышения привилегий пользователя в данной локальной системе; анализаторы сетевой активности — sniffеры; локальные взломщики паролей и других защищенных ресурсов (шифрованных документов, таблиц, архивов); дизассемблеры и отладчики; конструкторы и генераторы вирусов и сетевых пакетов; генераторы номеров кредитных карт и др.

Перечислим известные возможности таких программ как примеры существующих атак.

- ❑ *Программа повышения прав* пользователя зависит от конкретной операционной системы и ее настроек. Позволяет пользователю с минимальными правами получить доступ к системным ресурсам с более высокими привилегиями, вплоть до привилегий администратора системы.
- ❑ *Сниффер* (англ. *sniffer*) — программа, перехватывающая пакеты, поступающие к данной станции, в том числе и те, которые станция при нормальной работе должна проигнорировать. Если сеть предприятия органи-

зована по принципу множественного доступа (то есть пакеты получают все станции, но обрабатывают только те, для кого они предназначены), то sniffер может накопить существенную информацию, так как до него могут прийти, например, авторизационные пакеты с паролями.

- ❑ *Программы подбора паролей* могут быть использованы пользователем на многопользовательских рабочих станциях с целью попытаться в фоновом режиме подобрать (например, по словарю) пароль другого пользователя этой же системы.
- ❑ *Взломыщики шифров* — программы, использующие различные криптоаналитические методы и известные уязвимости в криптографических алгоритмах для приведения данных в незашифрованное состояние без использования ключа дешифрования. Если в качестве зашифрованного файла для программы был использован, например, файл паролей локальных пользователей (в том числе администратора), то при успешной работе программы будут получены соответствующие пароли.
- ❑ *Дизассемблеры* (англ. *disassembler*) и *отладчики* (англ. *debugger*) позволяют провести анализ и пошаговое выполнение программного обеспечения (например, операционной системы или приложения) с тем, чтобы понять его внутреннюю логику и уязвимости или вызвать в его работе сбой с предсказуемым результатом, либо изменить ход работы программы в свою пользу.
- ❑ *Атаки на переполнение буфера* — когда поступающие в программу данные вызывают сбой либо проблемы с выдачей программой информации, которая должна быть скрыта, либо с выполнением ряда действий иначе, чем это было запланировано разработчиком программы.
- ❑ *Конструкторы и генераторы вирусов* позволяют соответственно создать программное обеспечение, которое в дальнейшем можно использовать для нанесения ущерба информационным системам.
- ❑ *Конструкторы и генераторы сетевых пакетов* позволяют создавать пакеты, отличные от создаваемых уполномоченной системой, например, с измененным адресом отправителя или некорректным размером. Применение таких программ может привести к сбоям информационных систем или ошибочным аутентификациям, если аутентификация производится по адресу отправителя.
- ❑ *Генераторы номеров кредитных карт* не являются угрозами локальной системе, они позволяют ввести в заблуждение удаленные системы, предоставляющие платные ресурсы путем предъявления к оплате несуществующих или чужих карт. Тем не менее предприятие, чей работник использует такой трюк, может понести ущерб в ходе проведения расследования атакованной стороной.

Кроме перечисленных программ, пользователь может использовать программу автоматизированного поиска уязвимостей системы. Хотя чаще всего

такие программы характерны для использования при удаленных атаках, тем не менее они могут быть использованы для анализа локальной системы. Подробнее они будут рассмотрены в *главе 23*.

Как следует из списка, сфера действий этих программ различна, поэтому вместо того, чтобы бороться со всеми возможными атаками от этих программ, проще заранее кардинально решить вопрос с запретом установки таких программ. Рекомендации в этом случае могут быть следующими.

- ❑ Пресечение источников проникновения постороннего программного обеспечения: блокировка внешних носителей, фильтрация входящего информационного трафика.
- ❑ Разграничение прав пользователей в системе таким образом, чтобы они не могли устанавливать программное обеспечение самостоятельно.
- ❑ Инвентаризация программного обеспечения, установленного у пользователей — регулярная автоматизированная проверка или выборочная инспекция.
- ❑ Подготовка соответствующих нормативов по правилам работы и ознакомление с ними пользователей.

Возможно, в ближайшее время или в далеком будущем появятся другие атаки. Необходимо быть постоянно осведомленным о таких возможностях, т. е. регулярно изучать соответствующие информационные ресурсы (специализированные книги, сайты в Интернете, подписка на рассылку и т. п.). Информацию о новых появившихся атаках следует обязательно анализировать применительно к собственной сети, при этом нельзя забывать о том, что потенциально исполнителем атаки может быть и сотрудник своего предприятия. При соблюдении этих мер служба информационной безопасности будет готова адекватно воздействовать в подобных ситуациях.

Пример комплексной локальной атаки



Для тех, кто хочет ознакомиться с описанными атаками подробнее, расскажем о примере, в котором использовалось около семи локальных атак.

Одним из известных опубликованных случаев применения социальной инженерии является рассказ Иры Винклер, директора по технологиям Национальной ассоциации компьютерной безопасности, о ее участии в тестовом проникновении в компанию с условным названием Zed Technologies. Данный случай интересен не столько сам по себе, сколько тем, что в нем описан целый спектр методов, которые может применить злоумышленник с целью овладения конфиденциальной информацией.

Для реализации своего плана Ира использовала пять видов атак: анализ открытых источников, выдача себя за другого сотрудника, использование авторизованного доступа для злоупотреблений, хакерские действия на информационные системы из локальной сети организации и из внешнего информационного пространства. Мы рассмотрим лишь те, которые относятся к социальной инженерии. Из архивов новостей она выяснила наиболее перспективные направления исследований в компании, финансовые затраты на проекты компании и объем потенциальных продаж. Узнала имя руководителя исследовательской группы, работающей над проектом, и получила краткую информацию о выпускаемых компанией продуктах и сотрудниках, участвовавших в их разработке. Из других открытых источников она установила имена руководителей подразделений компании, финансовое состояние компании и разнообразную общую информацию о компании и ее политике, организации работы сотрудников. Поиск в интернет-конференциях названия компании выявил имена ряда служащих компании. Письма служащих в группы новостей компьютерной тематики рассказали ей об оборудовании и программных средах, использующихся в компании. Письма в нетехнические группы помогли выявить области интересов служащих, посылавших эти сообщения. Газета, выпускаемая компанией, описывала шесть главных проектов и приводила имена большого числа сотрудников, работающих над этими проектами.

Далее, Ира оформилась на работу в Zed Technologies как временный специалист по контракту. Она не указывает, как это было сделано, но по ее дальнейшему рассказу, она могла поступить туда даже уборщицей. Перед прибытием на рабочее место она менее чем за день, используя реальную бизнес-карту как образец, сделала в местном магазине бизнес-карту, которая выглядела точно так же, как сделанная в Zed Technologies. В ней было указано имя Иры и должность — администратор информационной безопасности.

На работе, встретившись с одним из известных ключевых работников, она заявила, что недавно ее приняли на работу администратором по информационной безопасности, дала ему свою бизнес-карту и сказала, что ей поставлена задача защитить информацию компании. Ира попросила его детально описать, какая информация является критической и какие люди имеют к ней доступ, спросила, есть ли источник, из которого можно взять полную информацию о технологии производства. В ответ он показал копии заметок с рабочих совещаний группы и список рассылки, члены которой получают эти заметки. По просьбе Иры он не только дал ей копии всех заметок из книги, но даже добавил Иру в список рассылки.

В беседе с другим работником они обсудили сделанные его отделом документы, и Ира узнала о типах документов, местах хранения файлов в сети, группе ответственных за архивацию файлов и имя человека, ответственного за их хранение. Этот сотрудник даже упомянул один документ, содер-

жащий спецификацию технологии продукта. В записках рассылки Ира обнаружила сообщение, в котором указывалось местоположение черновиков документов, переданных правительству. В следующем предложении автор сообщения давал пароль для доступа к этому документу, в той же директории находились аналогичные документы о двух других приоритетных проектах компании. Ира использовала несколько типовых комбинаций в качестве пароля и одна из них оказалась правильной, она получила доступ к файлам. Как оказалось, каждый коммерческий менеджер отвечает за несколько проектов, поэтому файлы одного из менеджеров, содержали информацию о большом числе проектов.

Затем Ира узнала о том, что Zed Technologies использует смарт-карты для аутентификации доступа из внешнего информационного пространства. Она получила копию формы, используемой для заявки на получение смарт-карты, подделала подпись администратора информационной безопасности на ней и передала секретарю на утверждение.

Далее Ира попыталась задержаться на работе допоздна. Несколько уборщиков ходили по зданию, когда она начала искать незапертые шкафы, комнаты и ящики на столах и осматривать компьютеры, которые не были защищены с помощью устройств блокировки рабочих станций, требуемых согласно руководящим документам компании. Было невозможно избежать встреч с уборщиками, поэтому она не скрывала своего присутствия.

Она обнаружила два компьютера, оставленных без блокировки консоли. Два монитора были выключены, она просто включила один и обнаружила, что сотрудник все еще работает в программе электронной почты. При этом он сохранял старые письма. Просмотрев их, Ира нашла сообщение, содержащее основной график разработки, один из самых критических документов компании.

В своей работе Ира использовала ряд дополнительных технических средств типа автоматических анализаторов уязвимостей, таким образом она смогла похитить практически всю наиболее важную конфиденциальную информацию компании. По ее оценке причина этого в том, что компания сосредоточилась на защите от внешних атак. Как только атакующий получал статус сотрудника организации, меры защиты становились бесполезными, и информация подвергалась риску.

При этом следует учесть, что предварительно с ней было согласовано, что она не будет наносить вреда компании и сотрудникам. Если предположить, что в арсенале злоумышленника будут такие средства, как активное вмешательство в работу оборудования компании, установка подслушивающих устройств, запугивание, шантаж или физическое воздействие на сотрудников, то результат может быть достигнут даже при более серьезном подходе к безопасности для внутренних работников.

Утилиты локальных атак



Известная утилита, получения доступа к файловой системе Microsoft Windows NT — NTFS называется NTFSDOS. Думаем, что даже яростные противники Билла Гейтса согласятся, что с установленными соответствующими заплатками и исправлениями правильно сконфигурированная система Windows NT с корректно распределенными NTFS-правами и защищенным паролем администратора представляет из себя достаточно надежную систему. Но все это верно только до тех пор, пока у злоумышленника нет возможности произвести загрузку с дискеты (в данном случае операционной системы DOS) и запустить пресловутый драйвер файловой системы `ntfsdos.exe`. Другим вариантом может быть физическое похищение жесткого диска компьютера и перемещение его туда, где возможно произвести загрузку с другой операционной системы, также NT-подобной (где у злоумышленника уже есть соответствующие права) или другой операционной системы (аналог NTFSDOS существует и для Linux).

Когда утилита запущена, она сканирует разделы жестких дисков, имеющих на компьютере, и ищет NTFS-драйвер. Найдя его, утилита делает раздел NTFS доступным операционной системе DOS как логический диск, игнорируя при этом права доступа, которые были определены в файловой системе NTFS. Свободно распространяемая версия делает раздел NTFS доступным только для чтения, однако коммерческие версии и аналоги позволяют иметь полный доступ к файлам NTFS. Дополнительные ограничения свободных версий могут быть на размер диска (до 4 Мбайт), поддержку чередования дисков, отображения времени создания/модификации файлов и др.

Другая программа `GetAdmin.exe`, использующая уязвимость операционной системы Windows NT, позволяет добавлять пользователя (если только он не работает под учетной записью `guest`) в группу локальных администраторов. Реализация атаки стала возможна, так как функция `NtOpenProcess` (Получить доступ к процессу) не производила проверку соответствующих привилегий после изменения одного из флагов (`NtGlobalFlag`). При этом процесс `winlogon` с учетной записью **System** мог добавлять или удалять пользователя из группы администраторов. Используя эту программу, став локальным администратором, далее пользователь может производить любую перенастройку системы под свои интересы.

Программа взлома паролей Windows NT — `L0phtCrack` — использовала при атаке необходимость обратной совместимости различных версий аутентификационных протоколов. Взлом стал возможным из-за того, что более старая версия протокола LANManager использовала откровенно уязвимый ал-

горитм шифрования (если быть более точным — хэширования) паролей. Таким образом, злоумышленникам не было необходимости работать с более сильным механизмом защиты NT — они использовали более слабый вариант, оставленный для поддержки аутентификации старых версий. Пароли LANManager не производят различия между регистрами (строчная и заглавная буква рассматривается как одна и та же), что существенно сокращает количество вариантов паролей. Кроме того, стандартный пароль NT длиной в 14 символов LANManager разбивает на два по 7. А это при наличии возможности проверять хэш половинок пароля независимо (именно так и происходит на самом деле) позволяет злоумышленнику перебирать $67^7 + 67^7 (=2^{43})$, а не $67^{14} (=2^{84})$ вариантов пароля.

Чтобы раскрыть пароли, на первом шаге программа использует хэш LANManager. L0phtCrack обрабатывает список слов, используя при этом тот же алгоритм, что и LANManager, и сравнивает полученное значение с оригиналом из базы данных паролей SAM (Security Accounts Manager). Если устанавливается соответствие, то пароль раскрыт. Если программа доходит до конца словаря, то она начинает просмотр словаря заново, добавляя понемногу символов в начало и конец каждого словарного слова, и т. д. Вычислив значение хэша LANManager, L0phtCrack получает пароль без различия регистра. Теперь программа хэширует все варианты написания пароля (с буквами в разных регистрах) и сравнивает хэш-значения с исходными хэш-значениями для NT.

Специальная утилита из пакета NT4-SP3 Syskey обеспечивает дополнительную защиту, шифруя хэш-значения паролей из SAM. Однако поскольку в базе продолжают присутствовать и пароли для LANManager, программа может перенести акцент атаки на перехват пакетов с паролями, предназначенными для LANManager.

Предварительные выводы

Перечисленные примеры атак не являются новыми — они давно известны. Приведены они здесь только для того, чтобы показать, какие разнообразные причины могут послужить уязвимостями для систем. Новые виды атак появляются постоянно. Поэтому повторим еще раз основные рекомендации по противодействию им:

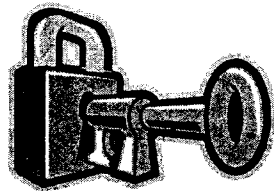
- ❑ стройте защиту на противостоянии не конкретной атаке, а классу атак;
- ❑ установите защиту на разных уровнях, так как новая атака, возможно, обойдет тот эшелон защиты, который был направлен против нее, но "споткнется" о другой эшелон соседнего класса (например, новый сверхвзломщик паролей, превосходящий имеющиеся заплатки и обнов-

ления системы, будет бесполезен, если сам файл паролей недоступен взломщику);

- ❑ будьте в курсе новых атак, о которых обычно информируют соответствующие организации, и своевременно устанавливайте обновления и заплатки на системы, выпускаемые производителем.

Более серьезным материалом на тему формирования парольной политики может служить документ CSC-STD-002-85 "Password Management Guide" Министерства обороны США [DOD2], одна из книг так называемой Радужной серии (Rainbow series), куда входят знаменитые "Оранжевая" [DOD1] и "Красная" [NCSC1] книги. В документе рассмотрены такие вопросы, как ответственность специалиста по безопасности, ответственность пользователя, функциональность механизма аутентификации и защита паролей. Кроме того, там предложены алгоритмы генерации и шифрования паролей, определение длины пароля, вероятности его угадывания и пр.

Глава 11



Удаленные атаки

Классификация удаленных атак



В нашей классификации удаленными атаками называются такие атаки, реализация которых не требует иных контактов злоумышленника с атакуемым объектом, кроме собственно средств реализации самой атаки. Здесь имеется в виду, что злоумышленник не обязан физически присутствовать во время атаки или перед ней на объекте или системе, вступать в контакты с ее пользователями и т. д. Теоретически он может вообще не иметь никаких знаний об объекте атаки до ее начала. Также подразумевается, что атакуемая система (для простоты — компьютер) имеет коммуникационное соединение какого-либо рода (чаще всего подключение к компьютерной сети либо устройство доступа "точка-точка" — модем) и на ней запущены соответствующие коммуникационные сервисы.

Мы постараемся не повторять там, где без этого возможно обойтись, элементы атак предыдущей главы (*глава 10*), хотя в реальной жизни они чаще всего будут совмещены. Например, предварительная обработка пользователя методами социальной инженерии, затем получение файла паролей путем удаленной атаки, локальный взлом паролей на станции злоумышленником и продолжение удаленной атаки с новыми знаниями.

Внимательные читатели, углубляясь в ознакомление с этой главой, возможно, отметят, что здесь классификация атак дана несколько иная, чем в *главе 10*. Разъясняя такую особенность, ответим, что на страницах этой книги специально не используется какой-либо определенный тип классификации, чтобы не выстраивать догм. Классификация должна адекватно отражать текущую ситуацию и служить удобству построения защиты. Хотелось бы указать возможные принципы построения классификаций, а не описывать одну, пусть даже хорошую модель.

По сценарию



С точки зрения атакующего разделим атаки следующим образом (данная классификация позволяет определить возможный сценарий атаки по имеющимся начальным признакам):

- ❑ Атаки без предварительного сбора информации об атакуемом объекте и без направленности на конкретный объект, т. е. злоумышленник не знает

ничего о потенциальной жертве (атакуемом объекте), более того, он даже может не знать, кто станет его жертвой. Злоумышленник готовит некий автоматизированный модуль (программу, скрипт, компоненту), которая распространяется им без какой-либо конкретной цели атаки. Данный модуль, попав к жертве, в условия, в которых он сможет функционировать, сообщит о результатах своей деятельности злоумышленнику, либо выполнит свою задачу автономно, без информирования создателя. К данному типу атак относятся распространение компьютерных вирусов и троянских программ, в случаях, когда они не учитывают специфику работы атакуемого (имеющиеся у него системы и службы, их особенности и т. п.), а носят некий более или менее универсальный характер.

- ❑ Атаки без предварительного сбора информации, но направленные на конкретный объект. В этом случае атакующий чаще всего использует некий имеющийся у него набор средств, каждое из которых направлено на использование конкретной уязвимости (набора уязвимостей). Злоумышленник последовательно применяет их к жертве в надежде, что одно из них сработает.
- ❑ Атаки с предварительным сбором информации. В этом случае атакующий производит предварительно анализ атакуемой системы, например, выясняет, какие службы запущены на ней путем сканирования портов. После этого он выбирает службу, уязвимую с наибольшей вероятностью или наиболее подходящую по имеющимся средствам.
- ❑ Атаки со сбором информации во время атаки. Здесь атакующий, обнаружив уязвимость у жертвы, немедленно производит атаку на данную уязвимость. В этом смысле процесс сбора информации и процесс атаки практически не разорваны во времени.

По цели



С точки зрения защиты, атаки можно рассматривать, как направленные на конкретные составляющие безопасности — конфиденциальность, целостность и доступность.

- ❑ *Атаки на доступность* являются наиболее просто реализуемыми хотя бы потому, что атаковать можно не саму целевую систему, а, например, устройства и средства коммуникации к ней. Кроме того, почти всегда проще сломать систему, чем забраться внутрь, и тем более заставить ее работать по своим правилам.
- ❑ *Атаки на конфиденциальность* сложнее в реализации и обнаружении. Так как информация из системы только копируется, оставляя саму систему в исходном состоянии (за исключением случаев, когда для успеха атаки на конфиденциальность необходимо провести атаку на целостность или доступность), обнаружить ее можно только если ведется подробный ре-

гистрационный журнал либо если система настроена на сигнализацию о несанкционированном или подозрительном доступе.

- *Атаки на целостность* могут различаться как атаки на целостность данных и на целостность самой системы (в последнем случае это может быть предварительной атакой перед атакой на конфиденциальность или доступность, или атакой на целостность данных). Наиболее неприятный случай — это успешная реализация атаки на целостность системы, так как в этом случае система начинает вести себя так, как нужно атакующему и, следовательно, перестает быть доверенной системой. Для возвращения ее в доверенное состояние приходится обычно производить переустановку системы заново. Однако и атака на целостность данных (вполне возможно — на целостность единственного документа) может принести к очень серьезным последствиям. Чаще всего в этом случае злоумышленник, как и в случае атаки на конфиденциальность, старается вести себя в атакованной системе как можно незаметнее, с тем чтобы внесенные им в документ изменения как можно дольше не были обнаружены и сыграли отведенную им роль.

По характеру взаимодействия с жертвой



Относительно характера взаимодействия злоумышленника с системой все удаленные атаки можно разделить на "интерактивные" и "безусловные".

- *Интерактивная атака* отличается тем, что в ходе ее хотя бы раз возникает ситуация, когда ее дальнейшее успешное продолжение зависит исключительно от того, правильные или неправильные действия предпримет пользователь или администратор атакуемой системы, т. е. приложение интерактивных атак на автоматические станции невозможно. Классическими примерами подобного класса атак является:

- приглашение зайти на определенную страничку в Интернете, в теле которой злоумышленник заблаговременно разместил зловредную программу, автоматически выполняющуюся на компьютере, с которого идет ее просмотр (а ошибок в программном обеспечении (ПО), позволяющих это сделать, на сегодняшний день обнаружено порядка нескольких десятков);
- письмо, с приложенной к нему зловредной программой, но имеющей какое-либо деловое или "соблазнительное" название и описание;
- сообщение, адресованное пользователям якобы от администратора системы с требованием установить на своем компьютере новую антивирусную программу, выложенную им на каком-либо файл-сервере, и которая на самом деле является ни чем иным, как троянской программой.

Интерактивные удаленные атаки неразрывно связаны с социальной инженерией, описанным ранее в *главе 10*.

- *Безусловная атака* совершенно не зависит от действий (или бездействия) оператора. Ее успех определяется только характеристиками установленного в системе аппаратного и программного обеспечения и его настройками.

Удаленный сбор информации о жертве

Выяснение адресов



Известно, что самый надежный корабль — это тот, который стоит на берегу и никогда не выходит в море. Так же и самая безопасная информационная система — это компьютер, не имеющий никаких сетевых соединений, выключенный и запертый в сейф.

Но как только информационная система начинает функционировать, готовиться обмениваться информацией с другими системами (напомним, что пока рассматривается пассивная система, которая сама не иницирует отправку данных куда-либо вовне), она становится уязвимой. В большем или меньшем смысле, для определения этого и существует информационная безопасность.

Примечание

Основным стандартом сетевой работы де-факто стал набор механизмов и протоколов, обозначаемых аббревиатурой TCP/IP, поэтому в основном речь идет именно об этой сетевой сфере. Но общие принципы могут быть прозрачно или с небольшими изменениями перенесены и на другие типы сетей.

Службы или сервисы, работающие в информационной системе, используют и предоставляют какие-либо информационные ресурсы, которые также могут быть использованы внешним субъектом, в том числе и несанкционированно. Для этого злоумышленнику необходимо иметь данные о том, какие службы/сервисы работают в системе. Но первое, что он должен узнать о самой системе — ее сетевой адрес.

Получение сетевого адреса (IP-адреса) или имени домена/хоста потенциальной жертвы уже дает злоумышленнику информацию, на основе которой он может начинать строить план атаки. Он может получить дополнительно альтернативным путем (без непосредственного обращения к системам жертвы) ряд сведений о ней, например, используя службу WHOIS. Дело в том, что, регистрируясь в Глобальной информационной сети, организация (например, ее провайдер) должна предоставить определенную информацию о себе — название, местоположение, возможно, имена и адреса электронной почты администраторов и т. п. Эту информацию можно найти, обратившись

в соответствующую службу Глобальной сети. А это уже может послужить основой для формирования, например, списка типовых паролей на основе имен/фамилий, районов города, где расположена организация, городских телефонов и т. д.

Сетевой адрес — это параметр, уникально идентифицирующий хост в пределах сети (локальной или глобальной), и именно его нужно указывать автоматизированной программе, которая займется сбором информации о службах, запущенных на исследуемой системе.

Поиск уязвимостей



Поскольку на конкретной машине может быть (и почти всегда бывает) запущено более одного сервиса, то возникает необходимость после поступления пакета в адрес хоста определить, какой из служб он направлен. Эта функция возложена на операционную систему, которая анализирует небольшой числовой параметр транспортного уровня — *логический порт*, специально размещаемый в пакете отправляющей стороной для этой цели. Так как за распределение пакетов между службами ответственна операционная система, то она же и занимается простейшими служебными сообщениями по поводу того, функционирует ли в данный момент тот или иной логический порт или нет. При этом о некоторых служебных пакетах (например, для протокола TCP о самом первом пакете — *SYN* — *запросе на соединение*) операционная система заботится сама. Поэтому удаленный субъект, исследующий систему, всегда имеет возможность получить полную информацию о запущенных на ней службах, даже если потенциально имелась бы возможность их скрывать — например, отвечать на запрос только в том случае, если его аутентичность проверена с помощью криптографической подписи.

За огромным количеством различных служб интернет-сообществом закреплены фиксированные значения логических портов по умолчанию. Например, WWW-сервер, будучи установлен на любой машине, по умолчанию будет ожидать запросов на порт 80. Наиболее известные пары "номер логического порта/служба" для протокола TCP следующие: 21 — FTP, 23 — Telnet, 25 — SMTP, 80 — HTTP, 110 — POP3. Более подробный список приведен в *приложении 1*.

Например, если такое исследование обнаружило подтверждение установления связи с хостом — потенциальной жертвой по порту 23, то с высокой степенью вероятности можно утверждать, что там функционирует сервис Telnet, который обладает для злоумышленника рядом привлекательных свойств. Например, позволяет выполнять команды в режиме удаленного терминала и не шифрует никакие данные (в том числе и пароли) при передаче. Почему речь идет только о высокой степени вероятности работы сервиса? Потому, что на самом деле хост-жертва может быть лишь *приманкой* для взломщика (англ. *honeypot*), и вместо реального Telnet там функционирует программа-имитатор.

Автоматизированные программы поиска уязвимостей разнятся по своей функциональности и назначению. Наиболее простые — это действительно просто сканеры портов, которые последовательно пытаются установить связь с удаленным хостом по имеющемуся списку портов и выдают отчет о выявленных открытых портах. Более проработанные программы могут на основе информации, хранимой в своей базе знаний, пытаться обнаружить параметры, указывающие на то, что та или иная известная уязвимость в конкретной службе не была устранена. Наиболее ужасные (для жертвы) программы могут при обнаружении признаков уязвимости инициировать соответствующую атаку вплоть до ее полного успешного завершения. Необходимо добавить, что такие программы бывают свободно распространяемыми и коммерческими.

Специалист по безопасности должен быть осведомлен о таких программах (их последних модификациях), тем более что независимые эксперты периодически проводят их сравнения по различным параметрам и публикуют результаты в изданиях, посвященных безопасности, в том числе и в Интернете.

До этого момента разговор шел только о наличии работающей службы на атакуемом хосте, но на самом деле очевидно, что успех атаки зависит в значительной степени от того, как реализована работа службы:

- ❑ как написано программное обеспечение, реализующее работу службы;
- ❑ какие настройки работы программного обеспечения установлены.

Указанное означает, что помимо потенциальных ошибок, заложенных в самой природе работы протокола или стандарта, уязвимости могут скрываться в некорректной реализации или настройке, т. е., говоря, например, о FTP-сервере, необходимо отметить не только уязвимость самого протокола FTP, но и то, какое конкретно программное обеспечение поддерживает этот сервер, а также, какие настройки сервера установлены — например, разрешены ли доступ анонимным пользователям или команда PUT.

Составление полного списка известных уязвимостей практически невозможно, поскольку сведения о новых ошибках в реализации и уязвимостях поступают чуть ли не ежедневно. Рассмотрим наиболее известные.

Без сомнения первой широко известной сетевой уязвимостью явилась неустраненная функция `debug` в программе приема/отправления электронной почты `sendmail`, которую использовал знаменитый червь Моррисона. Уязвимость позволяла исполнять код, содержащийся в полученном программой письме. Дата обнаружения уязвимости — ноябрь 1988 года. Одним (но, конечно, не определяющим) из признаков работы программы `sendmail` на хосте можно считать функционирование сервиса протокола SMTP.

А вот, например, бюллетень об уязвимости в одном из самых популярных современных почтовых серверов Microsoft Exchange 5.5. Бюллетень

имеет номер MS02-037 и уведомляет о том, что ошибка в реализации разбора параметров одной из команд протокола SMTP позволяет при некоторых дополнительных условиях выполнить в потоке почтового сервера свой программный код. Это дает возможность получить полное управление над службой почтового сервера и, при неправильно сконфигурированных настройках безопасности Microsoft Windows, над всем компьютером.

Эти два примера приведены, как достаточно разнесенные во времени, но имеющие схожие условия проявления уязвимостей. На наш взгляд, это демонстрирует потенциальную неиссякаемость источников уязвимостей с развитием программного обеспечения, особенно в связи с ориентацией программного обеспечения на предоставление максимального количества услуг и удобства для пользователей.

Сообщения об обнаруженных уязвимостях регулярно публикуются уполномоченными организациями, например CERT — Computer Emergency Response Team (Команда реагирования по вопросам компьютерной безопасности), адрес сайта в Интернете — <http://www.cert.org>, в том числе и с техническими описаниями сущности потенциальной атаки. Ознакомление с такими материалами должно быть задачей одного или даже нескольких технических специалистов из подразделения информационной безопасности.

В отчете по уязвимостям корпорации Cisco приводятся следующие интересные факты, представленные в табл. 11.1 и 11.2.

Таблица 11.1. Наиболее часто встречающиеся сервисы в Интернете

Рейтинг	Сервис	Частота обнаружения %
1	SNMP	12,2
2	Telnet	10,5
3	HTTP	9,8
4	FTP	7,1
5	SMTP	6,6
6	finger	4,8
7	DNS	4,6
8	Rsh	3,9
9	RPC	3,8
10	TFTP	3,4

Авторы отчета обращают внимание на то, что наиболее опасные сервисы (сервис управления сетевыми устройствами SNMP и сервис удаленного терминала Telnet) являются одновременно и самыми распространенными.

Таблица 11.2. Наиболее уязвимые интернет-сервисы

Рейтинг	Сервис	Уязвимые инсталляции, %
1	RPC	93,4
2	SMTP	61,1
3	Finger	59,6
4	TFTP	57,4
5	HTTP	42,4
6	DNS	35,0
7	FTP	33,0
8	NFC	30,2
9	SNMP	27,1
10	X-Windows	23,0

Предполагая SNMP-сервис наиболее распространенным, а RPC — наиболее уязвимым, и принимая во внимание, что оба этих сервиса предназначены для управления, авторы очень рекомендуют ограничить использование любой организацией этих сервисов через Интернет.

До сих пор речь шла об открытых портах как о показателях функционирования штатных сервисов, потенциальные уязвимости которых можно использовать. Однако по ряду открытых портов можно сделать вывод о том, что хост уже взломан, т. е., что на нем была использована имеющаяся уязвимость. Это происходит в тех случаях, когда злоумышленник или автоматическая программа взлома уже установили на хост-жертву троянскую программу. В большинстве случаев взломщики низкой и средней квалификации пользуются широко распространенными троянскими программами и не меняют в их настройках номер порта управления. Например, наличие открытого порта 31337 говорит о том, что на хосте присутствует известная троянская программа Back Orifice — инструмент удаленного администрирования хоста.

В этом случае при соответствующей квалификации любой злоумышленник, обнаруживший установленную троянскую программу, а не только тот, кто реально взломал систему, может получить практически полный доступ к данному хосту со всеми вытекающими последствиями. Список портов

управления, занимаемых по умолчанию наиболее распространенными в последние годы троянскими программами, приведен в *приложении 2*.

Другой вариант поиска путей реализации атак — получение данных об операционной системе, установленной на удаленном хосте. Поскольку сетевая операционная система — это набор различных служб, логично предположить, что каждая операционная система, соответственно, имеет набор уязвимостей. Таким образом, достоверное определение операционной системы на хосте потенциальной жертвы позволяет злоумышленнику начать поиск уязвимостей из предопределенного списка, характерного для данной операционной системы. Одним из методов определения операционной системы удаленного хоста является анализ *реакции IP-стека системы* (набора сервисов обработки сетевых пакетов по уровням) на специальном образом сформированные некорректные пакеты (англ. *system IP-footprint* или *system IP-fingerprint*). Различные операционные системы будут по-разному реагировать на получение таких пакетов, в том числе отвечать на них в уникальном для данного семейства стиле. Наиболее известным примером удаленного анализатора операционной системы, работающим на основе отправки некорректных сетевых пакетов, является программа nmap.

Наиболее распространенные классы удаленных атак

Вирусы и троянские программы



Как уже указывалось в *главе 2*, вирусы и троянские программы (трояны) функционально похожи, и многие из зловредных программ в настоящее время несут в себе черты и тех и других, поэтому как пример удаленной атаки рассмотрим их совместно. Теоретически и вирус и троянская программа могут быть примером и локальной атаки (когда злоумышленник сам находится за консолью управления атакуемой системы), но этот случай не так интересен, как удаленная атака "в чистом виде".

Для того чтобы начать свои деструктивные действия, вирус/троянская программа должны быть доставлены на атакуемый хост, и далее, в зависимости от реализации кода и используемой уязвимости, возможны два варианта развития событий.

- ❑ Внедрение вируса/троянской программы произойдет только после некоторых действий пользователя.
- ❑ Внедрение произойдет автоматически, используя уязвимость одного из работающих сервисов.

На самом деле оба случая довольно трудно различимы, так как даже во втором случае автоматическое срабатывание, возможно, будет вызвано некими

действиями (либо, наоборот, бездействием) пользователя за некоторое время до поступления деструктивного субъекта на хост, т. е. например, пользователь неверно сконфигурировал настройки определенной службы либо вовремя не произвел обновление уязвимого модуля, что и послужило причиной успеха атаки. В чистом виде успех атаки совершенно без участия пользователя может быть только в случае серьезной уязвимости в службе, допущенной скорее всего разработчиком при проектировании или реализации службы. Такие атаки встречаются реально не очень часто, но тем не менее возможны.

Большую же часть успешных попаданий вирусов/троянских программ на хост обуславливают соответствующие действия пользователя, поэтому интересны именно такие случаи.

Пользователь может выступать либо активной, либо пассивной стороной. Яркий пример активных действий — ситуация, когда пользователь сам загружает на свой хост программное обеспечение или другой объект, который может скрывать деструктивный код — с дискеты, компакт-диска, по сети. Пассивный вариант действий атакуемого пользователя — он получает предложение от некоего субъекта к загрузке кода (письмо, переданная дискета, диск и т. п.). В этом случае успех или неуспех реализации атаки будет зависеть от того, насколько легкомысленно пользователь отнесется к такой загрузке.

Теперь попробуем еще раз перечислить возможные причины потенциального успеха атаки вируса/троянской программы и дать общие рекомендации (рис. 11.1).

1. Ошибки в работе сервиса. Наиболее сложный вид атак, так как может быть произведен без участия самого пользователя. Решение — быть постоянно осведомленным о новостях по безопасности производителя сервиса, о новостях с основных форумов и он-лайн служб безопасности.
2. Несвоевременное обновление ПО. Различают два вида таких ошибок.
 - Несвоевременное обновление соответствующих модулей самого сервиса. Решение — подписка на регулярную рассылку информации об имеющихся обновлениях сервиса.
 - Несвоевременное обновление модулей системы защиты сервисов (например, антивирусного программного обеспечения). Решение — приобретать лицензионные системы защиты с соответствующей поддержкой производителем обновления систем.
3. Ошибки в конфигурации. Подобные ошибки довольно трудно предупредить, так как они требуют серьезной квалификации администратора сервиса. В качестве общей рекомендации можно высказать следующее: не давать сервису в функционировании больше прав и возможностей, чем это нужно для выполнения работы. Скажем, если FTP-сервер пред-

назначен только для публикации материалов, нет необходимости оставлять поддержку команды PUT.

4. Загрузка пользователем постороннего программного обеспечения или других контейнеров зловредных программ (документов и таблиц с макросами, скриптов, компонент и т. п.). Наилучшей рекомендацией в данном случае, конечно, был бы полный запрет на любую загрузку перечисленных объектов, однако по производственной задаче такие загрузки могут быть необходимыми. В этом случае возможны следующие варианты.
 - Служба безопасности должна иметь отдельный "полигон" (компьютер или набор компьютеров) для проверки посторонних информационных объектов. На этот "полигон" пользователи соответственно направляют подозрительные или незнакомые информационные объекты.
 - Если создание подобного "полигона" невозможно, то его аналог придется создать на своем хосте. Последовательность действий в этом случае такова. Следует обеспечить ограничение распространения возможного ущерба — отключить компьютер от сети. Затем создать учетную запись тестового пользователя с минимальными привилегиями, необходимыми для исследования информационного объекта. При этом особое внимание обратить на то, чтобы права этой учетной записи на доступ к критическим разделам (системные каталоги и файлы, другие важные объекты) были минимальными — скажем, только на чтение, а полный контроль или даже право на запись на жесткий диск имелось бы только для ограниченного (проще всего, одного) каталога, в котором и проводить исследование. Далее, необходимо зарегистрироваться на хосте от имени этого тестового пользователя и получить доступ к исследуемому объекту (переписать его на жесткий диск, вставить дискету и т. д.), произвести его проверку имеющимися средствами защиты — скажем, антивирусным программным обеспечением. Перед запуском исполняемой части объекта на выполнение рекомендуется посчитать и записать все процессы, выполняемые в текущий момент на хосте, с тем чтобы можно было обнаружить новые несанкционированные процессы. Кроме того, если есть возможность, запустить службу аудита — регистрации попыток записи и другой работы с критическими разделами, либо сосчитать и сохранить на независимом устройстве контрольные суммы системных и наиболее важных пользовательских файлов. После этого можно произвести запуск исполняемой части и протестировать ее функциональность, получить и скопировать необходимую информацию и т. д. Затем остановить выполнение модуля и проанализировать имеющиеся результаты (новые процессы, новые и модифицированные файлы на диске, журнал аудита). По указанным источникам можно сделать определенные выводы об опасности или безопасности исследуемого объекта.

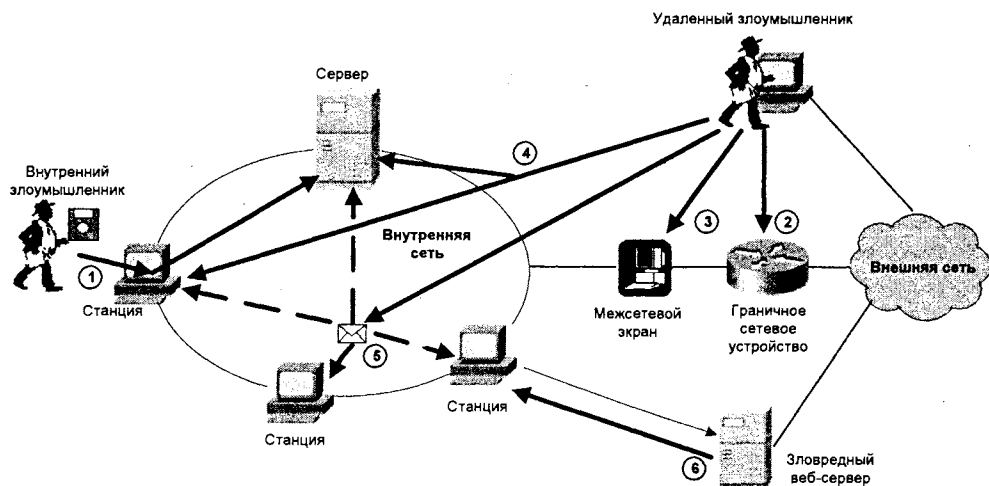


Рис. 11.1. Возможные варианты проникновения в систему (атаки):

- 1 — с консоли (например, используя дискету); 2 — на маршрутизацию;
- 3 — межсетевой экран; 4 — на операционную систему или приложение (сервера или станции); 5 — рассылкой вируса или троянской программы;
- 6 — загрузкой зловредного кода с удаленного сервера

Чтобы отвлечься от теоретических рассуждений, приведем примеры некоторых нашумевших вирусов 2001 года по бюллетеням CERT.

Вирус *Курниковой* (*Anna Kournikova VBS malicious code*) — февраль 2001 года. В качестве основных жертв названы пользователи Microsoft Outlook, которые не произвели соответствующих обновлений по безопасности для своего приложения.

Пользователи получали письмо следующего формата:

SUBJECT: "Here you have, ;o)"

BODY:

Hi:

Check This!

ATTACHMENT: "AnnaKournikova.jpg.vbs"

Самое неприятное заключалось в том, что письмо приходило от адресата, с которым пользователь обычно состоял в переписке и которого более или менее знал. Кроме того, если в настройках системы был установлен флажок **Скрыть расширения файлов**, то пользователь мог не увидеть, что вложенный в письмо файл — это скрипт на языке Visual Basic. Если пользователь пытался открыть присоединенный файл, то происходило следующее.

Код создавал в реестре следующий ключ:

HKEY_CURRENT_USER\Software\OnTheFly="Worm made with Vbswg 1.50b"

Далее копировал себя в C:\WINDOWS\AnnaKournikova.jpg.vbs

После этого он пытался послать отдельное письмо с тем же скриптом по всем адресам, обнаруженным в адресной книге. Затем, чтобы предотвратить повторную рассылку, вирус создает ключ:

```
HKKEY_CURRENT_USER\Software\OnTheFly\mailed=1
```

Широкое распространение вируса может привести к сбоям в работе почтовых серверов.

Вирусу *Красный Код* (*Code Red worm*) посвящено несколько сообщений CERT, так как вирус модифицировался злоумышленниками. Первые сообщения датируются июлем 2001 года. Основное воздействие вирус оказал на веб-серверы IIS 4.0 и IIS 5.0, используя атаку на переполнение буфера в динамической библиотеке (DLL) службы индексирования (IIS Indexing Service). Деструктивное действие развивалось следующим образом.

Производилось сканирование хоста-жертвы на наличие открытого порта 80 (протокол HTTP). При обнаружении высылались соответствующим образом сформированная строка (HTTP GET-запрос), которая и приводила к переполнению буфера в обрабатывающем ее сервисе, в результате чего атакуемый хост исполнял деструктивный код.

Код проверял на хосте наличие каталога c:\notworm, при обнаружении прекращал выполнение, в противном случае начинал порождать процессы по сканированию случайного набора IP-адресов других жертв. Если на атакованном хосте по умолчанию был установлен английский язык, то после запуска 100 процессов сканирования и установленного промежутка времени, все страницы на сервере заменялись на сообщение:

```
HELLO! Welcome to http://www.worm.com! Hacked By Chinese!
```

(Привет! Добро пожаловать на http://www.worm.com!. Взломано Китайцем).

Если же английский не являлся языком по умолчанию, то вирус просто продолжал сканирование.

Кроме замены страниц и некоторого падения производительности, мог быть получен и следующий ущерб. Так как каждое новое порождение вируса использовало тот же самый генератор случайных чисел для формирования списка IP-адресов, могло оказаться, что конкретный адрес попадал в большое количество списков и массовое сканирование приводило к отказу в обслуживании (описание таких атак см. разд. "Отказ в обслуживании" данной главы).

Вторая версия вируса *Code Red II* появилась в августе 2001 года и требовала на хосте-жертве, кроме IIS, еще и установленный Indexing Server 2.0. Вирус использовал тот же механизм переполнения буфера, однако дальнейшее развитие атаки происходило по-другому. Вирус сканировал тот же 80 порт, предполагая наличие веб-сервера. Если устанавливалось успешное соединение, тогда посылались специально сформированный HTTP GET-запрос для переполнения буфера в Службе индексирования (Indexing Server). При ус-

тип "audio/x-wav", но содержит присоединенный бинарный исполняемый файл `readme.exe`. Некоторые почтовые приложения автоматически запускают на выполнение присоединенный файл с таким MIME-типом при открытии или предварительном просмотре письма. Также вирус может быть инициирован неосторожным пользователем, запустившим присоединенный файл. Обычно длина этого файла была 57 344 байта.

Адреса, по которым рассылался вирус, собирались автоматически из следующих источников: из файлов `htm` и `html` в каталоге для веб-кэша пользователя, из содержимого электронных писем пользователя через службу `MAPI`.

Кроме того, вирус сканировал IP-адреса хостов в поиске уязвимых серверов `IIS` — лазеек, оставленных другими вирусами и не закрытых соответствующими обновлениями уязвимостей. Адреса для сканирования выбирались следующим образом по отношению ко времени сканирования:

- ☐ 50% времени сканировались адреса с теми же двумя первыми октетами, что и скомпрометированный хост;
- ☐ 25% времени сканировались адреса с тем же первым октетом;
- ☐ 25% времени сканировались случайно выбранные адреса.

Зараженный хост пытался отправить копию кода по протоколу `TFTP` (69 порт протокола `UDP`) на обнаруженный сервер `IIS`. Запущенный на сервере вирус в каждой доступной директории размещал свою копию с расширением `eml` или `nws`, а к `html` или `asp` файлам добавлял `JavaScript` код.

Дополнительно он открывал доступ по сети к диску `C` (разделенный ресурс — `sharing` — для "`C:\`"), создавал учетную запись `GUEST` и добавлял ее в группу администраторов. Кроме того, он изменял копии исполняемых бинарных файлов, добавляя в них троянскую программу, вызывающую код вируса на исполнение.

Сканирующую активность вируса возможно было определить по наличию в регистрационном журнале веб-сервера для порта 80 `TCP` следующих записей:

```
GET /scripts/root.exe?/c+dir
GET /MSADC/root.exe?/c+dir
GET /c/winnt/system32/cmd.exe?/c+dir
GET /d/winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET /msadc/..%5c../..%5c../..%5c/..\xc1\xlc../..\xc1\xlc../..\xc1\xlc../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\xlc../winnt/system32/cmd.exe?/c+dir
```

```
GET /scripts/..\xc0../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0\xaf../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\x9c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\%5c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\%2f../winnt/system32/cmd.exe?/c+dir
```

Вirus *Покойник (W32\Goner worm)*, декабрь 2001 года. Уязвимые системы — Windows системы с установленными приложениями MS Outlook, MS Office, ICQ. Распространяется по электронной почте или при передаче ICQ файла. Представляется в виде программы-заставки (screen saver) — gone.scr, инициализируется при запуске на исполнение. Электронное письмо имеет следующий вид:

Subject: Hi!

Body: How are you ?

When I saw this screen saver, I immediately thought about you

I am in a hurry, I promise you will love it!

Attachment: gone.scr

Когда присоединенный файл запускается, он имитирует работу программы-заставки с выдачей сообщения об ошибке. Тем временем код копирует себя в системный каталог Windows C:\WINDOWS\SYSTEM32\scr.exe или C:\WINNT\SYSTEM32\scr.exe и изменяет реестр для запуска кода после перезагрузки.

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\
C:%WINDIR%\SYSTEM\gone.scr=C:%WINDIR%\SYSTEM\gone.scr
```

Вirus распространяет себя, высылая электронное письмо по списку адресной книги MS Outlook и направляя файл по списку пользователей ICQ.

Дополнительно вирус ищет и останавливает процессы, связанные с известными антивирусными пакетами и программами по безопасности, и пытается удалить исполняемые файлы этих программ. Если удаление сразу не получается, то вирус создает файл WININIT.INI, который производит удаление при перезагрузке.

Отказ в обслуживании



Атаки на *отказ в обслуживании* (англ. *deny-of-service* — *DoS*) являются одними из самых распространенных. Это связано с тем, что они сводятся к выведению информационного объекта из строя, а не к получению какого-либо вида несанкционированного доступа к нему. Действительно, наиболее примитивный вариант подобной атаки — разрыв сетевого кабеля или физическое разрушение сетевого устройства — не требует от злоумышленника серьезной технической

квалификации. Несмотря на некую крайность, тем не менее этот пример хорошо демонстрирует сущность атак: приведение в неработоспособное состояние устройства приема/передачи либо приведение канала передачи информации в состояние невозможности транспортировки данных, т. е. возможны два основных направления реализации подобных атак.

- ❑ Критическая часть информационного объекта, отвечающая за обеспечение доступа к информации, разрушается или приводится в неработоспособное состояние. Ситуация требует дополнительного вмешательства администратора для восстановления работы.
- ❑ Некий контейнер приема/передачи информации подвергается запредельной нагрузке, превышающей его пропускную способность, и не может функционировать до тех пор, пока нагрузка не вернется к нормальной. Ситуация может не требовать дополнительного вмешательства, если такая нагрузка не приводит к повреждению или разрушению контейнера.

Другая классификация атак на отказ в обслуживании носит более подробный технический характер.

- ❑ *Перегрузка пропускной способности сети.* Атакующий генерирует такое количество сетевого трафика, которое полностью занимает возможности данного сетевого соединения.
- ❑ *Перегрузка системного процессора.* Атакующий создает вычислительные задания для атакуемого узла, которые превосходят возможности процессорной обработки.
- ❑ *Занятие возможных портов.* Атакующий устанавливает соединения с имеющимися портами таким образом, что занимает все имеющиеся открытые порты и допустимое количество соединений на порт.
- ❑ *Перерасход системных ресурсов (памяти, процессорного времени).* Атакующий генерирует либо огромное количество данных, сохраняемых в системной памяти, либо большое количество запросов, на обработку которых уходит практически все процессорное время системы. Этот класс атак опасен тем, что может быть выполнен на каком-нибудь другом (более уязвимом) сервисе этого же компьютера, а приведет в итоге к неработоспособности всех служб.

Соответственно, согласно этой классификации природа атак на отказ в обслуживании разделяется:

- ❑ на формирование некоего специального, возможно, некорректного, запроса информационному объекту, обработка которого приведет к выводу объекта из строя;
- ❑ генерирование ненормально большого количества корректных данных, с обработкой которых не справятся либо канал передачи информации, либо устройство (объект) приема/передачи информации.

Первый вид атак связан с определенной технической и интеллектуальной работой злоумышленника, которая заключается в поиске уязвимого параметра информационного объекта, который может быть подвергнут нападению. Такой вид атак наиболее непредсказуем при реализации в первый раз, однако после того как он становится известен производителю, выпускается соответствующее обновление, которое, будучи установленным на объект, решает проблему с данной атакой (по крайней мере на данном объекте), т. е. после установки такого обновления можно с высокой степенью вероятности утверждать, что данная атака больше не достигнет успешного результата.

Второй вид атак не требует специальной подготовки и может быть осуществлен даже случайно. Например, когда пользователь посылает письмо большого объема (атака типа mailbomb) или большое количество писем (атака типа spam), он может серьезно блокировать работу корпоративного почтового сервиса. Такие атаки при соответствующей квалификации администратора легко обнаруживаются и обычно устраняются путем выдачи устройству приема/передачи инструкций по запрету обработки информации от источника атаки.

Так было до тех пор, пока не появился новый вид атак на отказ в обслуживании — *распределенный отказ в обслуживании* (англ. *distributed deny-of-service* — *DDoS*). О ней стоит рассказать подробнее.

Сама атака состоит из нескольких частей. В подготовительной части атаки хост потенциальной жертвы не принимает участия. Злоумышленник набирает некоторое количество хостов — промежуточных жертв, на которых размещает некоторое программное обеспечение, вообще говоря, троянскую программу. Способ доставки троянской программы здесь не рассматривается, это может быть и вирус, и обычное письмо с вложением: главная задача злоумышленника на данном этапе — это набрать количество, вне зависимости от того, кто конкретно это будет. После размещения достаточно большого количества программ, в определенное время или по дополнительной команде промежуточные жертвы (которые называются в терминах данной атаки *зомби*) начинают посылать запросы собственно на основную цель атаки, выполняя таким образом функции инструмента атаки злоумышленника. Запросы на хост-жертву либо представляют из себя реальные корректные запросы от клиентов хоста (например, от браузеров — веб-серверу), либо очень похожи на них таким образом, что атакуемый хост не может отличить ложные запросы от реальных и вынужден расходовать время на их обработку. То, что запросы приходят не от одного источника, затрудняет идентификацию их, как атаки. Большинство рекомендаций по противодействию такой атаке сводятся к увеличению производительности атакуемого хоста путем приостановки на нем второстепенных сервисов и процессов. Также, если возможно по какому-либо параметру отделить ложные запросы от запросов реальных клиентов, необходимо установить соответствующий фильтр на обработку запросов хостом.

Замечание

Описания указанных атак взяты из бюллетеней CERT, однако исследования проведенные другими экспертами в [Медведовский] показывают, что не все такие атаки несут заявленную опасность. Наша задача рассказать о многообразии и природе порождения атак, даже если они представляют не практическую, а скорее теоретическую опасность. На наш взгляд, специалисту по безопасности опыт подобного изучения будет полезен в любом случае (хотя бы потому, что вопросы по подобным атакам входят в тесты на экзаменах по информационной безопасности).

Приведем примеры некоторых известных атак.

Атака Ping смерти (Ping-of-death), декабрь 1996. Атака производится на сервис стека сетевых протоколов, отвечающего за работу с протоколом ICMP, частью которого является утилита Ping. Превышение стандартной длины сетевого пакета (в данном случае 65 536 байт) может привести к тому, что соответствующий сервис выйдет из строя и операционная система даст сбой (зависание или перезагрузку). Формирование такого пакета может осуществляться, например, стандартной командой `ping` операционной системы с указанием в качестве цели адреса хоста-жертвы и размера посылаемого пакета, превышающего стандартный.

Атака потоком SYN запросов (англ. *SYN flood*), сентябрь 1996. Атака основана на специфике функционирования процесса установления соединения протокола TCP. Для установления соединения с сервером клиент посылает серверу запрос SYN, сервер отвечает подтверждением готовности к установке соединения SYN-ACK и ожидает от клиента согласия на соединение ACK, при получении которого соединение считается установленным и можно передавать данные. Однако, если клиент не ответил пакетом ACK, соединение считается полуоткрытым, и сервер, в зависимости от конфигурации, некоторое время ждет сообщения от клиента. Обычно, если время ожидания указано достаточно большим (а ведь количество соединений, которое может быть установлено с сервером одновременно, — это конечное число), то в такой ситуации может быть достигнуто состояние, когда сервер не может больше принимать запросы на соединение и оказывается фактически недоступным. По истечении установленного времени ожидания, полуоткрытое соединение ликвидируется, однако, если злоумышленник посылает SYN-запросы быстрее, чем истекают тайм-ауты фальшивых соединений, то сервер останется недоступным до тех пор, пока SYN-поток не прекратится. Для некоторых систем переполнение списка текущих соединений может привести к зависанию операционной системы.

Атака обычно сопровождается также и подменой IP-адреса отправителя SYN-запроса — каждый раз в пакет помещается в новое произвольное значение адреса, с тем чтобы на сервере нельзя было предотвратить атаку установкой ограничений на количество одновременных соединений с одного IP-адреса.

Атака SYN-flood опасна по следующим причинам:

- потенциально она непреодолима, так как неразрывно связана с идеологией протокола TCP (полумерами могут быть только повышение производительности серверов, увеличение их служебных буферов, и, как следствие, количества поддерживаемых одновременно соединений, а также сокращение времени ожидания ответа клиента до предельно возможной разумной величины — например, 0,2—3 с — в зависимости от характера сетевых маршрутов к клиентам);
- у довольно большой части серверов для создания подобного потока достаточно модемного соединения (30—40 Кбит/с) от злоумышленника, несмотря на то, что сам сервер может быть подключен к Интернету качественным каналом пропускной способностью в несколько Мбит/с.

Атака Smurf. Для полного понимания технического механизма атаки необходимо знание принципов функционирования протокола ICMP (Internet Control Message Protocol — Протокол Управляющих Сообщений Интернета). Не углубляясь в технические подробности, укажем, что одной из функций протокола является выяснение, доступен ли для работы удаленный хост. Выяснение производится выполнением команды операционной системы `ping`, которая формирует пакет запроса (ICMP echo) с указанием адреса удаленного хоста и обратного адреса, куда присылать ответ. Если удаленный хост получают такой запрос и в состоянии его обработать, он посылает в ответ на обратный адрес пакет подтверждения (ICMP echo reply). Протокол допускает широковещательную рассылку запроса, когда сразу все хосты подсети получают запрос на доступность. Этой возможностью и воспользовались злоумышленники. Атака состояла из трех частей. В первой части (рис. 11.2) атакующий формирует широковещательные запросы на несколько подсетей, однако при этом вместо обратного адреса указывает не свой сетевой адрес, а адрес хоста-жертвы. На втором этапе шлюз промежуточной подсети получает запрос хостам подсети и, если на шлюзе не установлена фильтрация широковещательного ICMP-трафика, то шлюз направляет запросы хостам подсети. Уже в этот момент данная промежуточная подсеть может испытать проблемы с пропускной способностью, когда все работающие хосты подсети направят ответ на запрос по обратному адресу, принадлежащему жертве.

На третьем этапе (рис. 11.3) хост-жертва получит множественные ответы на запросы, которые он не посылал. При достаточном количестве запрошенных подсетей с большим количеством хостов в них, подобный поток ответов приведет к тому, что хост-жертва будет выведен из рабочего состояния.

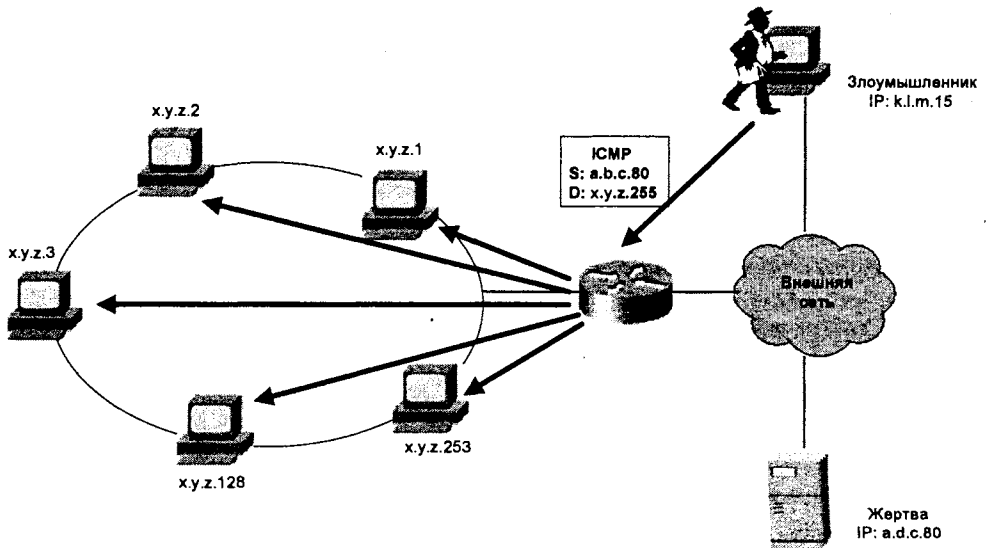


Рис. 11.2. Атака Smurf: этапы 1 и 2

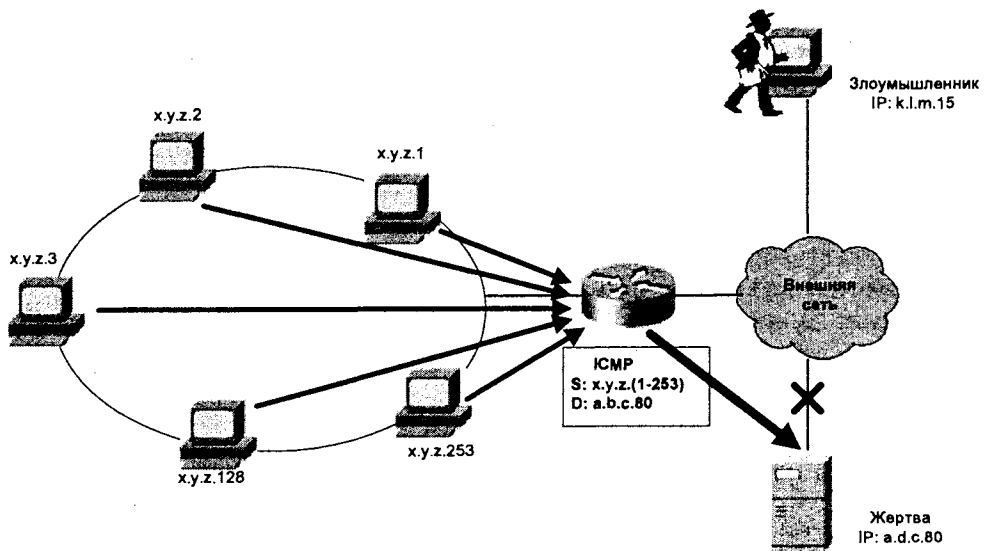


Рис. 11.3. Атака Smurf: этап 3

Разновидностью подобной атаки можно назвать атаку Fraggle, основанную на специфике работы протокола UDP. Стандартный, но редко используемый отладочный сервис chargen (character generation — создание потока символов) функционирует следующим образом — при установке соедине-

ния с хостом по UDP-порту 19, хост начинает генерировать непрерывный поток данных. Указав в качестве обратного адреса сетевой адрес жертвы, возможно вызвать перебои в работе сети.

Маскировка



Маскировка (англ. *masquerading*) — общее название для очень большого класса атак, которые заключаются в том, что атакующий выдает себя не за того, кем является на самом деле. Атака, например, использовалась на ранних стадиях развития межсетевых экранов, а в настоящее время может быть использована на маршрутизаторах там, где идентификация и аутентификация производилась только на основе сетевого адреса. Если определенные существенные права присваиваются процессам, инициированным рядом доверенных хостов (то есть пакеты, с адресом источника, входящим в список доверенных, пропускаются без применения к ним ограничивающих правил, и инструкции в этих пакетах принимаются на получающем хосте к исполнению), то злоумышленнику достаточно указать доверенный адрес отправителя в сформированном деструктивном пакете, и он будет пропущен без ограничения.

Маскировкой можно назвать и случай, когда злоумышленник присваивает себе аутентификационные параметры (идентификатор и пароль) легального пользователя и, таким образом, выдает себя за него. Более сложный вариант — это, например, уязвимость в сервере ключей протокола Kerberos версии 4, которая заключалась в слабом генераторе случайных чисел, что давало злоумышленнику возможность взломать сессионный ключ и благодаря этому стать авторизованным пользователем.

Все рассмотренные случаи атак исходят из того, что, маскируясь, злоумышленник является активным субъектом. Но маскировка может происходить и в пассивном варианте. Например, если злоумышленником был получен доступ на изменение записей на соответствующем DNS-сервере, то он сможет выдать свой хост за HTTP, FTP или другой сервер и ввести в заблуждение пользователей, которые станут к нему обращаться. В этом случае атакующий может разместить на ложном сервере выгодную для себя информацию либо, имитируя приглашение к входу, собирать идентификаторы и пароли пользователей для реальных серверов.

Маскировка под другим названием — *подмена* (англ. *spoofing*) — используется тогда, когда атакуемый хост использует для сетевого взаимодействия с другим хостом некоего посредника в качестве поставщика определенного рода справочной информации. Наиболее известный вариант — использование сервера разрешения имен DNS. В определенных случаях злоумышленник, выдавая себя за такой сервер, может произвести успешную атаку вне зависимости от того, насколько хорошо защищены общающиеся между собой хосты или сам сервер. Кратко опишем эту атаку.

Как известно, основная задача сервера DNS — предоставить запрашивающему хосту (хосту-жертве) в ответ на имя хоста-назначения сетевой адрес хоста-назначения. Если злоумышленнику удастся вместо ответа легитимного DNS-сервера направить хосту-жертве свой ответ, то хост-жертва будет направлять свои пакеты не на адрес хоста-назначения, а на тот адрес, который укажет злоумышленник в своем ложном ответе.

Аналогичного эффекта можно добиться, атакуя и другие протоколы и базы данных, ответственные за сопоставление адресов или имен одного уровня адресам другого уровня — ARP (преобразует IP-адреса в MAC-адреса канального уровня), WINS (преобразует Windows-имена хостов в IP-адреса) и т. д.

Атаки на маршрутизацию



Возможно, читатель обратил внимание, что иногда в сфере интересов злоумышленника находится не только собственно хост жертвы, но и некоторые промежуточные узлы. В связи с этим рассмотрим еще один класс атак — атаки на маршрутизацию пакетов. Довольно часто для достижения атакующим пакетом хоста-жертвы может существовать альтернативный путь, либо путь, игнорирующий и изменяющий правила доступа к хосту-жертве. Кроме того, сам хост-жертва может по-разному реагировать на пакеты, поступившие различными путями. В любом из этих случаев в область интересов злоумышленника попадают промежуточные устройства доставки пакетов — маршрутизаторы. Учитывая то, что некоторые организации вместо полноценных межсетевых экранов используют маршрутизаторы с фильтрующими функциями либо без таковых, интерес злоумышленников будет вполне обоснованным.

Наиболее удобным с точки зрения атакующего вариантом будет получение доступа на редактирование таблиц маршрутизации — в этом случае он получает полный контроль над способами направления пакетов данным устройством. Однако получение такого доступа — задача нетривиальная, так как маршрутизаторы обычно находятся под пристальным вниманием администраторов и соответствующим образом защищены.

Одним из вариантов атаки на маршрутизацию без получения доступа к таблицам маршрутизации является возможность, заложенная в IP пакетах и называемая *маршрутизацией от источника* (англ. *source routing*). Данная опция позволяет отправителю управлять движением пакета, предоставляя промежуточным шлюзам маршрутизирующую информацию и игнорируя тем самым настройки, определенные администратором сети. При этом возможны два варианта: *гибкий* (англ. *loose source route*), когда между двумя соседними адресами в пути маршрутизации может стоять несколько других шлюзов, и *жесткий* (англ. *strict source route*), когда два соседних адреса должны находиться в сетях, соединенных напрямую. Сама опция состоит из

добавления к стандартному IP-заголовку набора из нескольких IP-адресов и указателя смещения, определяющего следующий адрес, который должен быть обработан (рис. 11.4). При этом адрес назначения считается достигнутым, если указатель смещения больше длины поля адресов.

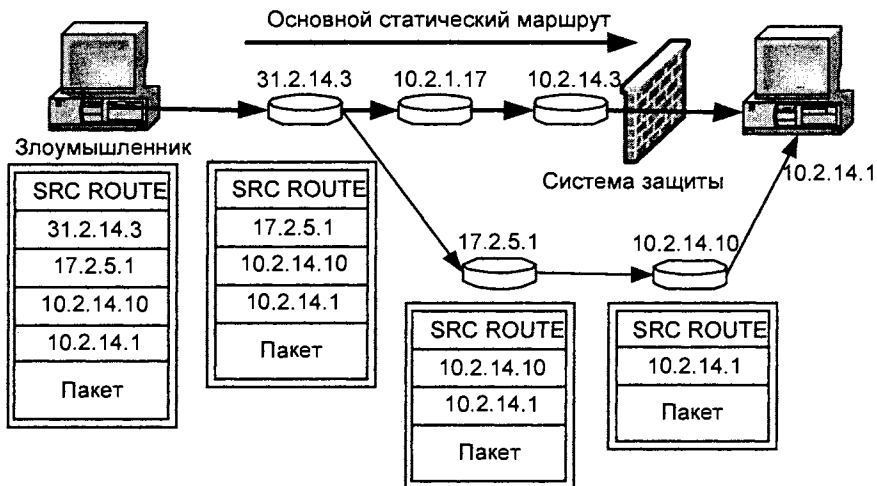


Рис. 11.4. Маршрутизация от источника

Это свойство дает возможность реализации, например, такой атаки, которая реализуема в некоторых сервисах TCP/IP. Злоумышленник формирует пакет, в котором смещение уже больше длины. Хост назначения, получив пакет, считает маршрут исполненным и формирует пакет ответа, выстраивая обратную цепочку промежуточных узлов и направляя пакет первому хосту в этой цепочке. Поскольку эта цепочка была сформирована злоумышленником, то первым хостом в ней может стоять хост из внутренней сети, на границе которой стоит хост назначения, то есть внутренний хост, достижимый только через внутренний сетевой интерфейс хоста назначения.

Кроме того, атаки на маршрутизацию используются для прослушивания и модификации трафика между хостами в сети. Этому будет посвящен разд. "Атаки на основе сетевой маршрутизации" главы 12.

Атаки на серверы: CGI и HTTP



Не вдаваясь в подробности функционирования механизма CGI (*Common Gateway Interface — Интерфейс Единого Шлюза*), скажем, что это набор программ (исполняемых скриптов, интерпретаторов и т. п.), выполняемый на сервере по запросу клиента. Он предназначен обеспечивать динамическое (то есть более гибкое) взаимодействие между удаленным клиентом-

браузером и Веб-сервером посредством запуска на серверной стороне приложений, выполняющих задания клиента.

Как и любым другим программам, данному механизму присущ ряд возможных уязвимостей, которые усиливаются тем, что он обслуживает клиентов (пользователей), о которых предварительно ничего не известно, т. е. возможны и потенциальные злоумышленники.

Основные проблемы, связанные с работой CGI, сводятся к тому, что:

- ❑ разработчиком допущены ошибки при написании кода;
- ❑ в коде оставлены преднамеренные люки;
- ❑ злоумышленник получил возможность (права) модифицировать код либо данные, используемые кодом на сервере;
- ❑ злоумышленник получил доступ к интерпретатору с правом исполнения команд.

Наиболее часто встречаются уязвимости, относящиеся к ошибкам на этапе разработки, причем в значительной степени это касается необходимости проверок данных, которые поступают от клиента на обработку программой. Этому вопросу был посвящен отдельный бюллетень CERT — CERT Advisory CA-1997-25 Sanitizing User-Supplied Data in CGI Scripts. В качестве примера возможности реализации атаки приведем CERT Advisory CA-1997-24 Buffer Overrun Vulnerability in Count.cgi cgi-bin Program. В документе говорится о программе `Count.cgi`, которая ведет подсчет посещений Веб-сервера. По причине недостаточной проверки данных, вводимых клиентом, возможно произвести переполнение стека. В результате `Count.cgi` исполнит команды злоумышленника, наделив их при этом правами доступа, равными правам доступа сервиса `httpd` в данной системе.

В целом, специальным образом подобранные некорректные (в смысле скорее "неожидаемые разработчиком") наборы данных представляют собой угрозу, характерную не только для CGI. Известны случаи, когда использование специальных символов ("/", "." и ряда других) в URL при обращении к Веб-серверу приводило к совсем иным последствиям, чем это предполагалось разработчиками и администраторами Веб-сервера — от отказа в обслуживании до возможности исполнения зловредного кода с высокими привилегиями. Не будем приводить весь список подобных атак — он довольно однообразен (сформированный особым образом URL — в ответ недокументированная реакция сервера). Остановим внимание специалистов на том вопросе, что, обеспечивая защиту своих Веб-серверов, например, с помощью межсетевых экранов, необходимо уделять внимание не только правильной адресации и обращению к службам, но и тому, какие данные получает конкретное приложение. Другими словами, строя защиту, необходимо в анализе угроз подниматься от сетевого и транспортного уровня сетевой модели — вверх до прикладного.

Атаки на клиентов: ActiveX, Java и другие



При взаимодействии клиента и Веб-сервера возможен запуск кода и на клиентской стороне; наиболее популярные технологии на сегодняшний день, это *Java Virtual Machine* и *ActiveX*. Необходимо сразу осознать, что загружаемый с удаленного сервера код — это набор команд, которые ваш хост (компьютер) будет исполнять, возможно, не информируя вас о результатах. Поэтому и относиться к такому коду следует соответственно. Возможно, мы загружаем этот код с доверенного сервера, но что мы можем, знать о его безопасности? Достаточно ли уверенности, что механизм электронно-цифровой подписи сообщил нам, что код не был модифицирован после того, как его разместил на сервере уполномоченный программист? Возможно, мы считаем, что настройка прав такова, что данный код не сможет получить большие привилегии?

Прежде всего необходимо определиться, что мы хотим от данного кода? Большая часть такого кода в Интернете предназначена для визуализации или озвучивания красивых, в значительной степени рекламных, эффектов при посещении данного сайта. Поэтому необходимо разделять следующее.

- ❑ С какого компьютера (хоста) мы сейчас работаем? Это компьютер в Интернет-кафе, обычная пользовательская рабочая станция или компьютер, выполняющий важную роль в сети либо хранящий конфиденциальную информацию?
- ❑ Что мы делаем? Либо это просто просмотр новых сайтов для удовольствия, либо поиск необходимой информации в Интернете, либо работа с функциональностью удаленного сервера?

Соответственно этому будут актуальны и разные варианты ответов на вопрос о том, что можно позволить загружаемой программе на нашем компьютере. При развлечении на временном, общедоступном компьютере мы можем загружать все что угодно, какое нам дело до последствий? Если мы ищем справочную информацию, то необходимы ли нам сложные, загружающие память и канал эффекты? Ведь мы ищем чаще всего текст. Графические файлы, исполняемые модули и даже видео-аудио клипы можно скачать отдельно и просмотреть позже, возможно, на другом компьютере соответственно поддержку исполнения загружаемого кода можно отключить.

Другой вариант — работа значимого компьютера с сервером, обеспечивающим доставку некоторой функциональности для выполнения задач. Наиболее очевидный пример — обслуживание организацией своих банковских операций на сервере банка. С одной стороны, и на хосте организации находятся важные финансовые сведения, но, с другой стороны, задача может функционировать в режиме тонкого клиента, т. е. все функциональные компоненты для обработки финансовой информации загружаются с сервера банка. В этом случае поддержка загружаемого кода должна быть включена.

Но должны быть соблюдены все необходимые меры по обеспечению безопасности:

- идентификация и аутентификация и клиента и сервера;
- создание защищенного канала обмена информацией;
- проверка целостности и авторизованности загружаемого кода;
- обеспечение настройки прав, с которыми будет работать код;
- доступность данных для кода и т. п.

Переполнение буфера



Способ под названием *переполнение буфера* (англ. *buffer overflow*) — это большой отдельный класс атак, основанный на возможности изменить ход исполнения атакуемой программы с помощью специально сформированных некорректных входных данных. Переполнение буфера используется различными зловредными программами и атаками: вирусами, программами повышения полномочий, удаленными атаками на сервера и клиентов. Те специалисты, которые отслеживают появления информационных бюллетеней по безопасности, согласятся, что такие атаки очень актуальны и в настоящее время.

Способ основан на том, что все программы для своей работы используют некоторые данные, получаемые, возможно и не от авторизованных клиентов. Обработкой полученных данных в программе занимается некоторая функция (или метод, если говорить языком объектно-ориентированного программирования). При этом предполагается, что данные, подлежащие обработке, имеют некоторый, заранее предопределенный формат, скажем, введенный пароль состоит не более чем из 15 символов.

Перед обработкой полученные данные (в нашем случае — пароль) будут записаны в некоторую переменную, место памяти которой будет выделено в некотором оперативном пространстве, которое и называется стеком. Не вдаваясь в особенности функционирования стека, скажем, что в нем же хранится и служебная информация, например, адрес возврата исполнения программного кода, куда программа должна будет вернуть управление после завершения работы данной функции (метода) — в нашем случае, после обработки пароля.

А теперь представим, что введенный пароль оказался длиной не 15 символов, а гораздо больше, настолько, что данные, вводимые как пароль, при записи в стек, "затерли" команды возврата управления. В этом случае скорее всего, программа не сможет продолжать исполнение инструкций кода и прекратит работу с сообщением об ошибке.

Если же предположить, что данные, вводимые как пароль, были не бессмысленным набором символов, а тщательно подобранными двоичными байтами, то при возврате управления из функции (метода) вполне можно добиться, чтобы они были интерпретированы программой как корректный

код процессора и выполнены компьютером (рис. 11.5). Тогда получается, что программа анализа пароля (то есть часть программы аутентификации, а следовательно, процесс с достаточно высокими привилегиями), выполнила инструкции, направленные пользователем, который еще не был авторизован (он еще только вводил пароль), возможно, злоумышленником.

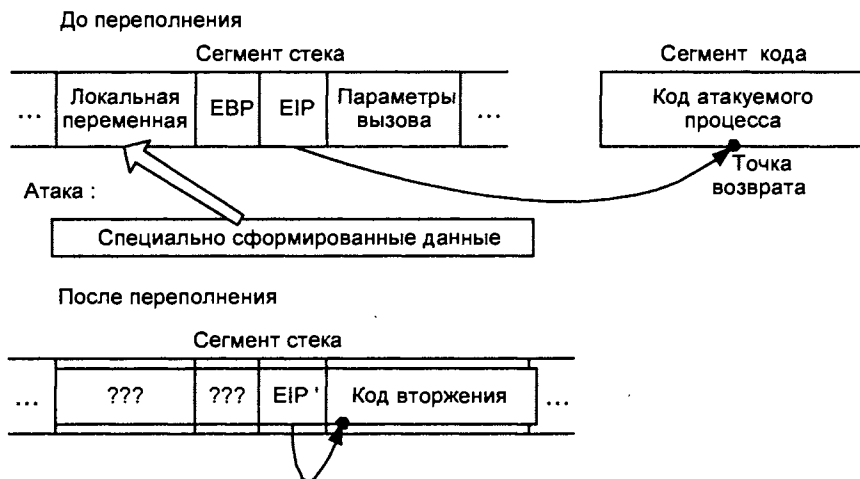
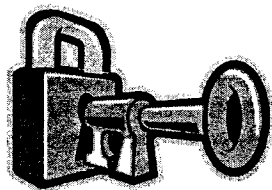


Рис. 11.5. Переполнение буфера

Естественно, данный пример утрирован. Реально программы аутентификации пишутся без возможностей реализации таких атак, хотя то, что о них ничего не известно, еще не означает, что их нет или не будет через некоторое время. Даже в приведенном нами примере опытные программисты сразу укажут ряд возможностей избежать подобных уязвимостей. Это и проверка данных на соответствие предопределенному размеру перед началом обработки, и корректная обработка исключительных ситуаций, когда программа возвращает код ошибки, и специальное размещение данных в стеке, и т.п. Вся беда заключается в том, что в программах на сотни строк кода, да еще создаваемых различными программистами, не всегда уделяется должное внимание таким подозрительным моментам.

В качестве реального примера приведем сообщение CERT Advisory CA-2001-21 Buffer Overflow in telnetd. Telnetd — это программа-сервер, обеспечивающая работу удаленных виртуальных терминалов. В данном случае результат выполнения функции `telrcv` сохраняется в буфере фиксированного размера, при этом предполагается, что размер результата меньше, чем размер буфера. Если же результат окажется по размеру большим, даже еще и сформированным специальным образом, то это может привести к сбою в работе сервера (отказ в обслуживании) или выполнению постороннего кода с привилегиями программы `telnetd` (обычно привилегии `root`).

Глава 12



Атаки на поток данных

Как было указано ранее, *атака на поток данных* — это ситуация, когда злоумышленник не применяет агрессивных воздействий на атакуемый хост или хосты, а, находясь вовне (например, в сети между двумя серверами, ведущими информационный обмен), пытается работать с данными, направляемыми от одного хоста к другому.

По своему характеру такие атаки могут быть *пассивными*, когда злоумышленник копирует себе данные для последующего анализа, и *активными*, когда злоумышленник вносит изменения в данные или полностью подменяет их. Случай, когда злоумышленник блокирует данные, не будет рассматриваться, так как в данной главе это просто означает разрыв связующего кабеля либо заполнение его вредоносным трафиком, а эти атаки не интересны своей примитивностью (хотя при этом они не перестают быть действенными).

Пассивные атаки



Прослушивание сетей

Другое название пассивных атак — *прослушивание сети*, или *сниффинг* (англ. *sniffing*). Различают два вида прослушивания — межсегментный (когда хосты, обменивающиеся данными, находятся в различных сегментах сети) и внутрисегментный (когда информационный поток идет между двумя хостами внутри одного сегмента).

В первом случае злоумышленник должен разместить устройство прослушивания в таком месте, где он сможет гарантированно иметь возможность копирования — у выхода из первого сегмента (у шлюза), у входа во второй сегмент, либо у одного из передающих промежуточных устройств (при условии, что сетевой трафик не имеет альтернативных каналов доставки). Поскольку в этом случае трафик не может миновать злоумышленника, его задача облегчается — он может анализировать на выбор любой из уровней сетевой модели (обычно от сетевого и выше).

Немного остановимся на рассуждениях о том, какой уровень сетевой модели наиболее интересен для злоумышленника. Больше всего значимой ин-

формации расположено на верхнем, прикладном уровне. Скажем, если злоумышленник имеет возможность перехватывать пакеты незащищенных Telnet-сессий, то скорее всего, рано или поздно, он получит аутентификационные данные (идентификатор и пароль) пользователя, работающего с Telnet-сервером. Анализ транспортных пакетов предоставит возможность узнать, какие службы функционируют на взаимодействующих хостах. Рассмотрение сетевого уровня может дать картину адресного пространства сегментов.

Поскольку основным средством защиты от прослушивания является шифрование, необходимо определиться, на каком уровне производить применение криптографических механизмов. Скажем, если исходящие из сегментного шлюза пакеты направлены только Веб-серверу (возможно еще DNS и почтовому серверу), а сам шлюз выполняет функцию сокрытия адресов, то применять шифрацию можно только на прикладном уровне. А если идет взаимодействие между управляющим и управляемым хостами (например, по протоколу SNMP), то защиту необходимо применять на сетевом уровне.

Во втором случае злоумышленнику не обязательно находиться на одном кабеле с хостом, он может использовать свойства работы протоколов канального уровня (например, Ethernet). В сетях подобной топологии сразу несколько (обычно около 10—20) сетевых карт подключено к одному и тому же носителю информации — сетевому кабелю. Каждый пакет, переданный в сети, поступает таким образом на вход каждой сетевой карты. Только после сравнения физического адреса карты получателя (так называемый *MAC-адрес*, от англ. *Media Access Controller* — контроллер доступа к среде передачи), указанного в заголовке пакета, с собственным MAC-адресом сетевая карта принимает решение — ей ли предназначался этот пакет. Один из служебных регистров драйвера сетевой карты определяет, должна ли она передавать на более высокий уровень только пакеты, предназначенные ей (штатный режим), либо же все пакеты, прошедшие по сети (англ. *promiscuous mode* — дословно "неразборчивый режим"). Программы-снифферы переводят сетевую карту во второй режим и, таким образом, получают все пакеты, проходящие по подключенному сетевому кабелю.

Казалось бы, в этом случае невозможно определить — установлен ли сниффер в данном сегменте сети или нет, т. е. ведется ли пассивная атака на хосты сегмента. Однако специалистами по безопасности было найдено довольно красивое решение — сниффер-детектор (англ. *sniffer-detector*), о котором следует рассказать, как о примере направления развития мыслей специалистов по безопасности для обеспечения необходимой защиты сетевых ресурсов. Для того чтобы прийти к такому решению, необходимо соответствующее знание принципов работы сети, программы-сниффера и техническая сообразительность.

Как уже упоминалось, компьютер со сниффером отличается от других тем, что принимает к обработке все канальные кадры вне зависимости от ука-

занного MAC-адреса. Идея заключается в том, чтобы сформировать набор кадров с некорректным MAC-адресом, который должен быть проигнорирован обычным сетевым узлом. Внутри кадры должны содержать сетевые пакеты, на которые обязаны будут отреагировать служебные сервисы сетевого уровня (например, это может быть `ping`-пакет протокола ICMP). Таким образом, получение от какого-либо хоста ответа на подобный сетевой пакет, означает, что он (кадры, его составляющие) был принят к обработке данным узлом и передан на сетевой уровень, который и ответил на служебный запрос. Следовательно, сетевая карта переведена в режим `promiscuous mode` и на данном хосте установлена программа-сниффер.

Другой, не столь красивый вариант решения, заключается в том, что сегмент должен быть заполнен большим количеством некорректных сетевых кадров, которые будут проигнорированы всеми узлами, кроме прослушивающего. Тогда наличие сниффера можно обнаружить по существенному замедлению работы данного хоста (например, в замедлении реакции на ту же команду `ping`).

Прослушивание в коммутируемых сетях Ethernet

Вследствие очень широкого распространения топологии сетей Ethernet, построенных на коммутаторах, необходимо немного рассказать и о прослушивании трафика в таких ситуациях. Технология построения сетей множественного доступа на активном оборудовании — *концентраторах* (англ. *hub*) и *коммутаторах* (англ. *switch*) — позволяет подключать каждую сетевую карту не к единому сетевому кабелю, а к этому концентраторному оборудованию своим собственным кабелем. Оба типа устройств "прозрачны" на канальном уровне, то есть прохождение через них кадра канального уровня никак не сказывается на его содержимом и не может быть обнаружено программами-снифферами.

Повторители выполняют единственную функцию копирования кадра, пришедшего по одному из подключенных к нему кабелей, во все остальные сетевые кабели. А вот коммутаторы выполняют роль "интеллектуального" перенаправления этого кадра. На каждом сетевом порту поддерживается список MAC-адресов, от которых в последнее время приходили пакеты с этого сетевого кабеля. Чаще всего на другой стороне кабеля находится один единственный компьютер, в этом случае список для данного порта будет состоять из одного MAC-адреса. Но если к коммутатору подключается другой коммутатор или повторитель, количество адресов в списке возрастает.

Перенаправление кадра, пришедшего на коммутатор, производится по следующему алгоритму: если MAC-адрес получателя обнаружен в списке для какого-нибудь из портов коммутатора, пакет отправляется только по этому направлению, если же адрес не найден, то пакет копируется во все подключенные сетевые кабели, как в повторителе. Коммутаторы в несколько раз снижают нагрузку на сетевой носитель, и, как следствие, очень ощутимо

увеличивают быстродействие локальной сети в целом. Также, как казалось бы, они надежно защищают от прослушивания трафика: пакет не попадает на сетевые карты других абонентов — он идет только из кабеля отправителя в кабель получателя.

Но уязвимость все же была обнаружена. Оказалось, что у многих известных производителей сетевого оборудования логика работы коммутаторов такова, что, если на каком-либо из портов происходит переполнение списка MAC-адресов, то коммутатор начинает временно работать в режиме повторителя относительно всех своих портов, дабы не привести к полному разрыву сети относительно этих не поместившихся в список MAC-адресов, т. е. для превращения такого коммутатора в повторитель программе-генератору пакетов достаточно через определенный интервал времени устраивать в сети "шторм" из произвольных сетевых пакетов с произвольными различными MAC-адресами. Далее она будет получать все пакеты, идущие по сети через атакованный коммутатор.

Мерами защиты от подобной атаки являются:

- ☐ постоянный мониторинг сети на предмет MAC-штормов;
- ☐ анализ регистрационного журнала коммутатора, если таковой ведется данным устройством;
- ☐ активация на всех портах, к которым подключены одиночные хосты, режима ограничения количества MAC-адресов или даже установка фиксированного разрешенного MAC-адреса, если какой-либо из этих режимов поддерживается коммутатором.

Еще одна возможность прослушивания коммутируемого Ethernet-трафика основана на специфике и ошибках протокола STP (Spanning Tree Protocol). STP используется коммутаторами для обнаружения и блокирования петель в сегментах, состоящих из нескольких коммутаторов. В адрес подавляющего большинства современных реализаций этого протокола есть возможность отправить сфальсифицированный служебный STP-пакет, после которого коммутатор на несколько секунд перейдет в режим повторителя относительно всех своих портов.

До сих пор речь шла о прослушивании незащищенного, нешифрованного трафика. Однако, если пакеты криптографически защищены даже на сетевом уровне, то это не означает, что прослушивание такого информационного потока абсолютно бесполезно. Во-первых, шифрованный пакет есть предмет для криптоанализа. Во-вторых, на основе знаний об объемах трафика и периодичности сетевой активности можно сделать определенные выводы о работе сети. И в-третьих, это возможность проводить статистический и корреляционный анализ шифрованного трафика с нешифрованным (например, пакетами службы сервера имен), который также может представить злоумышленнику полезную информацию.

Активные атаки



Проще всего активные атаки проводятся, если хост злоумышленника является узлом какого-либо уровня (коммутатором, маршрутизатором, прокси-сервером, почтовым сервером) на пути движения информации, т. е. устройство, управлением над которым получил злоумышленник, должно иметь возможность фильтровать интересующий его трафик и направлять его на другой алгоритм обработки или даже по другому сетевому маршруту. Более сложные алгоритмы овладения потоком информации, связанные с сетевой маршрутизацией, рассмотрены в разд. "Атаки на основе сетевой маршрутизации" данной главы.

Если злоумышленник имеет подобным образом сконфигурированный хост-шлюз, то он имеет широкие возможности для работы с сетевым трафиком. Когда передаваемый трафик не имеет криптографической защиты ни на одном сетевом уровне, злоумышленник может полностью им манипулировать. Если же криптографическая защита применена, то тем не менее остаются некоторые возможности для ряда интересных атак. Оставим в стороне возможности криптоанализа, так как, если шифр успешно взломан, то трафик становится для злоумышленника открытым, по крайней мере, на время действия данного криптографического ключа.

Атака повтором



Атака повтором (англ. *replay attack*) возможна только в том случае, если в системе защиты пакетов отсутствуют или не включены механизмы обнаружения повторного приема одного и того же пакета. Сама атака заключается в том, что перехваченный и сохраненный злоумышленником пакет посылается повторно (возможно, несколько раз) в надежде на то, что он будет повторно принят к обработке системой получателя. В наиболее фантастическом варианте на банковский счет злоумышленника будет несколько раз зачислена одна и та же сумма. В менее фантастическом варианте, если система, например, в ответ на полученный пакет должна произвести серьезные вычислительные действия, то повторное направление нескольких пакетов может привести к перегрузке и отказу в обслуживании. Механизмом защиты от таких атак является проверка в каждом пакете *отметки времени* его отправки (англ. *timestamp*) или последовательного номера пакета. В качестве графического разъяснения атаки повтором приведем рис. 12.1.

Пример несколько утрирован хотя бы потому, что в системах межбанковских платежей существуют различные, в том числе некриптографические, способы защиты от подобных махинаций, например, регулярный обмен между банками информацией по подтверждению переводов. Но для демонстрации атаки повтором такой схемы вполне достаточно.

1. Злоумышленник открывает счет в Банке 2, а из Банка 1 переводит на этот счет некоторую сумму S.

2. Банк 1 готовит электронный документ — поручение Банку 2 о зачислении на счет C1 указанной суммы, применяет к документу методы криптозащиты и отправляет в Банк 2.
3. На этом этапе злоумышленник перехватывает (копирует) зашифрованное сообщение о переводе.
4. Банк 2, получив документ и проверив его корректность с точки зрения криптозащиты, исполняет поручение.
5. Злоумышленник, пусть даже не имея возможности обойти средства криптозащиты, просто повторно направляет перехваченный документ (возможно, через несколько дней), в адрес Банка 2.
6. Поскольку в документ не было внесено изменений, Банк 2 воспринимает его как корректный и повторно выполняет поручение на зачисление средств.

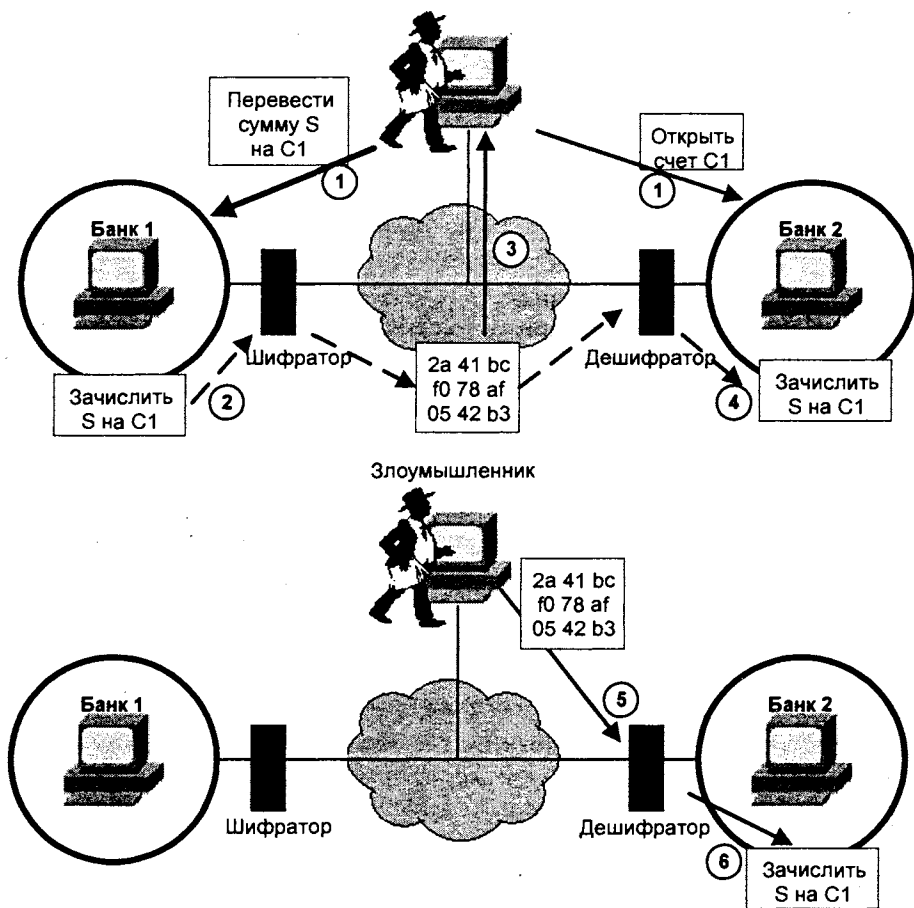


Рис. 12.1. Атака повтором на финансовое учреждение

Атака "злоумышленник-посредник"



Данный вид атак — "Злоумышленник-посредник" (англ. *man-in-the-middle*) породил целое новое направление в электронном бизнесе, а именно, создание центров сертификации (СА — *certification authority*). Для понимания сущности атаки необходимо представление о функционировании криптографической системы с открытыми ключами. Подробные системы с открытыми ключами (или как они еще называются, асимметричные системы) описаны в *части V*, посвященной криптографии, в данной главе рассказ о них пойдет в общих чертах.

Для успешного защищенного взаимодействия пары пользователей А и Б каждый из них должен сгенерировать пару криптографических ключей — открытый и закрытый, одним из которых можно только шифровать, другим — только дешифровать. Открытый ключ, в отличие от закрытого, не требует сохранения в конфиденциальности и отсылается удаленному адресату, т. е. А посылает свой открытый ключ (ОА) Б, а Б посылает свой открытый ключ (ОБ) А. Теперь, если А хочет отправить Б зашифрованное сообщение, то он шифрует его открытым ключом ОБ. Зашифровать такое сообщение может кто угодно, поскольку открытый ключ известен всем желающим. Но дешифровать сообщение сможет только обладатель закрытого ключа Б (ЗБ), т. е. сам Б. Идентификацию отправителя, зашифровавшего сообщение, можно производить по второй паре ключей, т. е. А может вставить в сообщение для Б некий элемент, зашифрованный закрытым ключом А (ЗА), а так как у Б есть открытый ключ А, то он сможет дешифровать этот элемент и проверить, что он был действительно зашифрован обладателем секретного ключа А, т. е. самим А (на этом принципе работает механизм электронной цифровой подписи — ЭЦП).

Данный защищенный механизм работает на основе убеждения, что открытый и закрытый ключи (ОА и ЗА) принадлежит именно А, а открытый и закрытый ключи (ОБ и ЗБ) принадлежит именно Б. Однако иногда абоненты защищенной связи могут быть введены в заблуждение. Было уже отмечено, что открытые ключи не требуют сохранения их в конфиденциальности, и это действительно так. Однако отправление открытых ключей адресатам без доверенного контроля пути следования как раз и приводит к возможности реализации описываемой атаки.

Предположим, что между А и Б имеется шлюз, управляемый злоумышленником — З. Тогда при подготовке к общению в защищенном режиме (или при очередной смене ключей), когда А направляет Б свой открытый ключ, З перехватывает его и сохраняет у себя. Вместо этого он отправляет Б открытый ключ из пары, сгенерированной им самим, т. е. открытый ключ ОА'. Когда Б направляет А свой открытый ключ, З также перехватывает его и сохраняет у себя. Вместо этого он отправляет А открытый ключ из другой пары сгенерированной им самим, т. е. открытый ключ ОБ'. В общем случае пары и ключи ОА' и ОБ' могут совпадать — это не принципиально. Важно то, что теперь

злоумышленник выступает прозрачным посредником в защищенном обмене информацией между А и Б, которые общаются друг с другом, не подозревая, что на самом деле их обмен данными контролируется (рис. 12.2).

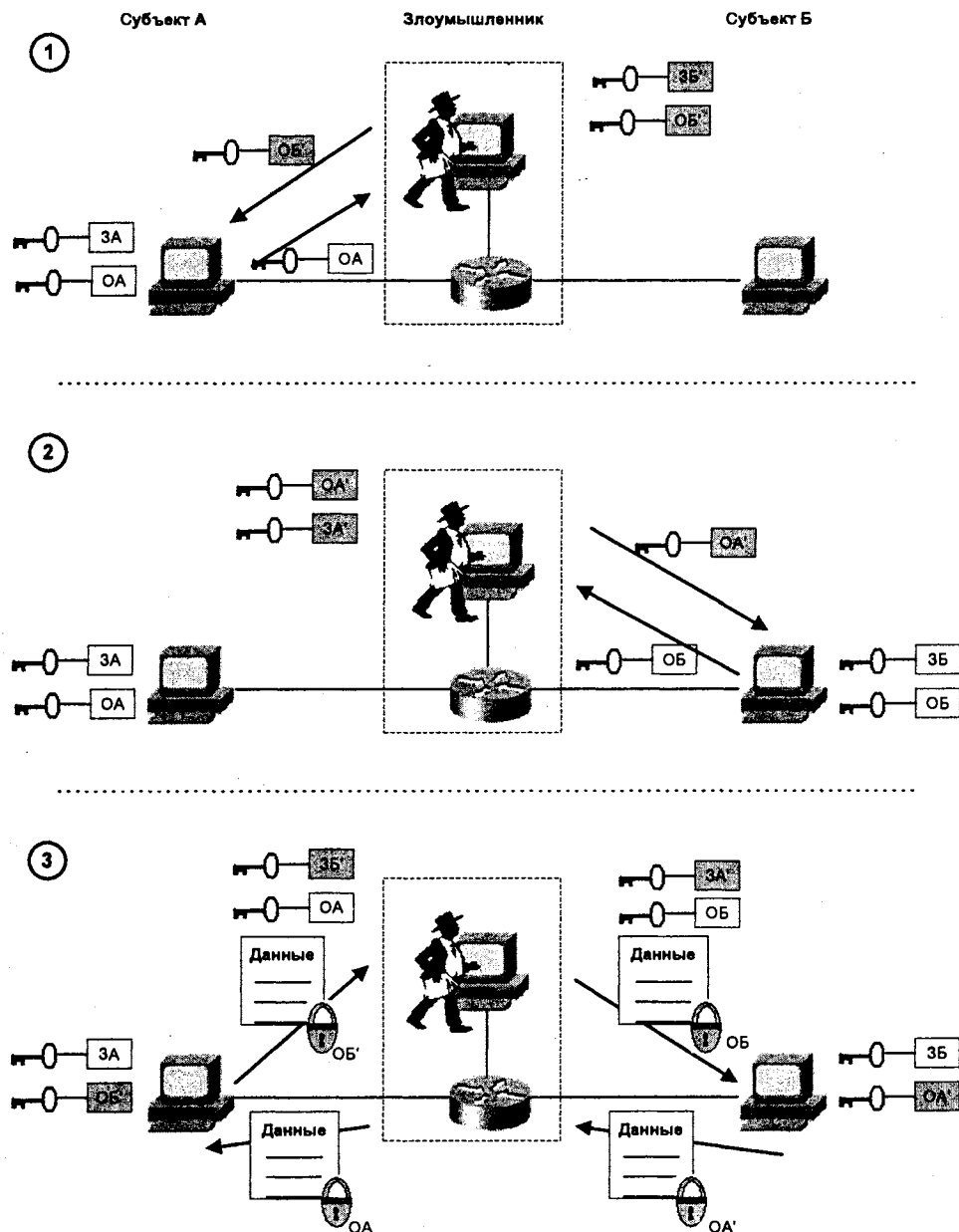


Рис. 12.2. Атака класса "злоумышленник-посредник"

Для предотвращения атак подобного класса необходимо обеспечение доверенной доставки открытых ключей абонентам. Если А и Б находятся в личном контакте или имеют альтернативный доверенный канал обмена информацией, то задача решается достаточно просто. Но если они находятся в разных концах света и общаются только с помощью Интернета, задача доверенной доставки становится нетривиальной.

Организации, занимающиеся сертификацией открытых ключей — СА, обеспечивают установку своей цифровой подписи на набор параметров пользователя сети, включающих его идентификационные данные и открытый ключ — т. н. цифровой сертификат. После этого сертификат с открытым ключом может быть свободно распространен без опасности подмены. Вопрос об удостоверении полномочий данного абонента в получении сертификата решаются, например, таким образом: пользователь устанавливает защищенную сессию связи с Веб-сервером СА, на сервере генерируется пара ключей, причем на сервере сохраняется только копия открытого ключа, а секретный ключ сохраняется только клиентом. После этого СА дополнительно альтернативными методами (факсом, обычной почтой и т. д.) получает подтверждение авторизованности пользователя, сгенерировавшего ключи, формирует для него сертификат, отправляет пользователю и сохраняет копию сертификата у себя для дальнейшего использования. Иногда СА сохраняет у себя и секретный ключ пользователя на случай его утери, но это уже предмет договорных отношений между самим пользователем и СА.

Атаки на основе сетевой маршрутизации



Возможность смены штатного маршрута движения информации может оказаться чрезвычайно необходимой злоумышленнику как при пассивных, так и при активных атаках. Достижимо это в сетях с динамической маршрутизацией пакетов (то есть направление дальнейшего следования для каждого очередного пришедшего пакета маршрутизатор выбирает, исходя не из своих настроек, а из сетевой обстановки). В тех случаях, когда злоумышленник может формировать фальшивые служебные пакеты протоколов динамической маршрутизации, и эти пакеты затем принимаются маршрутизаторами как корректные, а у него появляется возможность манипулировать сетевым трафиком в самой произвольной форме (рис. 12.3).

В некоторых протоколах динамической маршрутизации для перестраивания логики отправки пакетов достаточно отправить всего один служебный пакет с определенным образом сформированной фальшивой информацией о сетевой обстановке. В других — для перенаправления трафика в сторону другого узла необходим постоянный поток фальшивых сетевых пакетов. При этом совсем необязательно, чтобы через хост злоумышленника проходил альтернативный путь следования трафика: он может находиться и в сетевом тупике — специальное (хотя и довольно сложное) программное обеспечение превратит его машину в транзитный узел.

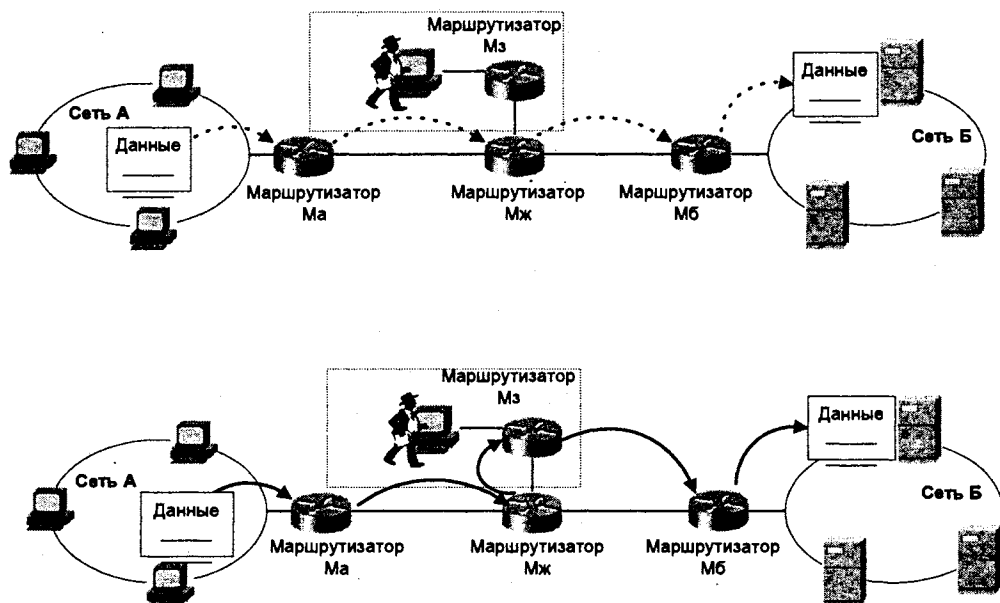


Рис. 12.3. Перехват маршрута

1. Нормальное прохождение маршрута от сети А через маршрутизатор Ма, находящийся под управлением администратора сети А, далее через маршрутизатор Мж — жертву атаки — на маршрутизатор Мб, находящийся под управлением администратора сети Б, и в сеть Б. При этом злоумышленник, — администрирующий маршрутизатор Мз, пока не вмешивается.
2. Результат после успешной атаки на маршрутизатор-жертву Мж. Поток данных перенаправляется на маршрутизатор злоумышленника Мз.

Где невозможен данный класс атак?

- ☐ В сетях со статической маршрутизацией.
- ☐ В протоколах динамической маршрутизации, авторизующих свои служебные пакеты с помощью криптографических механизмов.
- ☐ Извне в сетях, где маршрутизатор сам выполняет функции межсетевого экрана, причем он правильно настроен — служебный трафик протоколов динамической маршрутизации не должен поступать в систему, если он ей не нужен.
- ☐ Изнутри в сетях с правильно настроенными межсегментными сетевыми экранами — в этом случае при атаке подобного рода информационный поток просто прервется, но не будет доставлен злоумышленнику и, следовательно, сохранит конфиденциальность и целостность.

Кроме того, необходимо отметить, что атака на смену сетевых маршрутов трудно реализуема в сетях со стыками различных протоколов динамической маршрутизации.

Перехват сессии



После того, как злоумышленник научился обходить аутентификацию, основанную на сетевом адресе (путем подмены этого адреса в пакетах), вопрос обеспечения доверия к сеансу работы хоста с хостом переместился на более высокий уровень. Хост-сервер, например, может производить предварительную идентификацию и аутентификацию хоста-клиента, устанавливающего сеанс работы, и далее учитывать установленную сессию как доверенную.

В стандартном наборе протоколов TCP/IP функцию сессионного протокола выполняет TCP, обеспечивая контроль (идентификацию) сессии путем последовательной нумерации каждого нового пакета уникальными номерами (sequence number и acknowledge number), увеличивающимися на длину переданной информации от пакета к пакету, при этом начальное значение каждого из этих двух номеров генерируется случайно на этапе установления соединения. Если сервер не предпринимает дополнительных мер по идентификации пакетов, то злоумышленнику достаточно знания о текущих номерах сессии, чтобы сформировать свой пакет, в котором адрес отправителя будет адресом клиента, а оба номера пакета будут соответствовать тем, которые он узнал. Доверенному клиенту будет послан пакет разрыва сессии — таким образом злоумышленник сможет продолжать работу в сессии от имени доверенного хоста (перехватить сессию).

Если злоумышленник имеет возможность установить свой хост или свое программное обеспечение на компьютере, расположенном в том же сегменте, что и клиент или сервер, либо на канале передачи данных между ними, то, используя средства пассивного прослушивания сети (сниффер), он может получить информацию о номерах из перехваченных пакетов. Если же такой возможности у него нет, то он может попытаться предугадать эти номера. Дело в том, что ряд реализаций сервисов TCP в операционных системах имеет такие алгоритмы генерации начального случайного числа, что оно может быть предсказано на основе предварительных знаний об уже сформированных пакетах других сессий (которые обычно иницируются самим злоумышленником с этой единственной целью).

Данный пример обычно приводится в литературе в ситуации, когда злоумышленник не имеет возможность непосредственно перехватывать поток сообщений между хостами А и Б. В этом случае злоумышленник инициирует соединение от имени хоста А, выполняет атаку "отказ в обслуживании" на хост А (чтобы тот не смог оповестить хост Б, что на адрес А пришли пакеты о начале сессии), и, предугадывая порядковые номера пакетов, про-

должает соединение (точнее, односторонний поток, так как ответные пакеты от Б не вернутся к злоумышленнику) с хостом Б.

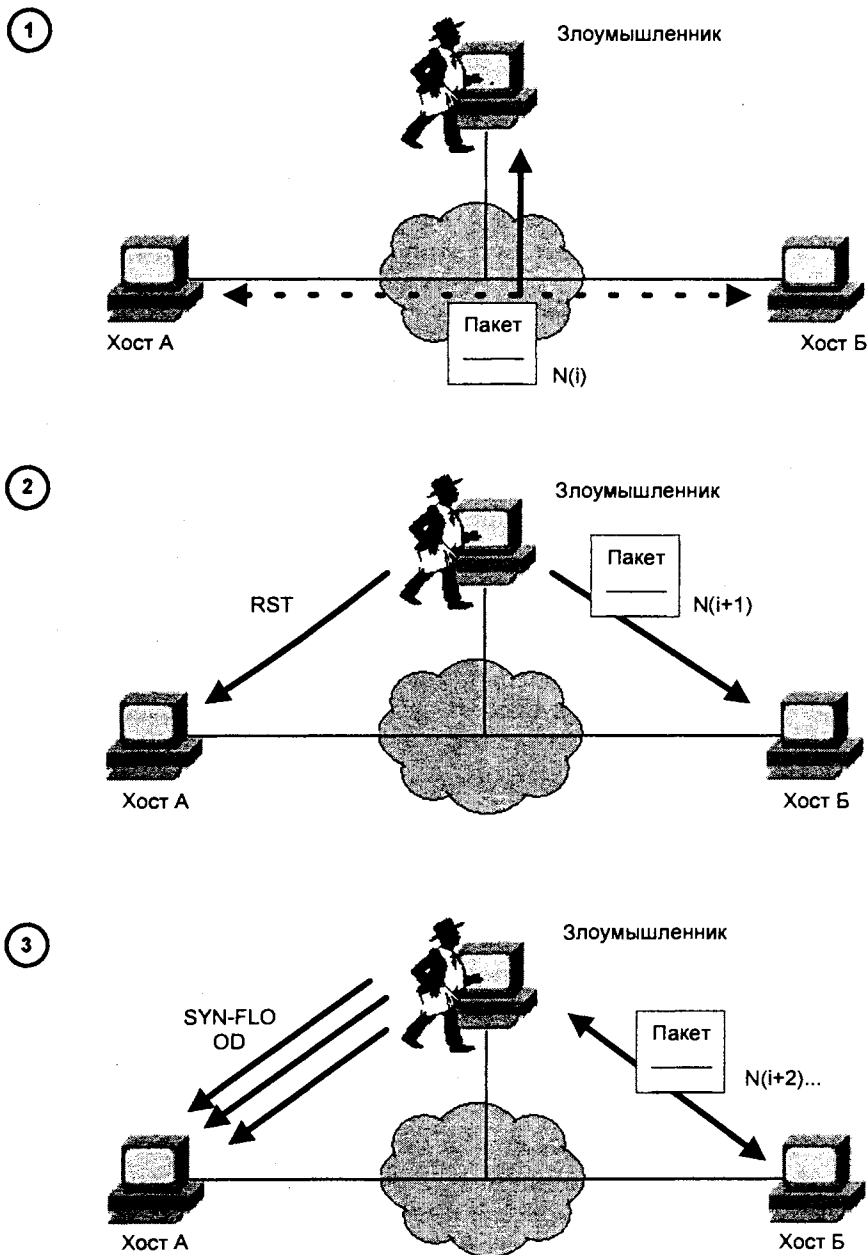


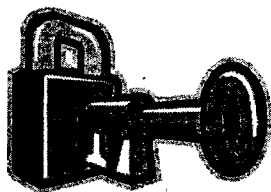
Рис. 12.4. Перехват авторизованной сессии

В нашем случае схема упрощена (злоумышленник может перехватывать пакеты сессии А-Б), но интересна тем, что атака может начаться после того, как пользователь хоста А был аутентифицирован и авторизован на хосте Б, т. е. получил некие расширенные права (рис. 12.4).

1. Злоумышленник "слушает" трафик А-Б, ожидая авторизации пользователя А на хосте Б.
2. Злоумышленник посылает на хост Б следующий по очереди пакет как продолжение нормальной сессии А-Б. На хост А он посылает сигнал о разрыве соединения, хотя это больше формальность, поскольку хост А далее выводится из работы любой из доступных атак класса *Deny of Service* (Отказ в обслуживании), например, атакой SYN-FLOOD.
3. Продолжая сессию от имени А, злоумышленник обычно поддерживает хост А в состоянии отказа от обслуживания.

Учитывая то, что даже некоторые межсетевые экраны (из тех, которые вполне могут находиться на службе), уделяя внимание аутентификации клиента при установлении соединения, не контролируют далее аутентичность пакетов, описанная атака — перехват авторизованной сессии, вполне может быть использована.

Злонамеренные воздействия могут быть применены на любом участке информационного потока, и ко всем им следует быть готовым.



Часть IV

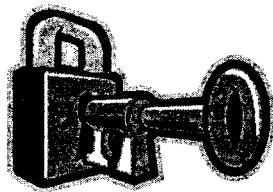
Адекватность механизмов защиты

Глава 13. Управление рисками

Глава 14. Методики оценки рисков

Глава 15. Высоко вероятные атаки

Глава 13



Управление рисками

Что такое "управление рисками"

Посещая семинары и презентации различных систем и продуктов по информационной безопасности, порой можно увидеть следующую картину. Представитель компании, представляющий продукт, расписывает в ярких красках достоинства и особенности своей системы, из которых можно сделать вывод, что данная система — это все что необходимо для обеспечения безопасности организации (пусть даже отдельного направления деятельности или отдельной подсистемы). При этом можно увидеть, как разгораются глаза у участников — посетителей семинара, специалистов, так или иначе обеспечивающих или отвечающих за безопасность конкретных предприятий. Можно с высокой степенью вероятности предположить, что некоторые из них по окончании семинара-презентации направятся к своему руководству с предложениями о приобретении рекламируемой системы, особенно если предприятие государственное или такое, где специалисты напрямую не заинтересованы в рациональном расходовании средств.

Отвлекаясь от достоинств или недостатков любой из систем или продуктов, необходимо осознать то, что защита информационных ресурсов должна быть продумана и эффективна, поскольку она в определенной степени предназначена и для сохранения финансовых ресурсов организации. В этом смысле приобретение дорогого, разрекламированного средства информационной безопасности может идти вразрез с самими целями мероприятий по информационной безопасности.

Часто можно услышать подобную логику рассуждений: на данном компьютере лежит конфиденциальная информация, следовательно, поместить его за межсетевой экран (или установить на нем персональный межсетевой экран), антивирусное (в данном случае анти-троянское) программное обеспечение, сканнер атак на хост, программу контроля целостности системных файлов и вдобавок ко всему оснастить его системой биометрического контроля доступа, скажем, по отпечатку пальца. Суммарные расходы на защиту компьютера начинают превышать стоимость самого компьютера с опера-

ционной системой и приложениями. (При этом, правда, иногда упускается из виду, что оператор, работающий на данной станции с указанными секретными данными, готов поделиться этими секретными данными за сумму, сравнимую с его месячной заработной платой, возможно, это несколько десятков, пусть даже сотен долларов.) Если же, как это обычно бывает, информация распределена между станциями сети, стоимость защиты вырастает в существенные суммы. Однако какой конкретный ущерб принесет разглашение этой информации? Очевидно, что это неприятно, особенно если такой инцидент стал достоянием публики. Но каково материальное выражение этого ущерба?

Некоторые руководящие документы по информационной безопасности косвенно или явно формально указывают на основной принцип определения требуемого уровня защиты. Обычно он формулируется так: "Сумма затрат на обеспечение защиты не должна превосходить суммы ущерба от атаки, которую данная защита должна предотвратить". Реже принимается другой принцип: "Суммарные расходы, понесенные злоумышленником на преодоление защиты должны превышать выгоду от результатов атаки". Первый подход обычно характерен для коммерческих организаций, ориентированных на получение прибыли, второй подход — для организаций работающих, например, с государственными секретами, когда построение защиты подразумевает более существенные расходы.

Однако, если измерить произведенные расходы на приобретение и установку систем защиты информации достаточно легко, то оценить, сколько необходимо потратить на обеспечение безопасности, сложнее. В *главе 14* будут представлены некоторые методики такой оценки, но следует учитывать, что увязывание числовых метрик с информационными активами вообще специфично для каждой конкретной организации, а на постсоветском пространстве еще и вносит дополнительные особенности. Скажем, если разглашение информации по компрометирующим действиям западного политика может привести к его отставке, то на территории бывшего СССР это может наоборот прибавить политике популярности.

Управлением рисками следует заниматься не только при планировании глобальных модификаций информационного пространства организации, но и в ряде более мелких случаев. Например, организация использует большую автоматизированную систему, приобретенную у стороннего поставщика, но при этом она открыта к модификации (в ней присутствуют детализированные описания API). Одним из элементов обеспечения информационной безопасности является анализ регистрационных журналов системы. Однако стандартная поставка системы такова, что журнал неполон и дополнительную информацию приходится собирать из других подсистем (допустим, из журналов операционной системы), для чего необходима дополнительная

работа, которая выливается в одну штатную единицу оператора мониторинга журналов. Возможные альтернативные решения имеющейся проблемы:

- ☐ принять на работу оператора для выполнения необходимых работ;
- ☐ заказать у поставщика обновление системы с расширенным регистрационным журналом;
- ☐ поручить своим программистам разработать дополнительный модуль с помощью имеющихся API;
- ☐ приобрести отдельный продукт, который будет производить анализ разрозненных регистрационных журналов, возможно, как одну из подфункций к другой полезной деятельности.

Выбирать подходящий вариант из имеющихся можно только после оценки потенциального ущерба оттого, что атака будет реализована по причине отсутствия каких-либо работ по анализу регистрационных журналов системы. Ведь возможно, что стоимость реализации любого из четырех приведенных проектов окажется значительно больше вероятных потерь организации в случае атаки.

Оценка рисков как часть направления информационной безопасности — *управления рисками*, является существенным инструментом в построении защиты. Однако для эффективного использования этого инструмента необходимо выполнить ряд условий, например, перейти от качественных понятий к количественным. Скажем, "получение доступа к данной информации приведет к краху компании" — это качественное описание, а "разглашение данной информации потребует выплаты суммы n_1 клиентам, n_2 конкурентам, n_3 судебные издержки" — это количественное описание. Этот процесс значительно облегчен, если в организации уже произведена классификация информационных активов, о которой рассказывалось в *главе 6*.

Управление рисками — это процесс определения, анализа и оценки, снижения, устранения или переноса (перенаправления) риска, который (процесс) заключается в ответе на следующие вопросы.

1. Что может произойти?
2. Если это "что-то" произойдет, то каков будет результат или ущерб?
3. Как часто может происходить это событие?
4. Насколько мы уверены в ответах на вышеуказанные вопросы (оценка вероятности)?
5. Что может быть сделано для снижения или устранения вероятности события?
6. Сколько будет стоить то, что может быть сделано?
7. Насколько эффективно то, что может быть сделано?

Сам риск как таковой и состоит из понятия вероятности (четвертый вопрос), т. е. чем больше вероятность первых трех вопросов (событие проис-

ходит часто и с большим ущербом), тем больше риск, чем меньше вероятность, тем меньше риск.

Информационный актив — набор информации, который используется организацией в работе и может состоять из более мелких поднаборов. При оценке должно быть проведено отделение собственно информации, как виртуальной составляющей, от ее носителя, как физического объекта. Соответственно стоимость возможного ущерба должна быть оценена как сумма:

- ☐ стоимости замены информации (если была произведена ее утрата в том или ином виде);
- ☐ стоимости замены программного обеспечения поддержки (если оно было повреждено);
- ☐ стоимости нарушения конфиденциальности, целостности, доступности (если таковые имели место).

Отдельно учитываются аппаратное и сетевое обеспечение, поскольку их оценка производится проще (они имеют известную рыночную стоимость).

Методика управления рисками в целом



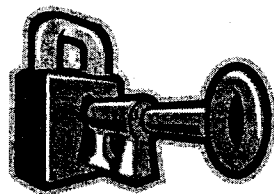
Стандартная методика организации работ по управлению рисками следующая:

1. Определение политики управления рисками. Построенная на общепринятых принципах обеспечения информационной безопасности (англ. *Generally Accepted System Security Principles — GASSP*) [GASSP] политика позволит избежать субъективного подхода.
2. Определение персонала, который будет заниматься управлением рисками и обеспечить его финансирование. Помимо оплаты труда, скорее всего придется провести обучение кадров и, возможно, приобрести автоматизированные инструменты или платные методики оценки рисков.
3. Определение методологии и средств, с помощью которых будет производиться оценка риска. Важно быть уверенным в том, что оценка риска проведена правильно до того как будут потрачены средства, необходимые для управления рисками.
4. Идентификация и измерение риска. На первом этапе необходимо определить сферу применения работ, имеющиеся угрозы, информационные активы и их значимость (это, возможно, уже будет сделано при классификации), проанализировать уязвимости активов, которые могут повлиять на частоту появления или размер возможного ущерба. Далее производится сведение полученных данных с установкой метрик. Для качественной оценки это может быть таблица, в колонках которой указаны активы, а в столбцах — уровень риска (высокий, средний, низкий).

Для количественной оценки используется конкретная количественная методика (например, любая из приведенных в *главе 14*).

5. Установка критериев приемлемости рисков. На основе полученных данных специалисты по управлению рисками совместно с руководством организации должны определить приемлемость риска на основе принятой методики, например, считать неприемлемым риск, если вероятность потери эквивалента 100 000 долларов США более чем 3/100.
6. Избежание или уменьшение рисков. Необходимо определить уязвимости, которые становятся неприемлемыми при принятых критериях, и определить меры для их устранения. Этот процесс проходит следующие фазы: выбор средств для снижения/устранения риска оценка эффективности этих средств (в смысле соответствия между ценой средства и его эффективностью), отчет руководству о предлагаемых мероприятиях.
7. Мониторинг работы управления рисками. Для обеспечения адекватности предпринимаемых мер соответствующим рискам, необходимо периодически производить переоценку рисков при изменении внешних и внутренних обстоятельств, угроз и другие превентивные действия.

Глава 14



Методики оценки рисков

В этой главе будут рассмотрены три методики оценки рисков:

- ☐ модель качественной оценки;
- ☐ традиционная (для Запада) методика под условным названием *Количественная модель рисков* (англ. *Quantative Risk Model — QRM*);
- ☐ именная методика *Модель обобщенного стоимостного результата Миоры* (англ. *Miora Generalized Cost Consequence Model*).

Модель качественной оценки



Качественная оценка обычно сводится к построению таблицы следующего вида (рис. 14.1).

Риск атаки \ Важность актива	Важный	Критический	Жизненный
Низкий			
Средний			
Высокий			

Рис. 14.1. Таблица качественной оценки

Таблица заполняется экземплярами информационных активов или отдельными системами на основе интуитивного представления заполняющего о той или иной информации, или на основе заполнения подготовленных вопросников, анкет для компетентных сотрудников организации. Решение принимается в зависимости от конкретной организации для объектов, лежащих в черной зоне либо в черной и серой зонах.

Возможен еще более упрощенный подход под названием *основная линия* (англ. *Baseline*), который заключается в том, что организация анализирует об-

становку с построением систем безопасности, сложившуюся в отрасли (возможно, в организациях, схожих по профилю), и сравнивает со схемой безопасности, построенной в самой компании. Если обнаруживаются отставания, ресурсы направляются на приведение ситуации к уровню, близкому к среднему в отрасли. Скажем, если все компании по электронной торговле затратили средства на приобретение цифрового сертификата, то данной конкретной компании можно принять решение о приобретении сертификата без сложных расчетов целесообразности его приобретения. Однако, если крупная компания приобрела систему обнаружения атак, распределенную по всем сегментам своей сети, то, возможно, более мелкой компании не стоит расходовать такие значительные средства, ограничившись защитой сегмента, на котором размещены серверы компании. Это уже вопрос управления рисками.

Положительные стороны оценки риска по качественной модели заключаются к следующему:

- вычисления ускоряются и упрощаются;
- нет необходимости присваивать денежную стоимость активу;
- нет необходимости вычислять частоту проявления угрозы и точный размер ущерба;
- не нужно вычислять соответствия эффективности предполагаемых мер угрозам.

Отрицательные стороны заключаются в основном в субъективности подхода к оценке и отсутствии возможности установить точное соответствие затрат угрозам.

Количественная модель рисков

Общая методика



Количественная модель рисков оперирует такими понятиями, как:

- *годовая частота происшествий* (англ. *Annualized Rate of Occurrence — ARO*), иначе говоря, вероятность появления ущерба;
- *ожидаемый единичный ущерб* (англ. *Single Loss Expectancy — SLE*), т.е. стоимость ущерба от одной успешной атаки;
- *ожидаемый годовой ущерб* (англ. *Annualized Loss Expectancy — ALE*), величина, равная произведению *ARO* на *SLE*.

$$ALE = ARO \times SLE,$$

где *ARO* — это частота появления события, приносящего ущерб на годовой основе, т. е., если событие происходит раз в 5 лет, то величина *ARO* равна 1/5

или 0,2, а если событие происходит 3 раза в год, то *ARO* равно 3. Как видим, эта величина отлична от математической вероятности, которая не может быть больше 1. Величина *ARO* не ограничена сверху. Если в организации из 100 человек 50 ежедневно эксплуатируют информационную систему, при этом 5 из них, обладая высокими правами, но низкой квалификацией, могут ежедневно допускать серьезные ошибки, приводящие к нарушениям в работе системы, то *ARO* для этой системы по данному событию будет равна $5 \times 12 = 60$.

SLE рассчитывается как произведение количественного (стоимостного) значения актива (англ. *Asset Value* — *AV*) на фактор воздействия (англ. *Exposure Factor* — *EF*). Фактор воздействия — это размер ущерба или влияния на значение актива (от 0 до 100%), т. е. часть значения, которую актив потеряет в результате события.

$$SLE = AV \times EF$$

Рассмотрим пример. Предположим, имеется здание с внутренней инфраструктурой общей стоимостью 10 000 000 долларов. Пожар может нанести ущерб с фактором воздействия 30% (здание и часть инфраструктуры сохранится). Пожар может случиться раз в 10 лет. Тогда:

$$SLE = 10\,000\,000 \times 0,3 = 300\,000$$

$$ALE = 300\,000 \times 0,1 = 30\,000$$

Таким образом, если организация будет тратить до 30 000 долларов в год на предотвращение риска пожара, то предпринимаемые меры будут эффективными с точки зрения управления рисками.

Определение вероятности события



Несмотря на предельную простоту формул, исчисление рисков по методике QRM таит в себе другую сложность: расчет составляющих.

Первым компонентом формулы является *ARO*. Как вычислить вероятность того или иного инцидента? Если разбить угрозы по категориям, то многие вопросы прояснятся. Так, по вопросам угроз, зависящих от климатических условий (затопления, ураганы, обледенения), надо обратиться в локальное или государственное метеорологического бюро и получить там необходимые статистические данные. По пожарам — в управление пожарной охраны, по землетрясениям — в сейсмический центр. Компьютерные преступления могут быть закрытой информацией, либо учитываться органами внутренних дел по другим статьям, тогда необходимо обратиться к доступной мировой статистике, делая поправку на то, что определенный процент таких преступлений скрывается пострадавшими компаниями. Статистику отказов оборудования и программного обеспечения можно запросить у поставщиков и, учитывая то, что производители склонны приукрашать свой товар, обратиться к имеющимся независимым экспертным оценкам.

Определение стоимости активов



Следующий тонкий вопрос — определение стоимости активов. Разделим их на осязаемые и неосязаемые активы. К осязаемым отнесем средства обслуживания информационных технологий (ИТ) — аппаратное обеспечение, сетевое обеспечение (среда передачи данных), запасные части, документация и зарплата персонала для поддержки функционирования систем. Стоимостные характеристики этих активов обычно известны либо легко вычисляемы.

Стоимость неосязаемых активов должна учитывать два вида расходов: расходы на замену/восстановление программного обеспечения и данных, расходы при нарушении конфиденциальности/целостности/доступности.

Определение стоимости неосязаемых активов — задача не всегда простая. Существуют отдельные методики такой работы, например, *Guideline for Information Valuation (GIV)* — Инструкция по Оценке Стоимости Информации, который доступен зарегистрированным членам Information Systems Security Association на сайте <http://www.issa.org/>. Приведем пример, который поможет понять принципы такого определения.

Предположим, группа специалистов, работает над исследованием, результатом которого должен стать некоторый набор данных (НД), неосязаемую стоимость которого необходимо оценить. Если НД будет утерян на каком-либо из этапов исследования, то это означает, что для его восстановления скорее всего потребуются следующие средства:

1. Если НД сохранился как интеллектуальный продукт (в виде мыслей, идей, возможно, черновиков), то затраты составят расходы на восстановление его на носителях, т. е. труд операторов по вводу данных, скачиванию, корректировке и т. д.
2. Если НД не сохранился ни в каком виде, то расходы составят столько же, сколько было потрачено на работу данной группы до того этапа, когда НД был утерян, при условии что актуальность разработки не утрачена. Если сохранилось какое-либо оборудование или материалы, которые можно использовать в повторном исследовании, то их следует вычесть из общих расходов.

Для оценки стоимости по конфиденциальности необходимо исследование ряда дополнительных условий:

1. Если разглашение информации о деталях исследования никак не повлияет на деятельность группы, то НД не имеет стоимости в этом смысле.
2. Если разглашение информации о НД, например, конкурентам, означает крах всего исследования, то расходы в этом случае по порядку величины равны организации новых исследований, т. е. затратам в позиции 2.
3. Если работы группы были сегментированы таким образом, что НД представляет собой отдельные объекты, не связанные друг с другом, то, воз-

можно, нарушение конфиденциальности одного из них не повлияет на состояние конфиденциальности остальных, расходы составят меньшую часть.

Для оценки стоимости нарушения целостности необходимо знать, насколько такое нарушение существенно для проекта или для работ. При этом следует различать смысловые ошибки самого исследования, ошибки персонала при работе с данными, ошибки оборудования, изменения, внесенные злоумышленниками, и т. п. Также следует различать данные по чувствительности к нарушениям целостности. В номере банковского счета или в сумме на этом счете изменение одной цифры может иметь более серьезные последствия, чем изменения 100 записей в базе данных библиотечного каталога в 10 000 записей.

1. Если изменение данных повлекло собой прямой материальный ущерб (недополучение или перерасход, как в примере с банковским счетом), то максимум такого возможного ущерба — это и есть стоимость нарушения целостности. Но при этом следует учитывать, что иногда максимальный ущерб не может быть нанесен просто потому, что будет моментально обнаружен и предотвращен, а иногда часть или весь размер этого ущерба возможно компенсировать путем возврата утраченных средств.
2. Изменение также может повлечь больший или меньший ущерб, оценить который могут, порой, только специалисты. Например, изменение в химической формуле или пропорциях химической смеси может привести к грандиозной катастрофе.

Нарушение доступности целесообразно различать по времени. Два основных вида недоступно в течение учитываемого периода времени и недоступны навсегда. Последняя ситуация аналогична случаям 1 и 2, а частичную недоступность следует классифицировать по промежуткам времени — какой ущерб последует, если НД не будет доступен (частично доступен) в течение минуты, часа, дня и т. д.

Для того чтобы не упустить из рассмотрения все возможные варианты получения ущерба, необходимо произвести связь между уязвимостью, угрозой и влиянием на актив. Пример такой связи приведен на рис. 14.2.

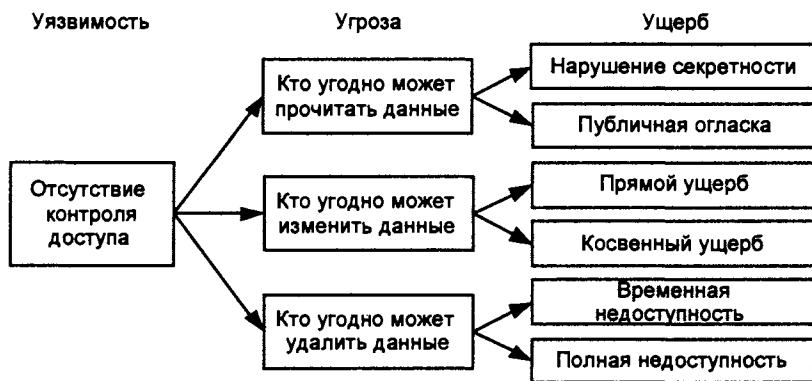


Рис. 14.2. Связь "уязвимость — угроза — ущерб"

Модель обобщенного стоимостного результата Миоры (GCC)



Модель GCC разработана как альтернатива Количественной модели рисков (QRM) для улучшения и облегчения расчетов и вычислений. Одним из основных недостатков QRM определяется ее вероятностная составляющая. Действительно, редко происходящие события с большим ущербом и часто происходящие события с маленьким ущербом часто требуют различных подходов в определении защитных мер, однако, как можно видеть из формул, выглядеть при расчетах они будут одинаково.

Модель GCC не учитывает вероятностей катастрофических событий, она оперирует понятием ущерба от простоя, как функцией от времени после наступления событий. Для каждого информационного актива или группы сходных по ряду признаков активов, называемых *категорией*, определяется размер возможного ущерба, срок начала его влияния на организацию и распределенность по времени.

Например, простой основного производства будет приносить ущерб в 50 000 долларов в день, начиная с первого дня. Простой службы, занимающейся заключением контрактов на продажу продукции, будет приносить 10 000 долларов в день, но начиная с пятого дня после начала простоя. И так далее. Затем все категории ущерба суммируются.

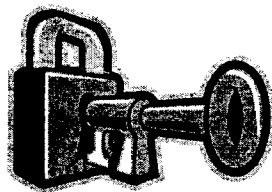
Иногда удобнее представить развитие картины ущерба в виде графика, где категории — это функции по двум осям "время в днях" — "ущерб в деньгах". Обычно в результирующем графике показаны две кривые:

- ❑ суммарный ущерб организации при отсутствии защитных мероприятий — *аварийного плана* (англ. *disaster recovery plan* — *DRP*);
- ❑ суммарный ущерб при наличии *DRP*.

На таком графике наглядно видна необходимость и эффективность применяемых мер для обеспечения защиты информации.

Модель различает три вида ущербов: прямой осязаемый ущерб, косвенный осязаемый ущерб и неосязаемый ущерб. Примерами первого вида являются категории, которые непосредственно участвуют в производстве прибыли организации, т. е. производственные активы. Второй вид — это в основном функции поддержки, активы, связанные с внешними источниками (например, на бирже убыток происходит за счет падения стоимости акций), и т. д. В третьем виде включены ущерб репутации, невыполненные обещания, негативное общественное мнение. Методики расчета категорий этих видов являются собственностью автора методики и могут быть получены по согласованию с ним. Автор — *Michael Miora* (mmiora@mmiora.com).

Глава 15



Наиболее вероятные атаки

Рейтинги атак

В этой главе пойдет речь о наиболее часто встречающихся компьютерных атаках как наиболее интересном проявлении угроз информационным ресурсам. При этом хотелось бы отметить, что, несмотря на повышенное внимание к взлому злоумышленниками информационных систем, созданное общественным мнением, на самом деле такие угрозы не всегда являются основной составляющей ущерба информационным системам. Статистические данные различных исследований отличаются, но тем не менее в среднем аварии и сбои электроснабжения порой приносят больший ущерб, чем атаки пресловутых хакеров. Техногенным и природным катастрофам будет уделено внимание в *части VIII*, а данная глава — посвящена человеческому фактору.

Дабы не оказаться субъективными, мы построили содержание этого раздела на основе двух очень интересных документов, созданных исследовательским институтом SANS (System Administration Networking and Security Institute — Институт администрирования, сетей и безопасности). Первый документ представляет собой список из 10 наиболее уязвимых продуктов и протоколов. Второй — расширенный список из 20 наиболее опасных уязвимостей, разделенный на три части:

- ☐ ошибки в методологии работы с информацией;
- ☐ уязвимости систем семейства Windows;
- ☐ уязвимости систем семейства UNIX.

Оригиналы документов размещены на сайте <http://www.sans.org/>. Кроме того, авторы добавили в ряде статей свои дополнения и комментарии.

10 наиболее уязвимых продуктов и протоколов по рейтингу SANS



1. *BIND (Berkeley Internet Name Domain)* — реализация службы Domain Name Service (Служба доменных имен), а именно ее сервисы *nxt*, *qinv* и *in.named*. Широко распространенная реализация для Unix/Linux систем, при версии 8.2.2 (системное обновление меньше 5-го) и ниже позволяет

злоумышленнику получить полный доступ (уровень root) к хосту, на котором запущена эта служба. Учитывая, что на сервере имен хранится существенная информация по конфигурации сети, данная уязвимость может быть использована как основа для дальнейшей атаки на критически важные хосты сети.

2. *Скрипты CGI и приложения Веб-серверов.* В поставку ряда Веб-серверов включены скрипты и приложения, которые не являются безопасными, т. е. при их реализации были упущены вопросы безопасности. В качестве примера уязвимого приложения приводится пакет приложений ColdFusion, часть программ-образцов которого являются уязвимыми.
3. *Remote procedure calls — RPC (Удаленный вызов процедур)* — это механизм, позволяющий одному компьютеру выполнять программы (процедуры) на другом компьютере, чаще всего используется при *совместной работе* с файлами многих клиентов (англ. *sharing*). В качестве примеров уязвимых реализаций приведены сервисы `rpc.ttdbserverd`, `rpc.cmsd`, и `rpc.statd`, компрометация которых позволяет злоумышленнику получить полный доступ (уровень root) к хосту.
4. *Microsoft Internet Information Server (IIS)* имеет компоненту под названием Remote Data Services — RDS (Сервисы удаленного доступа к данным), в которой присутствует уязвимость, позволяющая злоумышленнику удаленно исполнять на Веб-сервере команды с привилегиями администратора.
5. *Sendmail* — почтовая программа, которую можно назвать родоначальником в сфере нарушений информационной безопасности, ведь именно с использования уязвимости в этой программе "червем Морриса" в 1998-м официально началась эра компьютерных преступлений. До сих пор в этой программе находят уязвимости, в настоящее время считается, что использование `sendmail` версии ниже 8.10 автоматически означает возможность получения злоумышленником доступ на уровне root.
6. *Сервисы `sadmind` (Solaris) и `mountd` (Unix)* позволяют получить доступ к административным функциям и управляют доступом к сетевой файловой системе (Network File System — NFS) соответственно. Атаки на переполнение буфера позволяют получить доступа на уровне root.
7. *Совместное использование файлов (`sharing`)* характерно для всех систем. NetBIOS (TCP-порты 135, 139, 445), UNIX NFS (TCP-порт 2049), Macintosh Web sharing или AppleShare/IP (TCP-порты 80, 427, 548) — это примеры использования такого механизма. Иногда эту возможность используют для удобства пользователей в ущерб безопасности — открывают больше каталогов (директорий), чем необходимо; с большими правами, чем необходимо; для большего числа пользователей, чем необходимо. Такое неверное конфигурирование системы приводит к появлению уязвимостей.
8. *Использование аутентификационных данных пользователя (идентификатор+пароль) с пустым или легко разгадываемым паролем.* Эта уязвимость основана на человеческом факторе и характерна для всех систем.

В зависимости от прав, определенных для данного пользователя, злоумышленник может получить уровень доступа вплоть до root.

9. *Уязвимости в протоколах IMAP и POP*, предназначенных для предоставления удобства пользователям, в частности, обеспечения доступа из Интернета и внешних сетей к почтовому ящику пользователя в корпоративной сети. Если межсетевой экран организации сконфигурирован таким образом, что он пропускает пакеты по данным протоколам, то это может обеспечить злоумышленнику возможность атаки переполнением буфера на почтовый сервер.

Примечание

Наличие возможности прохода трафика этих протоколов через межсетевой экран не означает автоматически уязвимость системы. Многое зависит от конфигурации использования протоколов, межсетевого экрана, ряда других причин. Вернее сказать, что это, возможно, повышает вероятность появления уязвимости, если не используются соответствующие механизмы безопасности.

10. *Simple Network Management Protocol — SNMP (Упрощенный протокол управления сетью)* — используется сетевыми администраторами для мониторинга и управления сетевыми устройствами. Однако механизм, обеспечивающий защиту от перехвата злоумышленником управления устройством, довольно примитивен.

Примечание

Этот механизм характерен первоначальному варианту протокола SNMP, который тем не менее остается наиболее распространенным до сих пор. Пароль-идентификатор доступа, именуемый *community string (имя группы доступа)*, передается в пакете открытым текстом. Более того, по умолчанию в подавляющем числе устройств он имеет вид строки 'public', т. е. фиксирован и общеизвестен, правда, доступ по нему открыт обычно только на чтение конфигурации.

Соответственно, злоумышленник, получив доступ к устройству, может считать всю техническую информацию о нем и о подключенных к нему сетях. А кроме того, если администратор включил еще и возможность управления устройством по SNMP, не сменив community, — вывести устройство из работы либо переконфигурировать.

Рейтинг SANS-20

Часть 1: методологические ошибки



1. Сохранение конфигураций и настроек программ и систем, оставленных по умолчанию после инсталляции. Подобные настройки предназначены для обеспечения максимального удобства пользователей и для минимального труда администратора, однако при этом не всегда соответствуют требованиям безопасности.

2. Использование пользователями счетов (бюджетов) с пустыми или легко разгадываемыми паролями. Получив доступ к бюджету пользователя даже с минимальными привилегиями в системе, злоумышленник может использовать его как начало атаки для получения полного доступа к системе с помощью атак класса повышения привилегий (англ. *privilege escalation*).
3. Отсутствие или неполнота резервных копий. Порой причины, приводящие к потере данных, не зависят от деятельности злоумышленников, а могут быть вызваны, например, природными явлениями. В любом случае сохранение резервных копий необходимо для того, чтобы с потерей оперативных данных, организация не прекратила свое существование.
4. Большое количество открытых портов (запущенных, необязательно необходимых, сетевых служб). Поскольку каждая служба (порт) — это средство соединения пользователей (а также и злоумышленников) с системой, то и запущено их в данный момент должно быть столько, сколько необходимо для функционирования системы.
5. Отсутствие фильтрации пакетов на входе/выходе из внутренней сети организации. Минимальные требования в этом случае — проверка адресов отправителя/получателя (например, чтобы входящий в сеть пакет не имел в качестве адреса отправителя внутренний адрес сети) и других параметров (например, флага самомаршрутизации).
6. Отсутствие или неполнота регистрационных журналов. Для успешного обеспечения безопасности периодически приходится проверять — что происходило за истекший период. Основным инструментом для этого являются электронные регистрационные журналы. Отсутствие или ущербность данного инструмента не позволит выявить попытки атак и обнаружить опасные тенденции в функционировании сети.
7. Уязвимые CGI программы. Данные программы рассматривали в позиции 2 рейтинга SANS-10.

Часть 2: Windows-системы



8. Уязвимость Unicode в Microsoft Internet Information Server (IIS). Unicode — это стандарт представления алфавитно-цифровых символов вне зависимости от платформы или программы, который должен поддерживать каждый производитель программного обеспечения, в том числе и Microsoft Corp. Если клиент направляет IIS специально сформированный URL (адрес запроса) с нештатными Unicode-символами, он может перемещаться по директориям сервера за пределами разрешенной для него области (англ. *directory traversal attack*) и выполнять определенные скрипты.

Примечание

Возможно, на момент выхода книги эта уязвимость заметно переместится к началу списка.

9. Переполнение буфера в расширении ISAPI. Вместе с установкой на сервер программного обеспечения Microsoft Internet Information Server (IIS) автоматически устанавливаются прикладные программные интерфейсы Internet Services Application Programming Interface (ISAPI), обеспечивающие разработчикам возможность расширения функциональности Веб-сервера с помощью динамически подключаемых библиотек — Dynamic Linked Library (DLL). Некоторые стандартные dll, такие как idq.dll содержат ошибки, например, отсутствие контроля длины входящей строки данных. Производя атаку на переполнение буфера, злоумышленник может получить полный контроль над Веб-сервером.
10. Microsoft Internet Information Server (IIS) Remote Data Services. Уязвимость описана в позиции 4 рейтинга SANS-10.
11. Пользовательское разделение (sharing) файлов по NetBIOS. Ошибки при разделении файлов пользователями описаны в позиции 7 рейтинга SANS-10. Специфика работы протокола Server Message Block (SMB) или Common Internet File System (CIFS), обеспечивающего такое разделение, позволяет злоумышленнику получить доступ не только к ресурсам, определенным пользователем при разделении файлов, но и к другой информации, такой, как данные по бюджетам пользователей, ключам реестра и т. д.
12. Нулевая сессия или системное разделение. Для ряда системных必要ностей в операционной системе оставлена возможность подключения к компьютеру для пользователя SYSTEM без аутентификации:

```
net use \\a.b.c.d\ipc$ "" /user:""
```

где a.b.c.d — это IP-адрес компьютера (может быть использовано также и NetBIOS имя компьютера). Данное разделение может быть отключено на тех компьютерах, где явно не используется.
13. Слабое хэширование в базе данных *Security Accounts Manager* — SAM (Менеджер учетных записей) — т. н. LANManager (LM) хэширование. С технологией хэширования и его применением для защиты паролей пользователей можно ознакомиться в *главе 19*. Для обеспечения обратной совместимости с предыдущими версиями Windows (сетевая работа которых обеспечивается компонентом LANManager) операционная система сохраняет пароли в базе данных SAM в устаревшем и уязвимом формате LM. Это позволяет злоумышленнику, получившему доступ к SAM, достаточно легко восстановить пароли из их хэш-значений. Если в сети нет компьютеров, сетевая работа которых обеспечивается LM, опция поддержки обратной совместимости должна быть отключена.

Часть 3: UNIX-системы



14. Уязвимость для атаки на переполнение буфера в сервисе Remote Procedure Call (RPC) описана в позиции 3 рейтинга SANS-10.
15. Уязвимость почтовой программы Sendmail описана в позиции 5 рейтинга SANS-10.
16. Уязвимость сервера имен Berkeley Internet Name Domain (BIND) описана в позиции 1 рейтинга SANS-10.
17. R-команды. Для администрирования распределенных Unix-серверов иногда используются команды удаленного администрирования rlogin (remote login) удаленный вход, rsh (remote shell) удаленное исполнение команд, rcp (remote copy) удаленное копирование — имеющих общее название R-команды. Их недостаток в том, что удаленная машина, поддерживающая исполнение этих команд, производит аутентификацию источника команд только по IP-адресу, который может быть подменен. Использование этих команд должно быть сокращено до минимума или устранено.
18. Уязвимость сервиса LPD (Local Printer Daemon). Сервис in.lpd обеспечивает взаимодействие удаленных пользователей с локальным принтером по TCP порту 515. Модуль, отвечающий за передачу заданий на печать, имеет ошибку. Если сервис получает слишком много заданий, он может выйти из строя или исполнить посторонний код с повышенными привилегиями.
19. Уязвимость сервисов sadmind и mountd описана в позиции 6 рейтинга SANS-10.
20. Уязвимость протокола SNMP в части "community string" описана в позиции 10 рейтинга SANS-10.

Примечание

Уязвимость данного протокола присуща не только Unix-системам.

Использование списков уязвимостей в управлении рисками



Как видно из перечисленных описаний атак, часть из них связана с особенностями функционирования систем, часть с ошибками в реализации или настройке систем, часть с ошибками при использовании.

Для оценки опасности конкретной атаки предлагается следующая формула [ISA]:

$$S = (C + L) - (SC + NC),$$

где:

- *S* — severity, серьезность или опасность ущерба от атаки;
- *C* — criticality, критичность объекта атаки;
- *L* — lethality, опасность цели атаки, т. е. потенциальный размер ущерба;
- *SC* — system countermeasures, системные контрмеры объекта;
- *NC* — network countermeasures, сетевые контрмеры информационного пространства, где находится объект.

Единицы измерения можно определить самостоятельно от 1 до 10 или от 1 до 100, на основе собственных классификаций активов и имеющихся данных по атакам и средствам безопасности.

Поскольку данная часть книги посвящена управлению рисками как средству построения эффективной защиты, попробуем оценить возможность противостоять таким атакам с точки зрения экономической эффективности.

На первый взгляд, все достаточно просто — поручить специалисту, собственному или приглашенному, пройти по данному (или более расширенному) списку и обеспечить настройку, конфигурацию систем, разработку инструкций и рекомендаций, обучение пользователей и пр. — и работа будет выполнена, т. е. информационное пространство защищено. К сожалению, против такого подхода существует ряд возражений.

- Во-первых, ряд мероприятий по информационной безопасности носит не разовый, а регулярный характер. Наиболее яркий пример — регулярное обновление антивирусных баз.
- Во-вторых, практически нет информационной сети, которая была бы полностью статична в своих настройках. Периодически приходится обновлять аппаратное и программное обеспечение, а значит, необходимо изменять конфигурацию либо создавать новую — для новых систем.
- В-третьих, не в любом информационном пространстве возможно полностью запретить использование уязвимых систем и сервисов, порой требования производственных задач конфликтуют с требованиями безопасности. Наиболее яркие примеры — использование разделяемых файлов и каталогов (sharing) на пользовательском уровне, использование RPC или SNMP на системном уровне. Это означает, что необходимо разработать модель для наиболее защищенной работы в этих условиях, возможно, с привлечением дополнительных средств и механизмов безопасности.

Это только наиболее общие причины, по которым информационная безопасность потребует более серьезных мер (в том числе и финансовых), чем это представляется на первый взгляд.

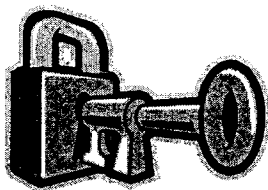
Вопрос о том, кто будет заниматься управлением рисков, в том числе и по отношению к атакам, инициированным злоумышленником, относится к

организации и определению функций службы информационной безопасности предприятия, о которых говорилось в *главе 3*. Здесь хотелось бы подчеркнуть только два важных момента.

Во-первых, при проектировании моделей и схем защиты, следует обращать внимание на то, чтобы вся полнота власти не концентрировалась в руках одного человека, каким бы доверенным он не был. Необходимо оптимально распределять контроль за информационной безопасностью между автоматизированными средствами и людскими ресурсами.

Во-вторых, несмотря на то что в сфере оперативного распознавания и предотвращения атак автоматизированные средства, без сомнения, значительно превосходят человеческие возможности, тем не менее наличие команды системных аналитиков или хотя бы одного высококвалифицированного специалиста, который периодически или регулярно производит анализ функционирования информационного пространства, — это фактор, который может сыграть ключевую роль. Его также необходимо учитывать при управлении рисками.

В качестве иллюстрации по  хотелось бы привести пример высокопрофессиональной и талантливой, на наш взгляд, работы специалистов по информационной безопасности, приведенный в книге [ISA1]. Хотя в сети не было, на первый взгляд, необычного трафика, специалистами была обнаружена троянская программа, работающая с TCP-портом 23. Поскольку этот порт предназначен стандартно для работы службы Telnet (которая имела в сети), трафик троянской программы не был бы обнаружен автоматикой, если бы специалисты не обратили внимание на размер пакетов, который был значительно больше, чем стандартные Telnet пакеты. Таким образом, профессиональные знания специалистами работы сетевых протоколов помогли обнаружить атаку, которая скорее всего не была бы распознана автоматизированными средствами в ближайшее к инциденту время.



Часть V

Криптография

Глава 16. Основные принципы криптографии

Глава 17. Симметричная криптография

**Глава 18. Асимметричная криптография
и цифровая подпись**

Глава 19. Прочие криптографические алгоритмы

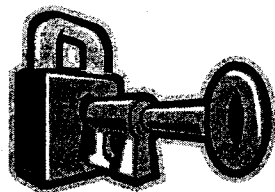
Глава 20. Сложные криптографические протоколы

В этой части авторы хотели бы рассказать о криптографии — отдельном большом направлении. С одной стороны, это направление входит составной частью в информационную безопасность и без знания криптографии невозможно представить полноценного специалиста в данной области. С другой стороны, это направление настолько огромно, что специалист, профессионально знающий только криптографию, уже ценен сам по себе. По этому принципу построена и данная часть — в ней можно найти и общие сведения, и более подробную информацию, и детальное описание алгоритмов и схем.

Криптография — наука о методах и средствах преобразования информации в вид, затрудняющий или делающий невозможным несанкционированные операции с ней (чтение и/или модификацию), вместе с методами и средствами создания, хранения и распространения *ключей* — специальных информационных объектов, реализующих эти санкции.

Криптография является самым мощным на сегодняшний день средством защиты информации. Она зародилась почти одновременно с самим искусством письма и на сегодняшний день превратилась в мощную прикладную науку, основанную на глубоких математических знаниях. Против собственно самой криптографической защиты не помогают ни различные уловки, ни использование самых мощных электронно-вычислительных машин (ЭВМ) мира — здесь на страже тайн стоит математика. Единственной возможностью для злоумышленника являются обходные пути, такие как кража действующих и использованных ключей, предсказание будущих значений генераторов псевдослучайных чисел и т. п. Вот почему наряду с собственно теорией шифрования необходимо уделять большое внимание разработке криптосистем, надежных в комплексе своих свойств. И первому и второму посвящена данная часть нашей книги.

Глава 16



Основные принципы криптографии

Немного истории



Конечно же, история криптографии, как впрочем и нынешнее ее применение, неразрывно связана с военными и дипломатическими делами. Первым документально зафиксированным в письменности шифром является *шифр Цезаря*, хотя существуют свидетельства о наличии простейших устройств шифрования и у древних греков в V—VI веках до нашей эры. Историки относят к шифрованным даже некоторые письма еще более древних цивилизаций — Месопотамии и Египта, хотя в то время сам факт письменного изложения уже являлся барьером для большей части населения. В те времена использование своего, несоответствующего общепринятому, идеографического или слогового письма было почти 100% гарантией сохранности сообщения в секрете.

Шифр Цезаря (I в. до н.э.), по свидетельству летописцев применявшийся в переписке с подотчетными ему правителями покоренных земель и в военных походах, состоял из замены одних букв на соответствующие им другие во всем тексте (рис. 16.1). Закон замены был очень прост: при шифровании сообщения вместо пятой буквы латинского алфавита ставилась вторая, вместо четвертой — первая, вместо третьей — двадцать шестая и т. д. При дешифровании, естественно, сдвиг уже нужно было делать в прямом направлении: вместо первой буквы брать четвертую и т. д.



Рис. 16.1. Шифр Цезаря

Конечно же, Юлий Цезарь не был автором этой идеи, а почерпнул ее из более ранних источников. Однако с его легкой руки шифр побуквенной замены прижился в Древнем Риме, а с различными модификациями спустя годы и века распространился на всю Европу, контролируемую тогда римлянами. В современной терминологии шифр Цезаря и его модификации, отличающиеся только законом выбора соответствий между буквами открытого и секретного текстов, относятся к *одноалфавитным подстановкам* или *одноалфавитным заменам*. Пожалуй, ни один из шифров за историю человечества не имел столько вариантов, областей применения и связанных с ним судеб, как этот. Его даже увековечили произведения, вошедшие в классику литературного жанра, — "Пляшущие человечки" Артура Конан Дойла, "Золотой жук" Эдгара По.

Необходимо отметить, что и два слабых места этого шифра, использованные героями этих книг и известные теперь, пожалуй, каждому школьнику, были обнаружены довольно давно. Этот момент, к сожалению, скрытый глубоко в истории, несомненно, является зарождением *криптоанализа* — науки, имеющей противоположные цели с криптографией и занявшей соответствующее место среди других в современном мире.

Первой уязвимостью одноалфавитных замен, замеченной нашими предками, была неизменность частот появления определенных букв в среднестатистическом тексте, т. е., если буква "О" является бесспорным лидером по частоте появления в русском языке, то с очень большой вероятностью буква, появляющаяся в зашифрованном тексте чаще всего, — скажем "Ф" — и заменяет в нем "О", далее аналогично по убыванию частот. Зависимость частот появления распространенных букв можно довольно отчетливо проследить уже, скажем, в 100-буквенном послании. Редкие буквы обычно не отслеживают подобным способом — их просто подбирают по смыслу в тех словах, где большая часть распространенных букв уже раскрыта. Данный способ получил название *метод частотного анализа*, позднее он применялся по той же самой схеме для пар рядом стоящих букв — *биграмм*, и даже для буквенных троек — *триграмм*. Конечно, в этом случае для анализа требовался большой объем шифрованного текста, однако сам метод зачастую оказывался намного быстрее и эффективнее. Средние частоты появления букв в современном английском и русском языке приведены в *приложении 3*.

Вторая уязвимость схем, подобных шифру Цезаря, связана с короткими часто встречающимися словами — предложениями, союзами и местоимениями. Их роль в тексте часто довольно важна, чтобы допускать их пропуск, а вот человеку, пытающемуся разгадать шифр, они дают очень многое. Например, не каждая буква алфавита может встречаться в тексте в полном одиночестве. А чего стоит специфичный русский союз "или"! К слову сказать, очень простое решение этой проблемы было известно уже грекам и римлянам — в шифрованном сообщении полностью удалялись пробелы, а иногда получившаяся строка

еще и разбивалась на "неправильные" слова случайным образом. Но еще много веков спустя подобные шифры ломались из-за того, что шифровальщики-самоучки не учитывали специфику языка исходного сообщения.

Значительной модификацией одноалфавитной замены является шифр, именуемый *квадратом Полибия* или *тюремной азбукой*. В нем определенные символы алфавита заменялись уже парой чисел согласно какому-либо простому правилу. Так, исходный квадрат Полибия имел, по всей видимости, следующий вид, представленный на рис. 16.2.

Вместо буквы в письме записывалась пара чисел: номер строки и номер столбца ячейки, в котором она располагалась. Также как и у шифра Цезаря, у данной схемы не счесть модификаций. В связи с тем, что в латинском алфавите 26 букв (а в современном русском — 33), и оба эти числа очень "неудачно" раскладываются на сомножители, модификации либо выбрасывали редко встречающиеся буквы, либо добавляли знаки препинания, цифры и пустые элементы, добиваясь прямоугольников и квадратов: 5×5 , 5×6 , 4×7 , 6×6 , 5×7 .

A	B	C	D	E
F	G	H	I,J	K
L	M	N	O	P
Q	R	S	T	U
V	W	X	Y	Z

Рис. 16.2. Квадрат Полибия

Подобное представление двумя небольшими числами было очень удобно для передачи на большое расстояние визуально (костры, факелы, различные предметы, сигнальные флаги, солнечные зайчики) либо звуками (выстрелы, громкие удары, перестукивания).

После значительного затишья, возможно, связанного с отсутствием большого числа письменных источников того времени, история криптографии, буквально "взорвалась" в средние века. В XVI веке Джероламо Кардано, итальянский математик, врач и философ, изобрел совершенно новый тип шифра, основанный на очень простой и в то же время надежной перестановке букв послания.

Для шифрования Кардано предложил использовать квадрат с прорезанными в нем несколькими ячейками (рис. 16.3). Ячейки прорезались таким образом, чтобы при повороте квадрата вокруг своего центра на 90° , потом на 180° , а затем на 270° в прорезах поочередно появлялись все позиции исходного квадрата и, причем только по одному разу. При шифровании квадрат

накладывался на листок для послания сначала в исходном положении — выписывалась слева направо сверху вниз первая порция (четверть) послания. Затем квадрат поворачивался на 90° , скажем, по часовой стрелке — выписывалась вторая четверть сообщения и т. д. (рис. 16.4).

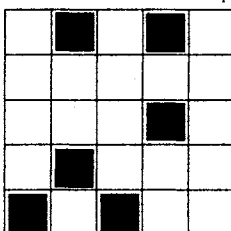


Рис. 16.3. Квадрат Кардано

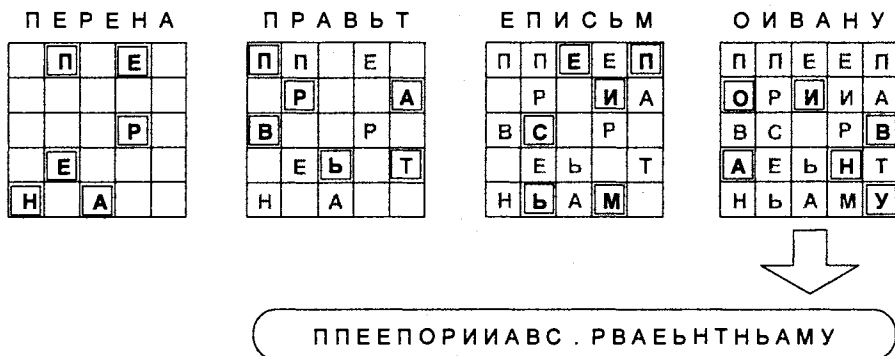


Рис. 16.4. Шифрование квадратом Кардано

Для дешифрования необходимо было иметь точную копию того квадрата, которым пользовался шифровальщик, и повторять с ней те же самые повороты. Устройство было удобно для хранения, предельно просто в обращении и при этом давало очень неплохую стойкость шифра — не зная, как именно расположены прорезы на квадрате размером $N \times N$, человеку, пере-

хватившему послание, теоретически нужно было перебрать $4^{\frac{N^2}{2}}$ вариантов. Например, для квадрата 6×6 это число уже равняется 262 144, т. е. несет 18 бит информации. А если бы не существовало способа по части известной о решетке информации восстанавливать поэтапно расположение остальных прорезей, учитывая особенности фонетики национального языка, то дешифрование квадрата 10×10 еще недавно было на грани возможностей современных ЭВМ. Ведь информация о прорезях такого квадрата составляет уже около 50 бит. Подобные шифры, не модифицирующие буквы сообщения, а только меняющие их расположение, называются *перестановочными*.

Примерно в одно время с Кардано французский дипломат Блез де Вижнер предлагает модификацию шифра замен, получившую название *таблица Вижнера*. Для шифрования по этой схеме необходима таблица из 26 латинских алфавитов, в каждом из которых по какому-либо (не обязательно одинаковому) закону изменен порядок букв, и секретное слово — *пароль*, известный только отправителю и получателю. Сам де Вижнер пользовался таблицей, приведенной на рис. 16.5.

Пароль с повторениями выписывался буква к букве под исходным сообщением. Каждая буква зашифрованного текста бралась из таблицы на пересечении строки, соответствовавшей букве исходного текста, и столбца, соответствовавшего текущей букве пароля. Подобный шифр получил название *многоалфавитной замены*. Очевидно, что он полностью лишен проблемы коротких слов и значительно более устойчив к частотному анализу. Для проведения настолько же успешного подбора букв необходим более чем в L раз больший объем шифрованного текста (где L — длина пароля). Это вызвано тем, что таблицы частот вычисляются отдельно для каждой из L групп букв, а, кроме того, и сама процедура подбора несколько усложняется.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Рис. 16.5. Таблица Вижнера

Кроме непосредственного увеличения стойкости шифр Виженера привнес в криптографию две качественно новых идеи. Во-первых, процесс шифрования стал зависеть в первую очередь от небольшого неизвестного третьей стороне слова — *пароля*. Конечно, сокрытие от злоумышленников всей таблицы значительно усложняет процедуру взлома шифра, но теперь случайное ее раскрытие не несет такого критического для всей системы значения, как, например, в шифре Цезаря.

Во-вторых, сама таблица, использованная Виженером, несет по своей сути первые намеки на идею цифрового шифрования. При том, если все буквы латинского алфавита пронумеровать по порядку от 0 до 25, то процедура шифрования по такой таблице превратится в обычную операцию сложения. Например, буква исходного текста Б (код 1) + буква пароля В (код 2) = код 3 — буква Г. При дешифровании, наоборот, из кода каждой зашифрованной буквы вычитается код буквы пароля. Если при сложении получается число, большее 25, то из него вычитается 26, если при вычитании получается отрицательное число — к нему прибавляется 26. Подобная схема сложения/вычитания называется в математике *сложением по модулю* — в данном случае сложением по модулю 26.

Дальнейшей модификацией данного способа, применяемой на практике, пожалуй, до наших дней стало *шифрование по книге*. В этом методе отправитель и получатель сообщения договариваются об одинаковой книге одного и того же издания. При шифровании отправитель выбирает в ней произвольное место и выписывает вместо повторяющегося пароля под исходным сообщением во всю его длину текст из книги, начиная с этого места. Само кодирование ведется по той же самой схеме, что и в схеме Виженера. Где-либо в письме или по другому каналу получателю передается номер страницы и слова в книге, с которого началось шифрование — без этой информации декодирование будет довольно проблематичным, даже зная все о самом издании.

С наступлением XIX века и шквалом изобретений в области электричества, а затем и радио криптография перешла на качественно новый уровень — началась *эра цифровой криптографии*. Представление символов сначала в виде чисел из некоторого диапазона, а затем и в виде двоичных сигналов (код Морзе: "точка/тире", "единица/ноль": "сигнал/пауза") открыло огромные возможности для шифрования.

Революционным шагом в криптографии стало предложение сотрудника телефонной компании Г. Вернама в 20-х годах XX века автоматизировать шифрование телетайпных посланий по следующей схеме. Информация на телетайпной ленте представляет собой последовательность отверстий и непробитых участков, соответствующих нулям и единицам. Вернам предложил на передающей стороне разместить замкнутую в кольцо ленту с довольно длинной последовательностью подобных же нулей и единиц — она играла роль пароля. В момент передачи очередного импульса в канал связи лента-пароль сдвигалась на одну позицию и с нее считывалось текущее двоичное значение.

Если был считан "0", то исходный двоичный сигнал не изменялся, если на ленте-пароле в этом месте оказывалась "1", то исходный сигнал инвертировался. На приемной стороне должна была присутствовать точно такая же лента и, причем в том же начальном положении, как у отправителя. Процедура дешифрования точно соответствовала процедуре шифрования.

Помимо того, что шифровальный аппарат Вернама был, по-видимому, первым полностью автоматическим электрическим шифратором, данная схема содержала две очень интересные идеи. Во-первых, в ней была применена операция *сложения по модулю 2*, ставшая базовой в цифровой криптографии. Заметим, что если примененную Вернамом операцию изменения одного бита в зависимости от значения другого представить как сложение двух двоичных чисел с отбрасыванием старшего бита, то правило выполнится для всех четырех возможных комбинаций: $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$, $1 \oplus 1 = 0$. Подобная операция имеет еще одно название — *исключающее ИЛИ* — и записывается двумя обозначениями: " $\oplus$ " и XOR — английским сокращением от *исключающее ИЛИ* (англ. *eXclusive OR*). В отношении сложения по модулю 2 интересен еще один факт — обратной к нему, т. е. восстанавливающей, операцией является кроме вычитания по модулю 2 еще и сама эта операция, т. е. для любых X и Y выполняется $(X \oplus Y) \oplus Y = X$ — это очень удобное свойство нередко используется в различных современных шифрах и схемах.

Вторым интересным моментом оказалось то, что если ленту, склеенную у Вернама в кольцо, сделать потенциально бесконечно длинной, а на самом деле — равной длине сообщения, то данный шифр станет *абсолютно стойким*. Не имея информации о ленте-пароле, третья сторона не сможет сделать никаких предположений о содержании исходного сообщения. Ведь сам факт взлома шифра может быть установлен только в том случае, когда дешифровщик получит из перехваченного послания какое-либо осмысленное сообщение. А если пароль по длине равен самому сообщению, то существует огромное множество потенциальных паролей, которые при наложении на шифровку будут давать осмысленный текст — и все время разный. Из одного и того же послания, подбирая разные пароли, можно прочесть и "Я_Вас_люблю" и "Уходи_прочь".

Недостатком такой системы является неудобство хранения большого пароля, а главное — одноразовость его использования. Ведь получается, что для каждого послания необходим свой пароль, а значит, отправителю и получателю нужно предварительно им обменяться. Решение состоит в *заблаговременном обмене* большим объемом ключей и надежном хранении их "до поры до случая". Естественно, в современном мире роль ленты-пароля играет уже двоичная информация на цифровом носителе, а само преобразование выполняется вычислительной техникой. Но *одноразовый блокнот*, как теперь называют данную схему, все еще имеет очень широкое применение именно из-за своей 100%-й стойкости к взлому при перехвате шифровок третьей стороной.

Классификация шифров

Классификация по ключевой информации

Первым принципиальным признаком, позволяющим произвести разделение шифров, является объем информации, неизвестной третьей стороне. В том случае, когда злоумышленнику полностью не известен алгоритм выполненного над сообщением преобразования, шифр называется *тайнописью*. Многие специалисты сегодня не относят тайнопись к криптографии вообще. По сути дела тайнопись является обычным *кодированием* информации, т. е. представлением ее в ином виде. При этом третьей стороне не известен сам принцип кодирования.

В конце XIX века голландец А. Кергофф (*Auguste Kerkhoff*) в своей книге, посвященной применению шифров, упомянул, как одну из настоятельных рекомендаций, идею о том, что при раскрытии самого алгоритма преобразований третья сторона не должна ни на шаг приблизиться к сокрытому тексту. Это должно было достигаться базированием всего алгоритма шифрования на небольшом объеме секретной информации. Данная идея легла в основу всей современной криптографии и получила название *принцип Кергоффа*. В противовес тайнописи *криптографией с ключом* называют сегодня алгоритмы шифрования, в которых сам алгоритм преобразований широко известен и доступен для исследований каждому желающему, но шифрование производится на основе небольшого объема информации — *ключа*, известного только отправителю и получателю сообщения. В современной криптографии в зависимости от методик размер ключа составляет от 56 до 4096 бит. Естественно, любой криптоалгоритм с ключом можно превратить в тайнопись, просто "зашив" в исходном коде программы некоторый фиксированный ключ и считая закрытым весь алгоритм. А вот обратное преобразование — попытка выделить из тайнописи некий небольшой аргумент-параметр — удастся далеко не всегда.

В некоторых (наиболее простых) случаях ключ формирует человек, отправляющий сообщение, в отдельных — ключ создается автоматически с помощью программного обеспечения или даже запрашивается у удаленной базы данных ключей. Подробнее сложные схемы создания ключей будут описаны в *главе 20*, а вот на методе ввода ключевой информации пользователем мы немного остановимся. Дело в том, что запоминать ключ, который является на самом деле просто большой последовательностью чисел, обычному человеку довольно сложно хоть в двоичной, хоть в десятичной записи. Скажем, символьный эквивалент ключа в 192 бита будет представлять набор более чем 55 цифр. Криптосистема, которая требует подобных действий от своего пользователя, попросту вынуждает его записывать ключ где-нибудь на листочке и хранить в портмоне, кармане или ящике стола, что сводит на нет все дальнейшие усилия по защите переписки. Поэтому все современные

системы предлагают пользователю вводить не ключ — набор цифр, а пароль — произвольную текстовую фразу. Пароль может состоять из одного или нескольких слов, быть осмысленным или нет, главное — чтобы он легко запоминался пользователем. Подробнее процедура превращения пароля в ключ будет рассмотрена в *главе 20*.

Симметричное / асимметричное шифрование

Все криптоалгоритмы с ключом делятся на *симметричные* и *асимметричные*. В *симметричных криптоалгоритмах* ключи, используемые на передающей и приемной сторонах, полностью идентичны. Такой ключ несет в себе всю информацию о засекречивании сообщения и поэтому не должен быть известен никому, кроме двух участвующих в разговоре сторон. Поэтому в отношении ключа симметричных систем часто применяется термин *секретный ключ*, а сами подобные системы называются *шифрами на секретном ключе*. Общая схема процесса передачи сообщения приведена на рис. 16.6.

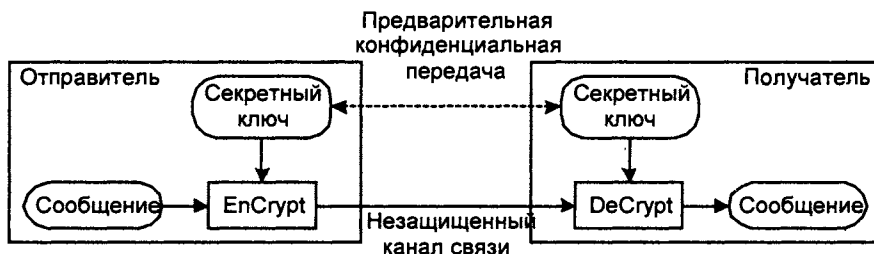


Рис. 16.6. Шифрование на секретном ключе

Симметричное шифрование можно применять как при отправке сообщений между двумя пользователями, разделенными большим расстоянием, так и при отправке "посланий" одним и тем же человеком самому себе, но во времени. Примером подобных отправок является шифрование файлов на жестких дисках и сменных носителях с тем, чтобы другие пользователи тех же ЭВМ не могли считать информацию в отсутствие владельца. Симметричным криптоалгоритмам посвящена *глава 17*.

В *асимметричном шифровании*, как это не покажется странным на первый взгляд, для шифрования сообщения применяется один ключ, а для дешифрования — другой (рис. 16.7). Почему это необходимо? Дело в том, что процедура шифрования в асимметричных системах устроена таким образом, что ни одно постороннее лицо не может, зная зашифрованный таким способом текст и ключ шифрования, восстановить исходный текст. Прочитать зашифрованный текст можно, только зная ключ дешифрования. А раз так, то ключ шифрования может быть известен всем пользователям сети — его раскрытие не несет никакого урона переписке. Поэтому ключ шифрования в

в асимметричных системах называется *открытым ключом*. А вот ключ дешифрования необходимо уже держать в строгом секрете, как и секретный ключ симметричных систем. Поэтому он носит название *закрытого ключа*, а сами асимметричные системы получили еще одно обозначение — *шифры на открытом ключе*.

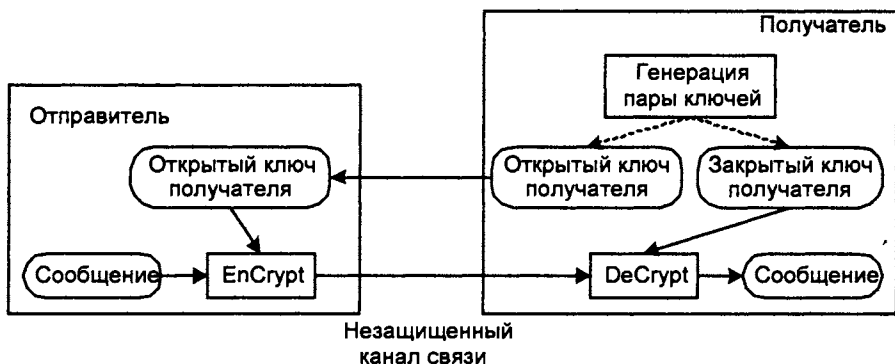


Рис. 16.7. Шифрование на открытом ключе

Асимметричные шифры не имеет смысла использовать для защищенного хранения документов. Их прерогатива — защита сообщений, электронной почты, прикрепленных к посланиям файлов и т. п. Эти алгоритмы будут подробно рассмотрены в *главе 18*.

Поточное / блочное шифрование

Следующим критерием классификации шифров является схема обработки ими потока информации. Согласно ему, симметричные криптоалгоритмы делятся на *поточные* и *блочные* шифры. *Поточный шифр* способен обрабатывать информацию побитно, т. е. подобная схема может, получив порцию из произвольного количества бит (может быть, даже одного), зашифровать/дешифровать ее и передать для дальнейшей обработки другим модулям. Подобная схема очень удобна в каналах последовательной связи, где сам процесс передачи информации может обрываться в произвольный момент и затем через некоторый промежуток времени продолжаться дальше.

Однако побитовая обработка информации является очень медленной в тех случаях, когда вычислительная техника имеет возможности для параллельной обработки (то есть в программной реализации). Разрядность основной массы современных процессоров равна 32 битам, существуют 64- и 80-разрядные аппаратные платформы. В этих условиях, особенно тогда, когда информация все равно переносится в буфер в том или ином объеме (пакеты в компьютерных сетях, файлы на носителях), гораздо выгоднее применять совершенно другие принципы криптографических преобразований, называемые *блочными шифрами*.

Основным законом блочного шифрования является "или блок, или ничего". То есть преобразования могут применяться только над информацией строго определенного объема. Размер блока на сегодняшний день равен 64, 128 или 256 битам. Частичное шифрование (скажем, попытка обработать 177 бит) невозможно. Блочное шифрование получило гораздо более широкое распространение из-за развития современной электронно-вычислительной техники (ЭВТ), и, если поточные шифры одинаково часто реализуются как программно, так и аппаратно, то блочные шифры в подавляющем большинстве имеют программную реализацию.

Асимметричные криптоалгоритмы были изначально разработаны как блочные. Теоретически возможно создание поточных асимметричных систем, но практическую ценность подобные схемы вряд ли имеют. Целью шифрования на открытом ключе является конфиденциальность некоего определенного послания, поэтому этап накопления в буфере (а именно — в программах пересылки сообщений) и блочность структуры тут налицо.

Общая схема классификаций шифров приведена на рис. 16.8.

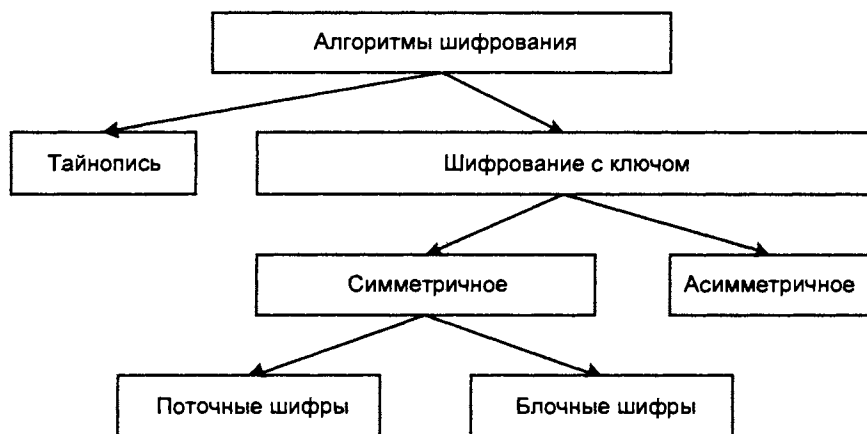


Рис. 16.8. Общая схема классификации шифров

Практическая стойкость шифров

Как уже было указано, *абсолютная стойкость шифра* достигается только в том случае, когда размер ключа равняется или превышает размер исходного текста. В этом случае, какой бы большой объем зашифрованного текста не перехватил злоумышленник, для него останется неизвестным, какой именно исходный текст был зашифрован, если он не владеет ключом шифрования.

Если же размер ключа меньше, чем объем исходного текста, то подобной непредсказуемости не остается из-за того, что любой национальный язык избыточен по своей структуре. Любой человек может практически сразу же сказать относительно небольшого отрывка текста, возможен (осмыслен) ли он для родного ему языка или нет. Подобные же (хотя и гораздо более примитивные, чем человеческое мышление) алгоритмы-тесты есть в распоряжении дешифровальщиков. Ключ считается подобранным, если в результате дешифрования в послании получилась часть осмысленной фразы. Таким образом, любой шифр, у которого длина ключа меньше длины передаваемого текста, не является абсолютно стойким — его можно всегда вскрыть хотя бы полным перебором всех возможных ключей до появления в результате дешифрования осмысленной фразы. Здесь можно говорить только о практической стойкости шифра.

Практически стойким называется шифр, если не существует более результативных методов атаки на него, кроме как полным перебором всех возможных ключей (а она существует всегда). Каким видам атак должен противостоять стойкий шифр?

Атака на основе шифртекста — это попытка злоумышленника выяснить либо исходный текст, либо ключ шифрования по определенному (достаточно большому) объему зашифрованных данных. Злоумышленнику почти всегда больше всего хотелось бы, конечно, произвести успешную атаку на основе шифртекста, поскольку для нее не требуется никаких дополнительных действий. Но как раз за долгую историю развития криптографии разработчики научились делать шифры особенно стойкими к данному виду атаки.

Атака на основе известного открытого текста — ситуация, когда злоумышленник знает и исходный, и получившийся в ходе преобразований над ним зашифрованный текст, и хочет узнать ключ шифрования. Дело в том, что в реальной практике злоумышленник в 99% случаев знает какую-либо дополнительную информацию об исходном тексте. Если это файл определенного формата — то известна структура его заголовка и шаблон формирования, если это письмо — предположительно известно его начало ("здравствуйте", "добрый день") и конец ("с уважением" и т. д.), если это стандартный бланк в электронном виде (например, банковский платеж) — известны практически все поля, кроме нескольких. В этом случае при успешной атаке злоумышленник на основе известных ему мест открытого текста и зашифрованного аналога получает ключ и декодирует сообщение целиком. Именно поэтому стойкость шифров к подобной атаке, т. е. *невосстановимость относительно ключа*, является столь же важной целью при их разработке, как и защита от первого типа атаки.

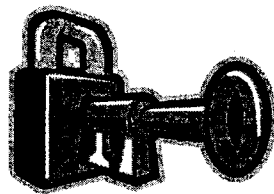
Атака на основе выбранного открытого текста является модификацией предыдущей атаки в том случае, когда злоумышленник, являясь, например, одним из равноправных, но недобросовестным пользователем системы, может навязывать службе шифрования, какую информацию необходимо передать

по криптоканалу. В этом случае он формирует какое-либо сообщение (корректное, но с необходимыми ему характеристиками) и заставляет систему шифрования произвести криптопреобразование. Результат шифрования перехватывается, на основе полученной информации делаются какие-либо вычисления или предположения и исходя из них, формируется очередное сообщение для дальнейшего выяснения ключа. Практически стойкий шифр должен выдерживать огромное количество подобных исследований, не давая злоумышленникам никакой новой информации о ключе шифрования.

На самом деле относительно любого конкретного алгоритма невозможно утверждать, что таких методов не существует. Однако, если в течение долгого периода (скажем, десяти лет) на какой-либо шифр не найдено ни одного способа взлома (а криптоаналитики всего мира постоянно занимаются исследованием и старых и современных алгоритмов), то он считается *надежным шифром*. Необходимо, правда, помнить, что в любой момент определенный шифр может покинуть семейство надежных шифров с открытием какой-либо новой математической теоремы или метода криптоанализа, и так было уже не раз. Например, после доказательства Э. Берлекампом и Дж. Месси теоремы о том, что при перехвате $2 \times N$ бит шифрующей последовательности линейного регистра сдвига всегда можно восстановить его внутреннюю структуру, линейные регистры сдвига покинули класс надежных шифров и применяются только как составные элементы более сложных криптоалгоритмов. До сих пор не существует теории, позволяющей генерировать надежные на 100% практически стойкие шифры.

Итак, в отношении надежных шифров вырисовывается следующая картина: злоумышленник не может взломать шифр кроме как перебором всех ключей, а, следовательно, чем больше их множество, тем больше времени потребуется на их полный перебор. В этом смысле размер ключа (в случае насыщенного двоичного представления — логарифм по основанию 2 от количества возможных вариантов — $\log_2 N$) является основной категорией вычислительной стойкости шифра. Самый хороший шифр с ключом длиной в 16 бит (65 536 вариантов ключа) будет взломан обычным домашним компьютером за несколько минут. На ключ длиной в 32 бита (4 млрд вариантов) ему потребуются дни или недели, но шифр все же будет взломан. Ключи в 60—80 бит могут быть взломаны только сетью мощных ЭВМ или же специализированными процессорами с параллельной обработкой, находящимися на вооружении государственных служб. Ключи длиной более 128 бит в современных надежных шифрах не взламываются полным перебором на современном уровне развития ЭВТ. Рекомендованной длиной ключей, применяемых в коммерческих целях, считается ключ 192 бита, в государственных — 256 бит. Запас сделан на несколько десятков лет столь же бурного экспоненциального роста вычислительных возможностей ЭВМ.

Глава 17



Симметричная криптография

Детальное знакомство с криптоалгоритмами начнем с огромного класса симметричных шифров. Напомним, что к этой категории относятся алгоритмы, которые для шифрования и для дешифрования используют один и тот же ключ. Симметричные шифры обладают в несколько раз более долгой историей, по сравнению с асимметричными алгоритмами. С этим же связано и огромное число исследований как собственно по симметричной криптографии, так и по криптоанализу симметричных шифров.

Поточные шифры



Общие сведения

Как уже упоминалось в *главе 16*, главной отличительной чертой поточных шифров является *побитная обработка информации*. Как следствие, шифрование и дешифрование в таких схемах может обрываться в произвольный момент времени, как только выясняется, что передаваемый поток прервался, и также восстанавливаться при обнаружении факта продолжения передачи. Подобная обработка информации может быть представлена в виде автомата, который на каждом своем такте:

- ☐ генерирует по какому-либо закону один бит шифрующей последовательности;
- ☐ каким-либо обратимым преобразованием накладывает на один бит открытого потока данный шифрующий бит, получая зашифрованный бит.

Рассмотрим подробнее данные этапы. Почему шифрующую последовательность достаточно ограничить только одним битом на бит исходного текста? Дело в том, что в арифметике по модулю 2, к которой относятся любые преобразования над битами, существуют только две обратимые операции: исключающее ИЛИ (англ. *exclusive OR* — *XOR*, оно же — сложение по модулю 2) и отрицание. *Обратимой* называется функция, у которой, зная результат и все операнды, кроме одного, можно восстановить этот неизвестный операнд. Очевидно, что в процессе шифрования потока можно применять только обратимые операции, иначе на приемной стороне полу-

чатель не сможет однозначно восстановить исходный текст по принятому сообщению, даже зная правильный ключ.

Следовательно, как бы много мы не создавали шифрующих бит на один бит исходного текста, все их придется накладывать на данный бит путем комбинации из операций XOR и отрицаний. Но отрицания можно вносить внутрь операции XOR —

для любых a и b : $NOT(a XOR b) = a XOR (NOT b) = (NOT a) XOR b$

Следовательно, как бы ни была сложна композиция из шифрующих бит и исходного бита, ее всегда можно *разделить*, т. е. представить в виде:

$$p XOR F(g_1, g_2, g_3, \dots),$$

где p — исходный бит (от англ. *plain* — открытый), g_i — шифрующие биты, F — некоторая функция, содержащая в качестве операций исключающее ИЛИ и отрицания. Очевидно, что логичнее сразу произвести это преобразование над промежуточными битами g_i и получить в результате только один шифрующий бит g . В результате вся формула шифрования примет универсальный вид: $c = p XOR g$, где c — зашифрованный бит (от англ. *ciphered* — зашифрованный).

Все современные поточные шифры действуют по данной схеме. Бит шифрования, получающийся на каждом новом шаге автомата, как впрочем и целый набор таких бит, принято обозначать символом γ (*гамма*), а сами поточные шифры получили из-за этого второе название — *шифры гаммирования*. Как можно заметить, описанный ранее шифр Вернама является классическим шифром гаммирования. Общая схема шифрования поточным шифром приведена на рис. 17.1.

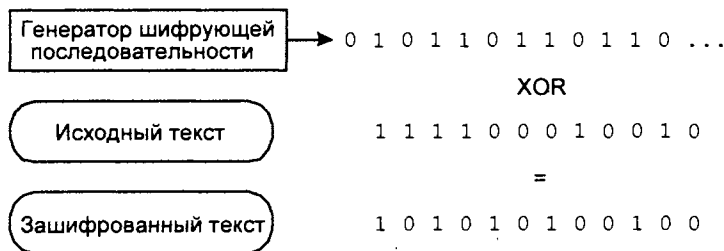


Рис. 17.1. Шифрование поточным шифром в общем виде

Шифры гаммирования намного быстрее своих ближайших конкурентов — блочных шифров — в том случае, если поточное шифрование реализуется аппаратно. Как мы увидим, базовые схемы шифров гаммирования устроены исключительно просто и как бы "сами просятся" в аппаратную реализацию — это и неудивительно, если принять во внимание историю и основную цель их создания. В тех же случаях, когда по каким-либо причинам

данные алгоритмы реализованы программно, их скорость сравнима с блочными шифрами, а иногда и гораздо ниже их.

Тремя основными компонентами, над которыми вычисляется функция, порождающая гамму, являются:

- ключ;
- номер текущего шага шифрования;
- близлежащие от текущей позиции биты исходного и/или зашифрованного текста.

Ключ является необходимой частью гаммирующего шифра. Если ключ и схема порождения гаммы не является секретным, то поточный шифр превращается в обычный преобразователь-кодер — *скремблер* (от англ. *scramble* — перемешивать, взбивать). Скремблеры широко используются в системах связи для улучшения статистических характеристик передаваемого сигнала. Частота появления единиц и нулей в обработанном скремблером потоке близка к (0,5), что дает выигрыш в качестве передачи сигнала. Кроме того, частая смена нулей и единиц необходима во многих системах синхронизации — по моменту изменения значения (*фронту*) сигнала с "0" на "1" или с "1" на "0" приемная сторона корректирует свои генераторы синхроимпульсов. Часто в отношении поточных шифров по аналогии с подобными кодерами употребляется термин "скремблер".

Зависимость шифрующего бита от номера текущей позиции, если таковая существует, чаще всего задается неявно. Не существует простой формулы определения по номеру позиции и ключу очередного бита гаммы. Просто на каждом такте шифрования над материалом ключа и внутренними переменными поточного шифра производятся какие-либо однотипные преобразования, а из-за этого каждый шифрующий бит фактически зависит от его положения (номера) в общем потоке гаммы.

При включении в подобный цикл преобразований бит исходного или зашифрованного текста поточный шифр получает совершенно новые свойства. Обычно обратная связь такого типа охватывает биты, располагающиеся не очень далеко от текущей позиции. Это вызвано тем, что увеличение подобного расстояния хотя и улучшает криптостойкость шифра к определенному типу атак, но требует реализации дополнительных ячеек памяти, что может стать обременительным при аппаратной реализации.

Матрица зависимости основных свойств поточных шифров от идеологии их построения приведена в табл. 17.1. Следует сразу оговориться, что описанные свойства зависят от многих факторов, в том числе очень сильно от конкретной структуры поточного шифра, поэтому воспринимать их как абсолютную истину нельзя.

Таблица 17.1. Свойства разновидностей поточных шифров

	Гамма зависит от бит исходного или зашифрованного текста	Гамма НЕ зависит от бит исходного или зашифрованного текста
Гамма зависит от номера текущего такта шифрования	(–) дешифратор теряет синхронизацию при ошибке "вставка/пропуск бита" в канале связи (–) дешифратор размножает ошибки "искажение бита" в канале связи (+) схема устойчива к атаке по известному исходному тексту	(–) дешифратор теряет синхронизацию при ошибке "вставка/пропуск бита" в канале связи (+) дешифратор не размножает ошибки "искажение бита" в канале связи (+) схема устойчива к атаке по известному исходному тексту
Гамма НЕ зависит от номера текущего такта шифрования	(+) дешифратор не теряет синхронизацию при ошибке класса "вставка/пропуск бита" в канале связи (–) дешифратор размножает ошибки класса "искажение бита" в канале связи (–) схема не устойчива к атаке по известному исходному тексту	

Шифры с гаммой, зависящей только от ключа и номера такта шифрования (см. табл. 17.1 правый столбец верхняя строка), получили наибольшее распространение в современной практике.

Стойкость шифров гаммирования

Атаки на шифры гаммирования, не связанные с конкретной их реализацией, основаны в большинстве своем либо на факте значительного отклонения статистических характеристик гаммы от действительно случайного потока, либо на повторном использовании некоторых частей гаммы в процессе шифрования. В первом случае отклонение характеристик гаммы не может полноценно скрыть частотные характеристики исходного текста, и появляется возможность проведения различных форм частного и корреляционного криптоанализа. Во втором случае повторное наложение по исключающему ИЛИ той же шифрующей последовательности g на новый исходный блок p_2 позволяет злоумышленнику на основе нехитрой формулы:

$$(p_1 \text{ XOR } g) \text{ XOR } (p_2 \text{ XOR } g) = p_1 \text{ XOR } p_2$$

полностью избавиться от гаммирующей последовательности. А это уже дает возможность при удачном стечении обстоятельств с помощью специальных

методов, включающих подбор букв и статистические исследования, извлечь из ($p_1 \text{ XOR } p_2$) отдельно p_1 и отдельно p_2 .

Эти методы криптоанализа позволяют сформулировать два основных принципа шифров гаммирования, не зависящие от их конкретной структуры:

- гамма, производимая шифром, должна обладать очень хорошими стохастическими характеристиками;
- период гаммы шифра должен превышать длину наибольшего из сообщений, планируемых к пересылке на одном и том же ключе.

Кроме того, нельзя забывать, что в каналах с возможностью модификации данных шифры гаммирования совершенно беззащитны перед фальсификацией. В том случае если злоумышленник знает побитное расположение данных в передаваемой последовательности, инвертирование любого бита не может быть распознано на принимающей стороне. Это приведет к инвертированию соответствующего бита в декодированном тексте сообщения. При использовании поточных шифров на каналах подобного рода необходима тщательная реализация системы подтверждения целостности сообщения.

Линейные регистры сдвига

Самыми простыми схемами, используемыми в качестве базовых при построении других, более стойких, поточных шифров, являются *линейные регистры сдвига* — ЛРС (англ. *linear feedback shift registers* — *LFSR*). Их строение предельно просто: устройство представляет из себя несколько (от 20 до 100) ячеек памяти, в каждой из которых может храниться один бит информации. Совокупность бит, находящихся в данный момент в ЛРС, называется его *состоянием*. Для выработки очередного бита шифрующей последовательности, т. е. гаммы, ЛРС производит один цикл преобразований, называемый *тактом*, по следующему алгоритму.

1. Первый (скажем, самый правый) бит из последовательности поступает на выход ЛРС — это очередной бит гаммы.
2. Содержимое всех промежуточных ячеек памяти сдвигается на одну позицию вправо.
3. В пустую ячейку памяти, появившуюся в результате сдвига у левого края ЛРС, помещается бит, который вычисляется, как операция *XOR* над значениями из ячеек ЛРС с определенными номерами.

Естественно, направление сдвига не играет никакой роли, и можно было бы сформулировать весь данный алгоритм сдвигами "справа налево".

Схематично линейный регистр сдвига выглядит, как показано на рис. 17.2.

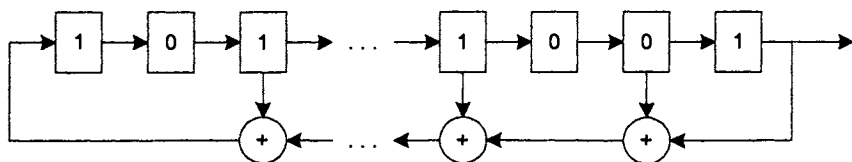


Рис. 17.2. Общий вид линейного регистра сдвига

Число бит, охваченных в ЛРС обратной связью, называется его *разрядностью*. При использовании в качестве простейшего шифра перед началом процесса в ячейки памяти ЛРС помещают побитно ключ. Как следствие, бит гаммы, порождаемый на каждом такте, зависит от ключа и от номера данного такта в общей процедуре шифрования. Пример работы ЛРС разрядности 3 с отводами обратной связи от нулевого и второго бита при установке ключа "011" представлен на рис. 17.3.

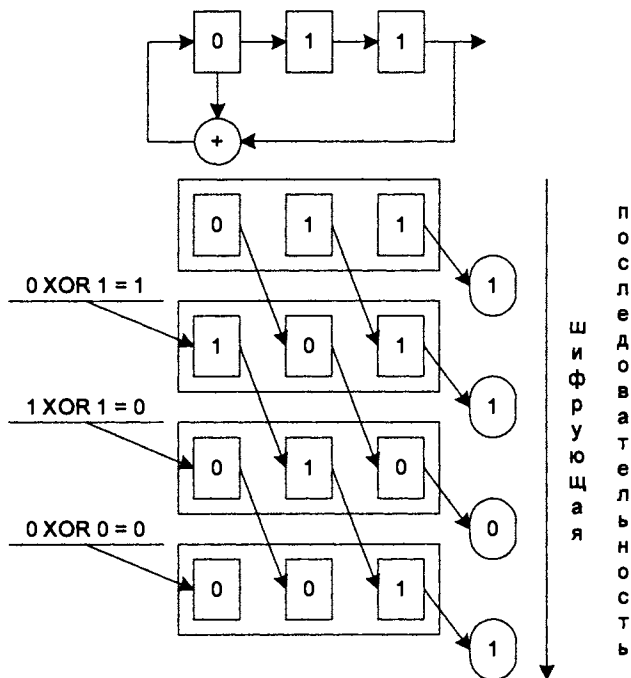


Рис. 17.3. Пример работы линейного регистра сдвига

При достаточно долгой работе скремблера неизбежно возникает его *зацикливание* — количество всевозможных вариантов состояний ЛРС конечно, а следовательно, по выполнении определенного числа тактов в его ячейках создастся комбинация бит, которая в нем однажды уже оказывалась. С этого момента кодирующая последовательность начнет циклически повторяться

с фиксированным периодом. Если представить множество состояний ЛРС и переходов между ними в виде графа, то в зависимости от номеров ячеек, порождающих обратную связь, могут получиться совершенно различные топологии. Некоторые из них приведены на рис. 17.4.

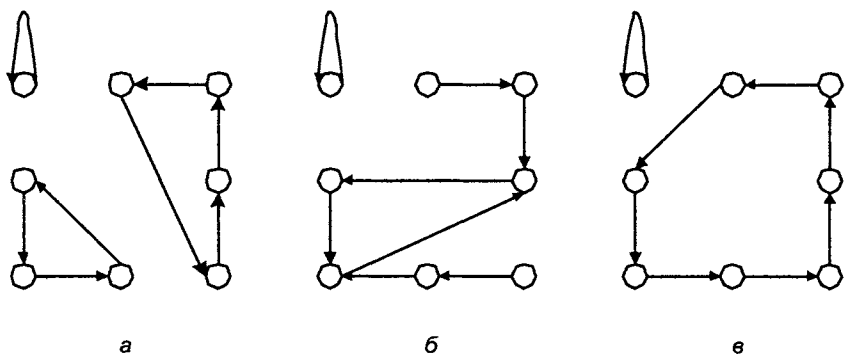


Рис. 17.4. Графы линейного регистра сдвига

Цикл "000" характерен для всех графов из-за строения ЛРС. На рис. 17.4, *а* кроме "нулевого" цикла мы видим еще два цикла — из 3-х состояний и из 4-х. На рис. 17.4, *б* цепочка сходится к циклу из 3-х состояний и уже никогда оттуда не выходит. И, наконец, на рис. 17.4, *в* все возможные состояния, кроме нулевого, объединены в один замкнутый цикл. Очевидно, что именно в этом случае, когда все $2^N - 1$ состояний системы (где N — разрядность ЛРС), кроме нулевого, образуют единый цикл, период повторения гаммы максимален, а корреляция между длиной цикла и начальным состоянием регистра (то есть ключом), которая привела бы к появлению более слабых ключей, отсутствует.

Схемы с выбранными по данному закону обратными связями называются *генераторами последовательностей наибольшей длины* (ПНД), и именно они используются в алгоритмах поточного шифрования. Таблицы номеров ячеек, от которых необходимо выбрать отводы обратной связи для получения ПНД, опубликованы во многих источниках. Например, в [Menezes] приведен список для генераторов ПНД разрядности от 2 до 229. В целом проблема порождения последовательностей наибольшей длины на ЛРС тесно связана с математическими понятиями *неприводимых* и *примитивных* полиномов над полем Галуа $GF(2)$.

Из множества генераторов ПНД заданной разрядности во времена, когда они реализовывались на электрической или минимальной электронной базе выбирались схемы с минимальным количеством отводов на обратную связь. Обычно генератора ПНД любой желаемой разрядности возможно достичь за 2, 3 или 4 отвода. Однако очень большие расстояния между отводами об-

ратной связи, которые неизбежно появляются при сочетании малого числа отводов и большой разрядности ЛРС, также могут негативно сказаться на стойкости шифрующей последовательности к криптоанализу.

Стойкость линейных регистров сдвига

Основным препятствием к использованию ЛРС в качестве шифров непосредственно является их неспособность противостоять атаке по известному открытому тексту. Во-первых, в том случае когда злоумышленнику известна структура обратных связей ЛРС разрядности N и хотя бы N последовательных бит, порожденных им на каком-либо этапе шифрования (то есть состояние регистра $(N - 1)$ тактов назад), то все последующие состояния регистра могут быть получены им самостоятельно. Таким образом, будет порождена вся следующая за этим моментом гамма, и дешифрован весь последующий текст. Более того, восстановление бит гаммы возможно и в обратном направлении из-за самого строения ЛРС. Это позволяет злоумышленнику записывать весь передаваемый шифртекст, одновременно выполняя попытки подобрать соответствующий ему открытый текст на основании набора высоковероятных для данного контекста фраз или данных. Если в какой-либо момент времени открытый текст длины N будет успешно подобран, то весь идущий затем зашифрованный поток будет успешно дешифроваться. Параллельно, возможно на другой ЭВМ, весь законспектированный до точки взлома шифртекст будет прочтен в обратном направлении.

Во-вторых, как оказалось, нестойки к атаке на основе известного открытого текста и ЛРС с секретной структурой отводов обратной связи. Алгоритм, предложенный Э. Берлекампом (*E. Berlekamp*) и Дж. Мессеи (*J.L. Massey*), позволяет на основе имеющейся непрерывной последовательности из $2 \times N$ бит построить ЛРС разрядности N , порождающий такую последовательность. Скорость выполнения алгоритма пропорциональна квадрату N . Таким образом, даже если злоумышленнику неизвестна структура ЛРС, то путем перебора сначала по начальной позиции известных слов или данных в открытом тексте, а затем по разрядности регистра сдвига (если она неизвестна) возможно построить ЛРС, зашифровавший этот текст. При этом восстанавливается как структура ЛРС, так и текущее его состояние, что позволяет производить дальнейшее дешифрование точно так же, как и в случае с известным видом обратной связи ЛРС. Разрешить две эти проблемы призваны нелинейные поточные шифры.

Нелинейные поточные шифры

На сегодняшний день из открытых к публикации разработок нелинейных поточных шифров наибольшую популярность получили три класса алгоритмов. Это *комбинирующие*, *фильтрующие* и *динамические* поточные шифры. Все они в качестве базовых элементов используют линейные регистры сдвига, причем только те, которые порождают последовательности наибольшей длины. Во всех трех методах основной целью ставится получение большого

периода порождаемой гаммы, высокой нелинейности, устойчивости к атакам по открытому тексту и корреляционным атакам.

Фильтрующие шифры имеют, пожалуй, наиболее простую структуру из нелинейных поточных шифров. Они строятся на базе одного ЛРС созданием от его ячеек дополнительных отводов, никак не связанных с отводами обратной связи. Значения, получаемые по этим отводам, смешиваются на основе какой-либо нелинейной функции — *фильтра* (рис. 17.5). Бит-результат данной функции и подается на выход схемы как очередной бит гаммы.

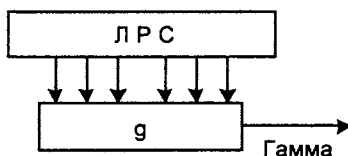


Рис. 17.5. Фильтрующий поточный шифр

Комбинирующие шифры строятся на основе нескольких ЛРС, объединяя нелинейной функцией биты, порождаемые каждым из них на очередном шаге (рис. 17.6).

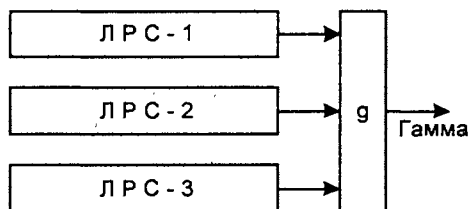


Рис. 17.6. Комбинирующий поточный шифр

Одной из основных проблем комбинирующих шифров является возможность просачивания информации из одного или нескольких ЛРС в выходную гамму. Так, в простейшем примере, приведенном на рис. 17.7, итоговая шифрующая последовательность порождается по формуле $g = a \oplus (b \& c)$. Несмотря на то что подобная гамма является сбалансированной, т. е. вероятности появления нулей и единиц в ней совпадают, в нее с неприемлемо большой вероятностью проходит поток бит, порождаемый первым ЛРС. Действительно, вероятность $P(g=b) = 4/8 = 1/2$, $P(g=c) = 4/8 = 1/2$, но $P(g=a) = 6/8$. Это позволяет при несколько большем объеме перехваченной информации, чем это требовалось ранее, проводить на первый ЛРС все те же атаки, что и на одиночный ЛРС. А получив всю необходимую информацию по первому ЛРС, можно либо непосредственно воспользоваться ею для

чтения с "домысливанием" открытого текста, либо попытаться узнать больше информации о втором и третьем ЛРС.

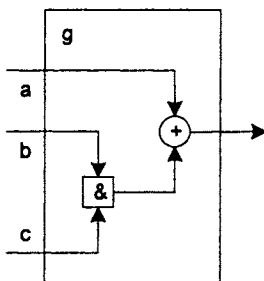


Рис. 17.7. Схема $g = a \oplus (b \& c)$

Описанный метод называется *корреляционной атакой* на комбинирующий шифр. Комбинирующий шифр называется *корреляционно-стойким*, если для любой комбинации ЛРС, входящих в его состав, значение коэффициента корреляции между битами этих ЛРС и результирующим битом шифра равно $1/2$.

Однако после проведения математических исследований оказалось, что чем выше степень нелинейности объединяющей функции нелинейного фильтра, тем ниже его корреляционная стойкость и наоборот. Разработчики оказались между двумя криптоаналитическими атаками. Одним из методов выхода из данной ситуации оказалось использование в объединяющей функции дополнительных элементов памяти. Подобные ячейки хранят несколько бит, характеризующих предыдущее состояние функции, и "подмешивают" его к ней же на этапе вычисления (рис. 17.8).

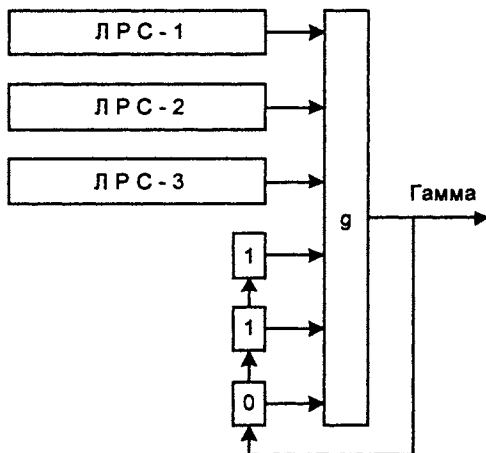


Рис. 17.8. Комбинирующий поточный шифр с элементами памяти

Динамические шифры строятся также на базе нескольких ЛРС, но объединяют их не на равноправных условиях, а по схеме "начальник—подчиненный". Так, например, в схеме, приведенной на рис. 17.9, бит, порождаемый управляющим регистром, определяет бит, какого из подчиненных ЛРС (первого или второго) будет подан на выход всего алгоритма. Бит от противоположного выбранному регистру просто отбрасывается.

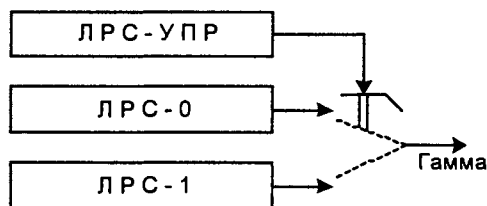


Рис. 17.9 Динамический поточный шифр с выбором ЛРС

По другой схеме, если на очередном своем шаге управляющий ЛРС генерирует "1", то бит подчиненного ЛРС, полученный на этом же шаге, поступает на выход системы, а если же управляющий ЛРС выдал "0", то бит подчиненного регистра просто отбрасывается. Разумеется, возможны и более сложные схемы взаимодействия ЛРС и их комбинации с уже описанными методами.

При правильном построении закона объединения нескольких ЛРС как в комбинирующих, так и в динамических шифрах период результирующей последовательности будет равен произведению периодов или полупериодов каждого отдельно взятого ЛРС (в зависимости от схемы их объединения). Единственным необходимым условием для этого является то, чтобы разрядности всех ЛРС были попарно взаимно просты, т. е. не имели общих делителей кроме единицы. Чаще всего разрядности всех ЛРС, входящих в общий блок, выбираются как простые числа из подходящего диапазона.

Основы блочного шифрования



Основные идеи

Отличительной чертой *блочных* шифров является обработка исходного текста по несколько (десятки, сотни) бит, т. е. поблочно. Блочные шифры являются логичным продолжением идеи алфавитных замен, перенесенной в новый цифровой век. Очевидно, что при выполнении алфавитных подстановок двухбуквенных пар (биграмм) на другие двухбуквенные пары, частотный анализ становится уже достаточно трудоемкой задачей. А в том случае, когда по какому-либо закону изменяются наборы из 4, 8 или даже 16 символов, ни о каком частотном вскрытии шифров речи идти не может —

объем шифртекста, необходимого для подобной атаки, растет экспоненциально. Проблема, из-за которой подобные схемы не были востребованы ранее, заключалась в отсутствии надежного и простого в употреблении алгоритма подобных преобразований, да еще в идеале и зависящего только от ключа шифрования.

Все изменила электроника. В современных системах обработки, передачи и хранения информации двоичное представление данных позволило достичь небывалой абстракции формы от содержания. Любой информационный поток, будь то стихи, биржевые сводки или поток телеметрии какого-либо устройства, может быть представлен в виде набора предельно унифицированных знаков — бит. А вычислительные машины, позволяющие выполнять за секунду недоступное для возможностей человека количество математических действий, позволили разработать совершенно иные, неожиданные для человеческого склада ума, законы модификации данных.

Любой блочный шифр в общем виде представляет собой закон отображения множества входных блоков исходного текста на множество блоков зашифрованного текста (рис. 17.10). Кроме того, этот закон очень сильно зависит от секретного ключа шифрования.

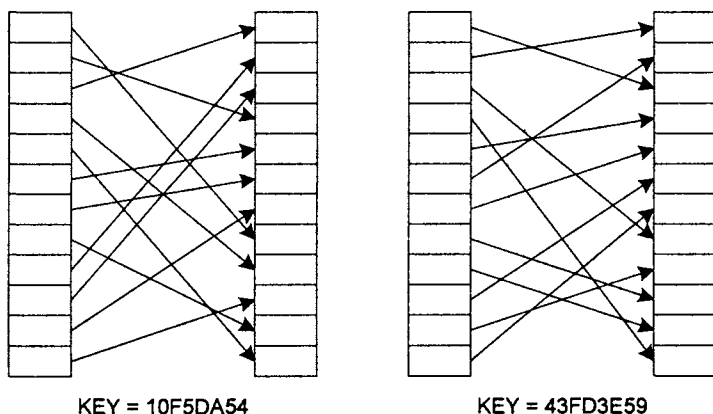


Рис. 17.10. Блочный шифр как закон отображения (на примере двух ключей)

В принципе, блочный шифр — это шифр замены над очень большим алфавитом (из десятков и сотен бит). Потенциально любой шифр замены можно описать полной таблицей соответствий между входными и выходными данными. Так, впрочем, и делали для побуквенных замен. Но для того чтобы описать только одно отображение над 64-битным блоком (то есть алфавитом из 2^{64} элементов) требуется $64 \times 2^{64} = 2^{70}$ бит запоминающего устройства, а ведь это отображение будет применено только при каком-то одном конкретном значении ключа. Поэтому современный блочный шифр — это закон, описывающий данное отображение *алгоритмически*.

Количество бит в обрабатываемом блоке называется *разрядностью блока*, количество бит в ключе — *размером ключа* алгоритма. Наиболее популярная на сегодняшний день разрядность блока — 64 и 128 бит, о размере ключа уже шел разговор в *главе 16* (это 128, 192 или 256 бит). В общем виде процесс блочного шифрования описывается формулой:

$$Z = \text{Encrypt}(X, \text{Key}),$$

где X — блок исходного текста, Z — зашифрованный блок, Key — ключ шифрования. Разрядность зашифрованного блока в классических блочных шифрах в точности совпадает с разрядностью исходного блока, поэтому за этап собственно шифрования поток данных не уменьшается и не увеличивается в размерах. Дешифрование описывается соответственно формулой

$$X = \text{Decrypt}(Z, \text{Key}).$$

В тех случаях, когда и для шифрования и для дешифрования используется одна и та же последовательность действий, т. е. $\text{Encrypt} \equiv \text{Decrypt} \equiv \text{Crypt}$ и, соответственно, $\text{Crypt}(\text{Crypt}(X, \text{Key}), \text{Key}) = X$ блочный шифр называется *абсолютно симметричным*. Подобные шифры обладают преимуществами и недостатками, а их построение на самом деле не такая сложная задача, как может показаться на первый взгляд.

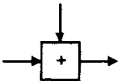
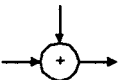
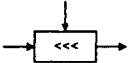
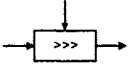
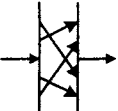
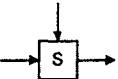
Основной идеей практически всех блочных шифров является тот факт, что последовательность бит любой длины Q можно представить в виде натурального числа из диапазона $[0 \dots 2^Q - 1]$. Более того, один и тот же объем данных, разбивая на подобные блоки разными способами, можно представить как в виде большого набора небольших чисел, например, от 0 до 15 или от 0 до 255, так и в виде нескольких огромных чисел, например, диапазона $[0 \dots 4294967295]$. Здесь все зависит от схемы разбиения потока данных на блоки бит. Вот над этими натуральными числами и производится множество простейших математических и алгоритмических преобразований, т. н. *криптопримитивов* — из числа тех, которые легко выполняются на современной архитектуре ЭВМ.

Для криптопреобразований нет никакой разницы, представляет ли обрабатываемая группа бит (число) какой-либо определенный смысл или нет. Например, на каком-то промежуточном этапе в обрабатываемом 8-битном блоке первые два бита могут быть позаимствованы из третьей буквы шифруемого текста, следующие три — из шестой, а последние три бита вычислены на основе первой и пятой букв. Главное, чтобы после восстановления на приемной стороне все биты в блоке заняли свои правильные места и получили правильные значения, т. е. снова превратились в осмысленный текст. В подобном абстрагировании от смысловой нагрузки на промежуточных стадиях и заключена сила двоичного представления данных.

Обратимые операции в блочном шифровании

Несомненно, что для однозначного дешифрования текста на приемной стороне все применяемые преобразования должны быть обратимы. Ни на одном из этапов шифрования из обрабатываемой последовательности не должно происходить "потери" информации. Перечень наиболее часто применяемых преобразований с их условными обозначениями на схемах приведен в табл. 17.2.

Таблица 17.2. Основные обратимые криптопримитивы

Операция	Условные обозначения	Математическая нотация	Примечания
Сложение		$X' = X + V$	При переполнении результатом разрядности операндов бит переноса отбрасывается — потери информации при этом не происходит
Исключающее "ИЛИ" (XOR)		$X' = X \oplus V$ или $X' = X \text{ XOR } V$	Операция обратима сама себе
Умножение по модулю $2^N + 1$		$X' = (X * V) \bmod (2^N + 1)$	Данное преобразование обратимо. Сомножитель по известному произведению и другому сомножителю находится по алгоритму Евклида.
Циклический битовый сдвиг влево		$X' = X \text{ ROL } V$	Операция является обратной к "циклическому сдвигу вправо"
Циклический битовый сдвиг вправо		$X' = X \text{ ROR } V$	Операция является обратной к "циклическому сдвигу влево"
Произвольная перестановка бит			Данная операция рассчитана в основном на аппаратную реализацию, и поэтому сейчас в значительной степени устарела
Табличная подстановка		$X' = S - \text{Box}[X]$	

Несколько слов о табличных подстановках, которые занимают отдельное место в криптографических преобразованиях. С помощью *табличных подстановок* (англ. *Substitute-box* — *S-box*) реализуются наиболее общие законы изменения данных, зачастую не описываемые ни аналитически, ни алгоритмически. В тех случаях когда в шифрующем алгоритме необходимо реализовать довольно сложную для микропроцессора или вообще не описываемую алгоритмом функцию, наиболее подходящим решением является занести все ее значения в массив, хранящийся в постоянной или оперативной памяти. Подобное задание функции называется *табличным*.

Количество бит в исходном операнде называется *разрядностью входа* табличной подстановки, количество бит в результате — *разрядностью выхода*. В большинстве случаев табличные подстановки используются в качестве *биективной* (однозначно обратимой) функции — в этом случае разрядность входа и выхода совпадает. Наиболее часто по такой схеме применяются подстановки 4×4 (читается "*четыре на четыре*") бита и 8×8 бит. Обратное преобразование ($S\text{-}Box^{-1}$), если в нем есть необходимость, задается на дешифрующей стороне также таблицей и, соответственно, такой же размерности. В тех случаях когда от подстановки не требуется обратимость, наряду с традиционными применяются подстановки 4×8 , 8×32 бита и некоторые другие.

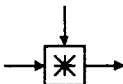
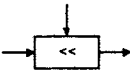
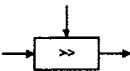
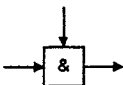
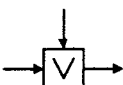
Запись функции в таблице имеет свои достоинства и недостатки. По сравнению с долгим вычислением значения функции выборка из таблицы производится гораздо быстрее (в принципе за две операции процессора). Однако эти две операции не сравнимы по быстродействию с обычным сложением или *XOR* — ведь одним из операндов в выборке всегда является ячейка памяти, и предсказать заранее, какая именно ячейка будет востребована, невозможно. Поэтому на современных процессорах табличная выборка производится довольно быстро только тогда, когда вся таблица размещается в кэш-памяти процессора. В противном случае каждая операция выборки инициирует многотактовый процесс обращения к настоящей оперативной памяти. Все это накладывает жесткие ограничения на размер хранимой таблицы и, соответственно, разрядность ее входа и выхода.

Одним из неоспоримых преимуществ табличных подстановок является быстродействие в отношении нелинейных операций. Практически все приведенные в табл. 17.2 операции, кроме умножения по модулю 2^N+1 и табличных подстановок, обладают очень высокой линейной зависимостью между входными и выходными значениями. Это очень плохо в отношении криптостойкости шифра к линейному криптоанализу, который будет детально рассмотрен в разд. "*Основы криптоанализа симметричных шрифтов*" данной главы. В плане повышения стойкости шифра к этому классу атак табличная подстановка намного опережает операцию умножения по модулю 2^N+1 , и является на сегодняшний день самым применяемым нелинейным преобразованием.

Необратимые операции в блочном шифровании

В тех случаях, когда от преобразования не требуется обратимости по входным параметрам, дополнительно к описанным обратимым функциям применяются операции, приведенные в табл. 17.3.

Таблица 17.3. Основные необратимые криптопримитивы

Операция	Условные обозначения	Математическая нотация	Примечания
Умножение		$X' = (X \cdot V) \bmod (2^N)$	$X' = (X \times V) \bmod (2^N)$
Остаток (модуль) от деления		$X' = X \bmod V$	$X' = X \bmod V$
Арифметический битовый сдвиг влево		$X' = X \text{ SHL } V$	$X' = X \text{ SHL } V$
Арифметический битовый сдвиг вправо		$X' = X \text{ SHR } V$	$X' = X \text{ SHR } V$
Логическое "И"		$X' = X \text{ AND } V$	$X' = X \text{ AND } V$
Логическое "ИЛИ"		$X' = X \text{ OR } V$	$X' = X \text{ OR } V$

Что может выступать в качестве параметра V в обратимых и необратимых операциях? Во-первых, конечно же, *фиксированные числовые константы*. Единственным ограничением, которым всегда пользуются создатели блочных шифров, является логическое обоснование ее выбора. В тех случаях когда шифр полон фиксированных числовых констант, выбор которых трудно объяснить, вполне может появиться подозрение, что создатели планомерно конструировали их для получения каких-либо недокументированных возможностей по взлому шифра. Поэтому, когда применение фиксированных числовых констант выполняет только роль "перемешивания" данных, их обычно выбирают из двоичной записи общеизвестных, не связанных ни с чем величин, например, чисел π (отношение диаметра к длине окружности), e (основание натурального логарифма) или ϕ (золотое сечение), или как контрольную сумму какого-либо распространенного текста.

Вторым возможным значением параметра V является материал ключа. *Материалом ключа* называется блок данных, вычисленный на основании ключа шифрования и зависящий только от него. В простейшем случае материалом ключа может выступать сам ключ (если его разрядность совпадает с разрядностью операции) либо произвольная его часть без какой-либо обработки. Однако прямое применение ключа в преобразованиях может привести к раскрытию некоторой информации о нем, например, при большом количестве известных пар "открытый текст/зашифрованный текст". Конечно, вероятность подобной атаки крайне мала и зависит только от общей структуры шифра, но криптография оперирует и гораздо меньшими рисками, поэтому общепринятым подходом является "перестраховаться и в этом случае". Для этого в качестве материала ключа обычно используют значения, вычисленные из него по необратимой схеме. При этом лучше всего, чтобы любой элемент материала зависел от всех бит ключа — даже при малейшем изменении ключа (например, только в одном бите) должен меняться и весьма значительно каждый блок материала ключа.

Операции, в которых в качестве параметра V используется материал ключа, называются *перемешиванием с ключом* или *добавлением ключа* (никакой связи с арифметическим сложением данный термин не несет). Подобных операций за время шифрования одного блока производится достаточно много. Таким образом, решается одно из главных требований к шифрам — невозстановимость ключа. Действительно, если к блоку данных добавить одну и ту же информацию, но неоднократно и разными способами, то подобная операция будет необратима относительно добавленной информации, хотя обратимость относительно первого операнда останется. Например, пусть X и K — четырехбитные исходное число и ключ соответственно (диапазон возможных значений 0...15 у каждого), тогда формула:

$$X' = ((X + K) \text{ XOR } (K + 7)) \text{ AND } 1111_2$$

(нижний индекс показывает систему исчисления)

является обратимой относительно X — всегда по известным X' и K можно восстановить X , но необратимой относительно K по известной паре X и X' . Не существует аналитической формулы, выражающей K через X и X' . Конечно, всегда можно составить табличную зависимость, перебрав при известном X все K и записав все получившиеся X' , но это легко сделать только в нашем гипотетическом примере из 16 возможных значений K . На практике же при длине ключа в 128 бит для решения этой задачи не хватит ни ресурсов памяти для хранения, ни, что главное, вычислительных возможностей ЭВМ. Таким образом, многократное добавление материала ключа по сложной схеме является надежным средством защиты блочных шифров от атаки по известным парам открытый/зашифрованный текст.

Третьим возможным значением параметра V является значение, вычисленное от независимой части шифруемого блока. Данный метод лежит в основе схемы блочного шифрования, называемой сетью Файштеля (автор — Н. Feistel), подробно рассмотренной в разд. "Сети" данной главы. Операции,

которые задействуют в качестве второго параметра само значение шифруемого блока, обратимы, только если они оперируют независимыми частями блока. Например, на каком-либо промежуточном этапе на первый байт операцией *XOR* накладывается значение пятого байта и т. п. Обратимость обеспечивается на том основании, что на приемной стороне алгоритм, подходя к дешифровке этого места, уже "знает" значение пятого байта, которым проводилось наложение, и повторным выполнением *XOR* восстанавливает первый байт. Проблема возникла, если бы мы наложили на первый байт функцию от этого же первого байта. Тогда, даже несмотря на потенциальную обратимость операции, дешифрующая сторона не смогла бы вычислить значение *V* и зашла бы в тупик.

Сети

Любой блочный шифр является многократной комбинацией описанных операций по определенной схеме или без таковой. В том случае, когда в последовательности применения операций не прослеживается закономерности, шифр называется *прямым*, или *комбинационным*. Подобные алгоритмы давно не используются в традиционной цифровой криптографии, однако, по-прежнему разрабатываются любителями. Следует признать, что хотя и не существует способов доказательства практической стойкости подобных шифров, некоторая (очень малая) часть из них может классифицироваться как надежные.

Одной из наиболее распространенных схем блочных шифров являются *сети* (англ. *network*). Блочный шифр, построенный по такой схеме, состоит из многократных повторений, называемых *циклами*, или *раундами* (англ. *round*), нескольких видов операций, называемых *слоями*. Разбиение всего процесса шифрования на несколько однотипных слоев позволяет:

- сократить размер программного кода использованием цикла;
- унифицировать "алгоритмическую формулу" шифрования и, как следствие, упростить проверку стойкости шифра к известным видам криптоатак;
- сделать шифр легко усложняемым при необходимости (путем увеличения числа раундов);
- ввести понятие *ключей раунда*, на которые разбивается весь материал ключа.

Некоторые из слоев не содержат в себе процедур смешивания с ключом. Обычно они выполняют функции обратимого перемешивания данных внутри блока, либо защищают блочный шифр от какого-либо конкретного класса криптоатак. Подобные слои выполняют свои преобразования всегда по одной и той же схеме. В тех слоях, где в качестве параметров операций используется материал ключа, на каждом раунде используются обычно все новые и новые блоки из материала ключа — *ключи раунда*. Поэтому каждый подобный слой на самом деле производит свое уникальное преобразование материала. Эта схема позволяет наряду с унифицированностью

преобразований получить очень сложную для дешифрования при неизвестном ключе схему.

Одной из самых простых и ранних сетей является *SP-сеть* (англ. *SP-network*), каждый раунд которой состоит из двух слоев. Свое название данная сеть получила от сокращений двух английских слов: *Substitution* (подстановка) и *Permutation* (перестановка). В слое подстановки над данными производятся преобразования класса сложения, исключающего ИЛИ, табличных подстановок, в качестве параметров используются константы и материал ключа. В слое перестановки биты либо реже байты меняются местами внутри блока обычно по фиксированной (не зависящей от ключа) схеме.

Более современной модификацией технологии сетей является *KASLT-сеть* с тремя слоями в каждом раунде. Первый слой — *добавление ключа* (*Key Addition, KA*) — производит смешивание данных с ключом раунда. Смешивание производится обычно какой-либо простой операцией — сложением либо *XOR*. Второй слой — *подстановка* (*Substitution, S*) — производит алгоритмическую или чаще табличную подстановку для обеспечения нелинейных свойств шифра. Третий слой — *линейное преобразование* (*Linear Transformation, LT*) — производит активное обратимое перемешивание данных внутри блока. Как видим, по сравнению с *SP-сетью* здесь добавление ключа вынесено в отдельный слой, разделены линейная и нелинейная часть преобразования, а перемешивание данных производится с помощью быстро реализуемых в программной реализации линейных преобразований.

Практически независимой от двух предыдущих разработок и очень популярной схемой блочных шифров является *сеть Файштеля*. В каждом из ее раундов только по одному слою, т. е. все преобразования однотипны. Однако сама структура преобразования специфична: на какую-либо часть шифруемого блока обратимой операцией накладывается значение функции (возможно, необратимой), вычисленное от другой части блока. Классическая структура сети Файштеля приведена на рис. 17.11.

Независимые потоки информации, порожденные из исходного блока, называются *ветвями сети*. В классической схеме их две. Величины *I* и *J* именуются *параметрами сети*, обычно их роль играет материал ключа. Функция *F* называется *образующей*. Оптимальное число раундов для сети Файштеля *K* — от 8 до 32. Сразу же возникает вопрос, является ли данная схема обратимой? Несомненно, да. Сеть Файштеля обладает тем свойством, что даже если в качестве образующей функции *F* будет использовано необратимое преобразование, то и в этом случае вся цепочка будет восстановима. Это происходит вследствие того, что для обратного преобразования сети Файштеля не нужно вычислять функцию F^{-1} — и при дешифровании вычисляется только прямое значение *F*. Обратимость требуется только для операции наложения *F* на шифруемую в данном раунде ветвь, поэтому здесь чаще всего применяется *XOR*, реже — арифметическое сложение.

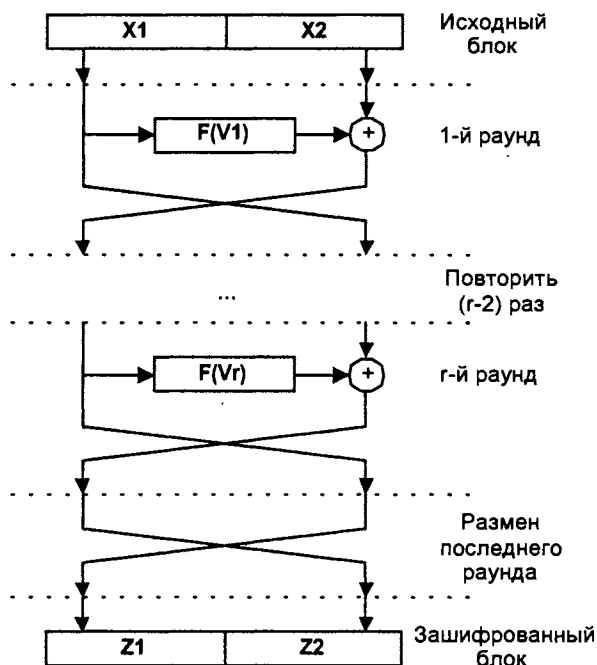


Рис. 17.11. Классическая сеть Файштеля

Более того, как нетрудно заметить, сеть Файштеля, в которой для наложения значения F используется операция XOR , симметрична. Исключающее ИЛИ обратимо своим же повтором, и потому инверсия последнего обмена ветвей делает возможным раскодирование блока той же сетью Файштеля, но только с обратным порядком параметров V_i . Заметим, что для обратимости сети Файштеля не имеет значения, является ли число раундов четным или нечетным числом. В большинстве реализаций сети Файштеля прямое и обратное преобразования производятся одним и тем же фрагментом программного кода, оформленного в виде процедуры, которому в качестве параметра передается вектор величин V_i либо в исходном (для шифрования), либо в инверсном (для дешифрования) порядке.

С незначительными доработками сеть Файштеля можно сделать и абсолютно симметричным шифром, т. е. выполняющей функции шифрования и дешифрования одним и тем же набором операций. Модификация сети Файштеля, обладающая подобными свойствами приведена на рис. 17.12. Как видим, основная ее хитрость в повторном использовании данных ключа в обратном порядке во второй половине цикла. Необходимо заметить, однако, что именно из-за этой недостаточно исследованной специфики такой схемы (то есть потенциальной возможности ослабления зашифрованного текста обратными преобразованиями) ее используют в криптоалгоритмах с большой осторожностью.

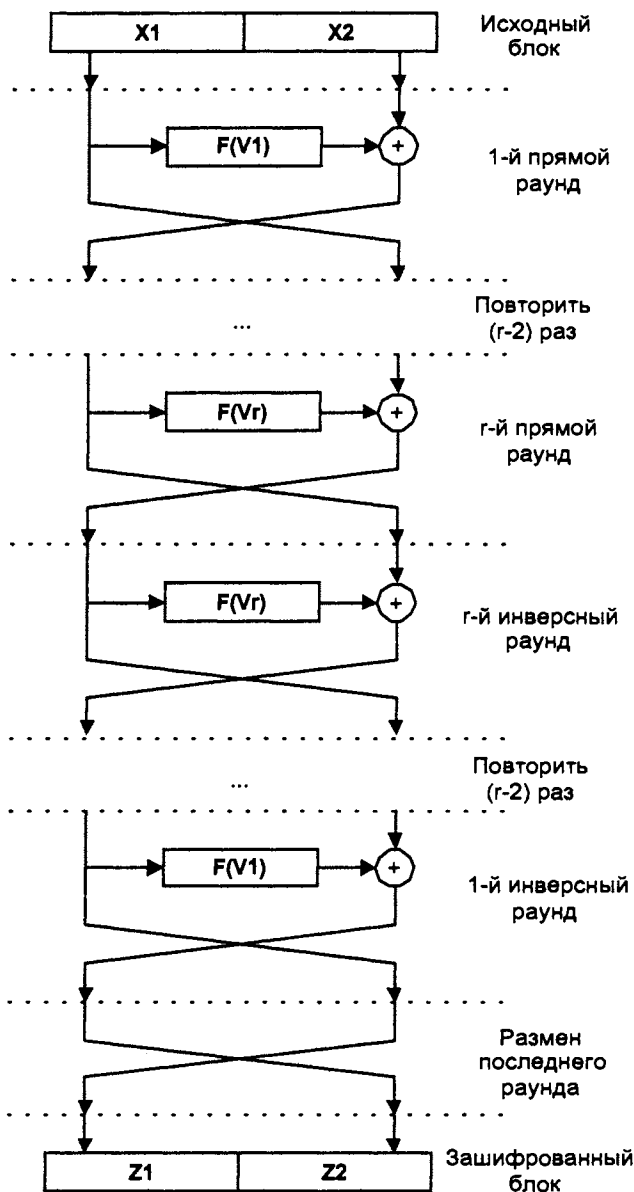


Рис. 17.12. Абсолютно симметричная сеть Файштеля

А вот модификацию сети Файштеля для большего числа ветвей применяют гораздо чаще. Это в первую очередь связано с тем, что при больших размерах шифруемых блоков (128 и более бит) становится неудобно работать с математическими функциями по модулю 64 и выше. Как известно,

основные единицы информации, обрабатываемые процессорами, на сегодняшний день — это байт и двойное машинное слово (32 бита). Поэтому все чаще в блочных криптоалгоритмах встречается сеть Файштеля с 4 ветвями. Самый простой принцип ее модификации изображен на рис. 17.13. Для более быстрого перемешивания информации между ветвями (а это основная проблема сети Файштеля с большим количеством ветвей) применяются две модифицированные схемы, называемые *тип-2* и *тип-3*. Они изображены на рис. 17.14 и 17.15 соответственно. За счет частичного объединения обработки потоков в двух последних схемах количество раундов в них лишь ненамного больше количества раундов в сети Файштеля с двумя ветвями.

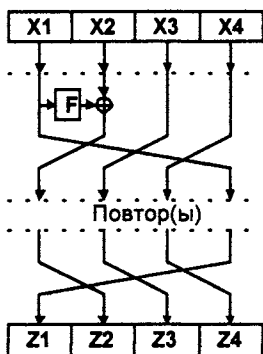


Рис. 17.13. Раунд сети Файштеля с 4 ветвями, тип 1

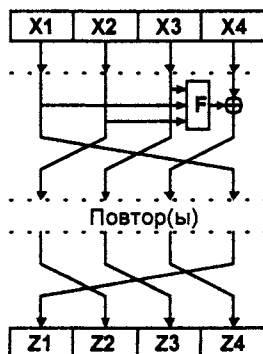


Рис. 17.14. Раунд сети Файштеля с 4 ветвями, тип 2

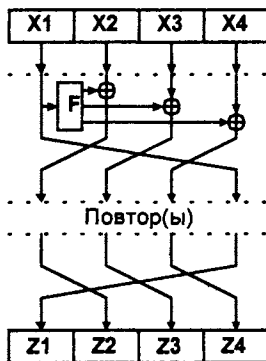


Рис. 17.15. Раунд сети Файштеля с 4 ветвями, тип 3

Основная часть современных криптоисследований, посвященных сети Файштеля, относится, конечно же, к структуре образующей функции F и правилу вычисления материала ключа K . Из других решений интерес представляют только дополнительные начальные и оконечные преобразования данных (англ. *whitening* — *забеливание*) над шифруемым блоком. Подобные обратимые преобразования выполняются либо операцией *XOR* либо сложением. Начальное "забеливание" имеет целью произвести перемешивание данных блока с частью материала ключа на самой ранней стадии и, тем самым, повысить рандомизацию входного текста — и, как следствие, еще сильнее снизить вероятность успешной атаки по известному открытому тексту. Выходное преобразование усложняет обратную трассировку при частично известном злоумышленнику ключе, а, кроме того, делает сеть Файштеля с "забеливанием" снова симметричной.

Практика блочного шифрования

Наиболее простой путь ознакомиться с основами блочного шифрования — рассмотреть, хотя бы поверхностно, наиболее важные для своего времени разработки в этой области. Изложение пойдет в хронологическом порядке с конца 70-х годов прошлого века — первых электронных блочных шифров до наших дней — конкурса на криптостандарт США XXI века.

Аппаратное шифрование: DES

DES: структура

Традиционно первой знаменательной вехой в истории блочного шифрования считается опубликование в 1977 году и принятие в 1980 году в США национального Стандарта шифрования данных DES (*Data Encryption Standard*). Параметры алгоритма таковы: разрядность блока — 64 бита, размер ключа — 56 бит. Алгоритм представляет собой классическую сеть Файштеля из 16 раундов с добавлением входной и выходной перестановки бит. Общая структура алгоритма представлена на рис. 17.16.

Во входной перестановке используется следующее правило перестановки бит (приведены порядковые номера бит из исходных данных, считая с "1"):

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

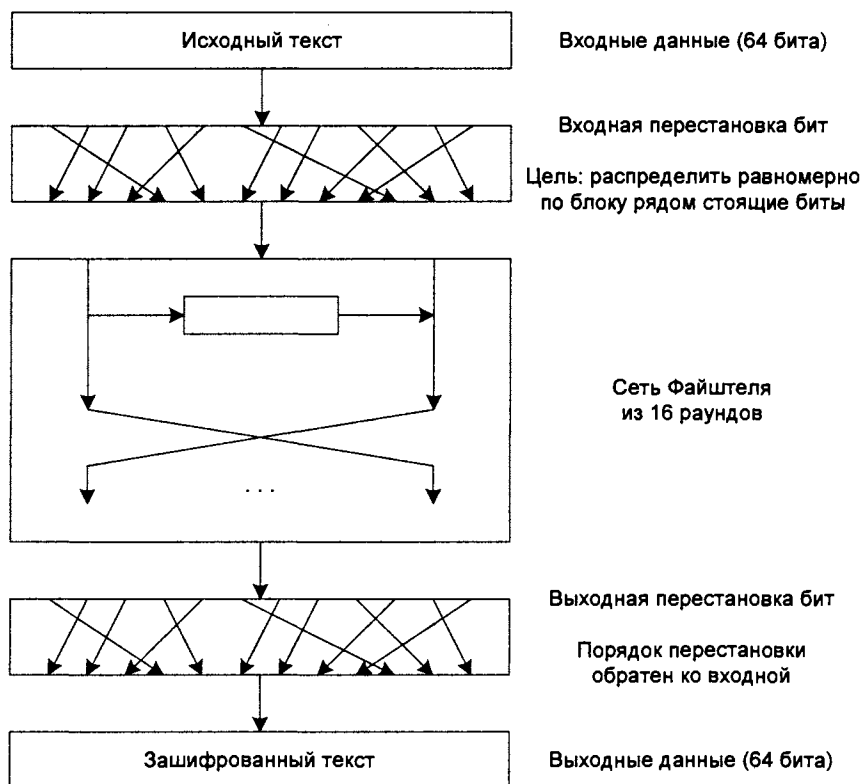


Рис. 17.16. Общая структура DES

Общая формула для вычисления индексов в псевдокоде имеет вид:

цикл по i от 0 до 63

$$\text{OUT} [(i_5 i_4 i_3 i_2 i_1 i_0)] = \text{IN} [(7 - (i_2 i_1 i_0) \parallel (i_4 i_3) \parallel 1 - (i_5))],$$

где запись x_r обозначает r -й бит в записи числа x , символ " $\parallel$ " — конкатенацию цифр в двоичной записи числа.

Выходная перестановка бит является обратной ко входной, правило перестановки бит в ней имеет вид:

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

Добавление материала ключа производится только в сети Файштеля, поэтому, по сути, только она выполняет собственно шифрование данных.

Разрядность ключей раундов, применяемых в DES равна 48 битам. Для симметричности алгоритма шифрования/дешифрования в DES, как и в классической сети Файштеля, производится дополнительный обмен последнего обмена ветвей. Это позволяет производить шифрование и дешифрование одним и тем же программным кодом, передавая ему в первом случае ключи раундов в прямом порядке ($k_0, k_1, \dots, k_{15}$), а во втором случае — в обратном ($k_{15}, k_{14}, \dots, k_0$). Общий вид сети Файштеля алгоритма DES приведен на рис. 17.17.

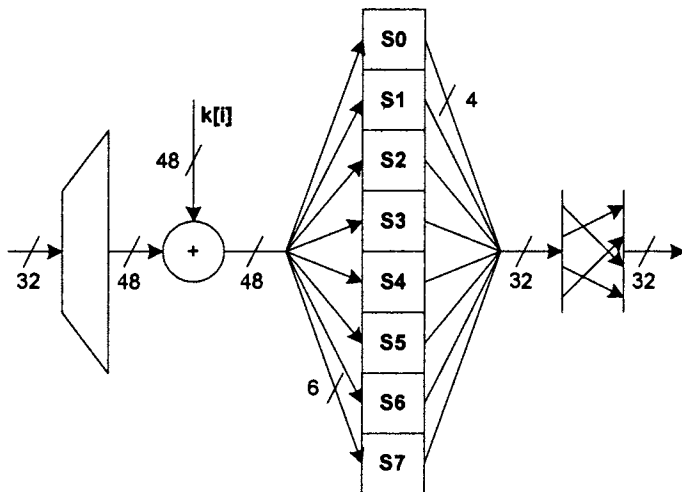


Рис. 17.17. Сеть Файштеля в алгоритме DES

Образующая функция сети в алгоритме DES состоит из 4 операций:

1. перестановка бит/расширение блока с помощью повторов по определенной схеме;
2. наложение ключа раунда операций *XOR*;
3. табличные подстановки;
4. перестановка бит.

Расширение входных данных производится на уровне бит записью некоторых из них в выходной поток дважды. В результате из 32 бит получается 48-битный блок информации, совпадающий по размеру с ключом раунда. Закон расширения выглядит следующим образом (приведены порядковые номера битов исходного блока, считая с "1"):

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.

Наложение ключа раунда производится операцией *XOR*, разрядность операции — 48 бит. Правило генерации ключей раунда рассмотрено в разд. "DES: расширение ключа" настоящей главы.

Табличные подстановки, применяемые в DES, имеют разрядность 6×4 бита. 48-битный блок данных после наложения ключа разбивается на 8 блоков по 6 бит. Каждый блок пропускается через свою табличную подстановку, обозначаемую S_0, S_1 и т. д. В результате получаются 8 блоков по 4 бита в каждом: итого 32 бита — первоначальная разрядность обрабатываемой ветви. Табличные подстановки алгоритма DES приведены в *приложении 4*.

Заключительной операцией образующей функции является перестановка бит внутри 32-битного блока, отвечающая за перемешивание информации. Правило перестановки выглядит следующим образом (приведены порядковые номера битов из 32-разрядного блока — результата табличных подстановок, считая с "1"):

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

Общий вид одного раунда сети Файштеля алгоритма DES приведен на рис. 17.18.

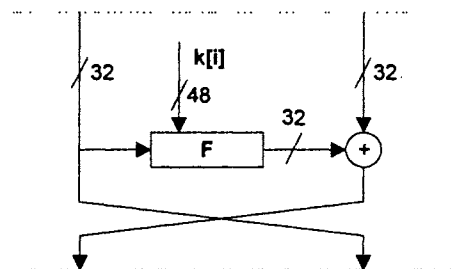


Рис. 17.18. Один раунд DES

DES: расширение ключа

Расширение ключа (создание ключей раунда) производится в DES по следующей схеме. На основе 56 значащих бит из ключа на начальном этапе создаются два 28-битных вектора C_0 и D_0 путем выбора бит из ключа в определенной последовательности.

Закон выбора бит из ключа для C_0 :

57, 49, 41, 33, 25, 17, 9, 1, 58, 50, 42, 34, 26, 18,
10, 2, 59, 51, 43, 35, 27, 19, 11, 3, 60, 52, 44, 36.

Закон выбора бит из ключа для D_0 :

63, 55, 47, 39, 31, 23, 15, 7, 62, 54, 46, 38, 30, 22,
14, 6, 61, 53, 45, 37, 29, 21, 13, 5, 28, 20, 12, 4.

Институтом стандартизации рекомендовано при хранении ключа для DES пользоваться простейшим помехоустойчивым кодированием — проверкой четности. Поэтому в каждом байте значащими являются только первые 7 бит, а восьмой добавляется при записи, исходя из правила "количество единиц в двоичной записи каждого байта нечетно". Таким образом, при истинной разрядности ключа в 56 бит хранится он в 64-битной структуре. Именно этим объясняется появление в законах выбора бит для C_0 и D_0 номеров, больших 56, и отсутствие бит с номерами 8, 16, 24, 32, 40, 48, 56, 64 (они являются контрольными — незначащими).

Для генерации каждого очередного ключа раунда k_i вектора C_0 и D_0 циклически сдвигаются влево на количество разрядов, указанное в табл. 17.4.

Таблица 17.4. Величины циклических сдвигов векторов C и D при генерации ключа

I	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ROL на... разрядов	1	2	4	6	8	10	12	14	15	17	19	21	23	25	27	28

Для выбора 48 бит ключа i -го раунда, получившиеся после циклических сдвигов вектора $C_i D_i$, объединяются, и из 56 бит получившегося массива отбираются 48 по следующему закону:

14, 17, 11, 24, 1, 5, 3, 28, 15, 6, 21, 10, 23, 19, 12, 4,
26, 8, 16, 7, 27, 20, 13, 2, 41, 52, 31, 37, 47, 55, 30, 40,
51, 45, 33, 48, 44, 49, 39, 56, 34, 53, 46, 42, 50, 36, 29, 32.

DES: анализ

Первое, что бросается в глаза, — это огромное количество битовых перестановок и выборок в алгоритме. Это неудивительно, основной схемой разработчиками планировалась только аппаратная реализация. Этому подчинено и наложение ключа именно операцией *XOR*, реализуемой в этом случае наиболее просто и распараллелено, и небольшой размер и минимальное количество табличных подстановок. Аппаратная реализация алгоритма на отдельной или встроенной микросхеме позволяет достичь высокой скорости шифрования при очень небольшом объеме устройства.

Единственным нелинейным элементом алгоритма является этап табличных подстановок в образующей функции. Создатели алгоритма не объяснили принцип выбора используемых таблиц, однако привели несколько правил,

которым удовлетворяют выбранные таблицы с целью защиты от некоторых видов криптоанализа.

Алгоритм обладает некоторыми специфичными свойствами, которые, однако, по мнению большинства специалистов, не являются значительными уязвимостями. Первым таким свойством является следующий закон: если $Y = \text{Encrypt}(X, \text{KEY})$, то $Y' = \text{Encrypt}(X', \text{KEY})$, где X' — побитная инверсия X , Y' — побитная инверсия Y , KEY' — побитная инверсия KEY .

Вторым свойством является наличие среди множества 2^{56} ключей четырех *слабых ключей* (англ. *weak keys*). Это следующие числа в шестнадцатеричной записи (с контрольным 8-м битом в каждом байте):

(01h 01h 01h 01h 01h 01h 01h 01h)

(FEh FEh FEh FEh FEh FEh FEh FEh)

(1Fh 1Fh 1Fh 1Fh 0Eh 0Eh 0Eh 0Eh)

(E0h E0h E0h E0h F1h F1h F1h F1h)

При шифровании данных любым из этих ключей среди 2^{64} вариантов входных данных существует 2^{32} блоков, которые в результате шифрования преобразуются сами в себя, т. е. для них $\text{Encrypt}(X, \text{WEAK\_KEY}) = X$. Несмотря на то, что вероятность подобного "бесполезного" шифрования очень велика ($1/2^{32}$ — это неприемлемо большой риск для криптографии), вероятность того, что сам ключ окажется слабым, очень мала ($1/2^{54}$). Это позволяет считать наличие слабых ключей у DES скорее оригинальной особенностью, чем реальной уязвимостью.

За многие годы исследований DES зарекомендовал себя надежным шифром. Относительно него было разработано несколько криптоатак, требовавших для выяснения ключа несколько меньше операций, чем взлом полным перебором. Однако ни одна из них не имела настолько качественного превосходства над методом полного перебора, чтобы ее можно было назвать полноценным взломом алгоритма. По-настоящему заставили DES уйти с рынка конкурентно-способных блочных шифров две причины: ориентированность на аппаратную реализацию и слишком малый размер ключа.

Эпоха пост-DES

Общие тенденции

Взлет производительности вычислительной техники внес очень сильные коррективы в развитие криптографии. Первыми "пострадали" примитивные операции, которые неприемлемо долго исполнялись в программной реализации. К таковым относятся, например, перестановки бит. Если в аппаратной схеме перестановка реализуется всего лишь распайкой контактов и дорожек и выполняется за один условный такт, то в универсальных

программных модулях типа микропроцессора ситуация гораздо сложнее. В общем случае на перестановку или выборку каждого бита требуется одна или две команды процессора, что требует, например, для перестановки 32 бит исполнения 64 операций. Тщательная оптимизация может позволить сократить их количество за счет параллельной обработки нескольких бит в 2–3 раза, но не более того.

Следующим параметром, величина которого намного быстрее, чем ожидалось, стала неприемлемо мала, оказалась длина ключа. Экспоненциальный рост быстродействия электронно-вычислительной техники (ЭВТ), появление параллельных архитектур и возможность быстрого объединения огромного количества ЭВМ в единую сеть свели время перебора всех вариантов 56-битного ключа до вполне приемлемых сроков. В настоящее время не взятым рубежом для стойких шифров является уже 128-битная длина ключа, во всяком случае, публично о фактах такого взлома для надежных шифров не сообщалось.

Безальтернативная тенденция к более универсальной программной реализации вынудила создателей криптоалгоритмов считаться с законами этого направления. Во-первых, для улучшения основной характеристики "скорость_шифрования/стойкость" требовались не только быстрые в программном исполнении операции, но и оптимизация их разрядности под разрядность микропроцессора. За довольно продолжительным периодом 8-битных операций проследовал короткий 16-битный и уступил свое место нынешней повсеместной 32-разрядной оптимизации.

Как, впрочем, скоро оказалось, ориентированность под какую-либо доминирующую на данном этапе архитектуру ЭВМ не совсем удачное решение. Криптография все активнее входит в повседневную жизнь: электронная коммерция, электронная защита авторских прав, электронные нотариальные системы — вот неполный перечень областей ее применения. Эта тенденция приводит к тому, что программных комплексов с совершенно различными характеристиками и ресурсами, использующих криптографию, появляется все больше и больше. Хорошо построенный криптоалгоритм должен пользоваться всеми "благами" мощных вычислительных комплексов с огромными объемами свободной оперативной памяти и при этом не слишком медленно работать на смарт-картах с минимальными ресурсами памяти. Это заставляет исследователей искать новые универсальные операции и оптимизировать уже существующие.

Еще одной неожиданной задачей для криптологов оказалась оптимизация алгоритмов под параллельные архитектуры ЭВМ. Все чаще в специализированных системах и серверных станциях можно встретить многопроцессорные платформы. Очевидно, что упускать такую возможность просто

непозволительно, учитывая то, что основным ресурсом, потребляемым криптопреобразованиями, является как раз процессорное время. Количественным критерием, характеризующим способность алгоритма к распараллеливанию, является *среднее количество потоков* (процессоров), на которые можно разделить данную задачу. Естественно, что во многих точках алгоритма обрабатываемые данные должны сходиться вместе на одном процессоре для перемешивания, так необходимого в криптопреобразованиях. Однако количество подобных точек необходимо уменьшать в разумных пределах, а количество независимых потоков, на которые можно разделить процесс между такими точками, устремить хотя бы к четырем-шести-восьми. В целом снова получается проблема поиска золотой середины между стойкостью шифра и скоростью его работы. Проследим за эволюцией блочных шифров 90-х годов XX века.

ТЕА: структура

Пожалуй, самым простым представителем семейства программно ориентированных надежных блочных шифров является *ТЕА* (*Tiny Encryption Algorithm*). Дословно его название переводится на русский как "крошечный криптоалгоритм". Разработан он был в исследовательском центре защиты информации Кембриджского университета и представляет собой классическую сеть Файштеля из 64 раундов. Схема алгоритма приведена на рис. 17.19. Алгоритм оптимизирован под 32-разрядные микропроцессоры, размер блока — 64 бита.

Результат образующей функции накладывается на независимую ветвь операцией "сложение", поэтому восстановить исходный текст простым обращением порядка ключей раунда не удастся. Алгоритм ТЕА является примером несимметричной сети Файштеля — для дешифрования придется реализовывать отдельную процедуру, хотя, как и шифрование, очень простую.

ТЕА: расширение ключа

Алгоритм использует 128 битный ключ, который превращается в ключи раунда по очень простой схеме. Ключ разбивается на четыре 32-разрядных машинных слова K_0, K_1, K_2, K_3 . Во всех четных (считая с 0) раундах в качестве 64-битного ключа раунда используется пара (K_0, K_1) , в нечетных — пара (K_2, K_3) . В общепринятой схеме номер раунда обозначается переменной i , количество раундов — r , компоненты ключа раунда — как массив $k[i]$ (он использует сквозную нумерацию по машинным словам — в нулевом раунде используются $k[0]$ и $k[1]$, в первом — $k[2]$ и $k[3]$ и т. д.). В таких обозначениях общая формула создания 128 компонент материала ключа для ТЕА выглядит следующим образом: $k[i] = K_{(i \bmod 4)}$.

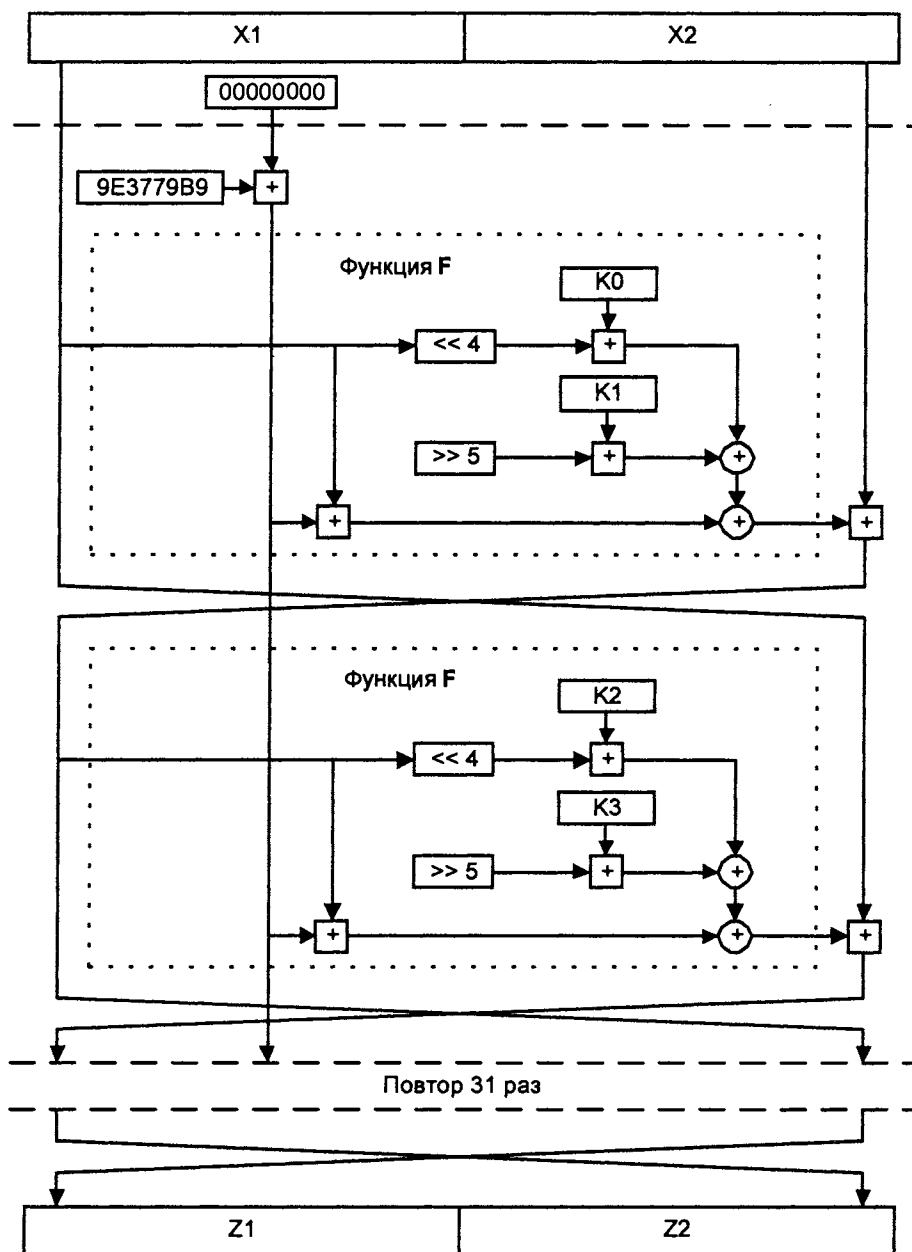


Рис. 17.19. Алгоритм TEA

TEA: анализ

В образующей функции TEA используются только битовые сдвиги, сложение и исключающее ИЛИ. Это приводит к тому, что в алгоритме полностью отсутствуют нелинейные преобразования. Авторы, по их заверениям, все же добиваются стойкости к линейному криптоанализу — основным объяснением этому является нехарактерно большое для сети Файштеля число раундов. Согласно исследованиям разработчиков, TEA стоек ко всем опубликованным на сегодняшний день видам криптоатак. Однако из-за своей предельной простоты, которая может вызывать подозрение на наличие уязвимости к какому-либо специфичному виду криптоанализа, алгоритм все-таки не получил широкого применения на практике.

IDEA: структура

Алгоритмом, который был разработан специально на смену DES, удовлетворял почти всем требованиям нового криптопоколения и получил в отличие от TEA широкое применение, является *IDEA* (*International Data Encryption Algorithm*). Пробразом шифра стал алгоритм PES (*Proposed Encryption Standard*), опубликованный в 1990 году группой специалистов из различных европейских организаций. Однако PES оказался уязвим к одной из видов криптоатак, и в его схему были внесены небольшие изменения. В результате на свет "появилась" IDEA. В настоящее время правами на алгоритм обладают Швейцарский федеральный технологический институт ETHZ (*Swiss Federal Institute of Technology Zurich*) и компания *Ascom Tech Ltd*.

IDEA построена по схеме, очень напоминающей сеть Файштеля. Подобный принцип проектирования блочных шифров в классификации получил название *обобщенной сети Файштеля*. Исходный блок данных длиной в 64 бита разбивается на 4 ветви — все операции в IDEA выполняются в 16-битной арифметике. Затем в каждом из 8 раундов над ветвями и материалом ключа выполняются в специально подобранной очередности три операции: сложение, исключающее ИЛИ и модифицированное умножение. Схема одного раунда IDEA приведена на рис. 17.20. Модифицированное умножение изображено на схеме символом *. Длина ключа алгоритма составляет 128 бит.

Модифицированное умножение представляет собой обычное умножение по модулю 65537 ($2^{16} + 1$) с двумя условностями. Во-первых, если какой-либо из сомножителей равен 0, то перед умножением он заменяется на 2^{16} . Во-вторых, если получившееся после взятия модуля число равно 2^{16} , то оно заменяется на 0. Определенная подобным образом операция обладает следующими свойствами:

- ☐ отображает пару 16-разрядных чисел на 16-разрядное число,
- ☐ обратима (по известному результату и одному из сомножителей возможно однозначно определить второй сомножитель).

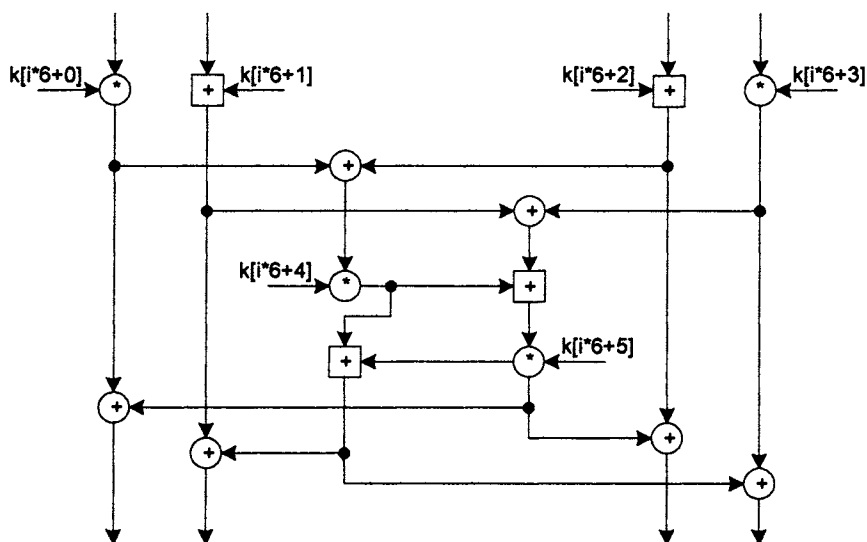


Рис. 17.20. Раунд алгоритма IDEA

Для достижения симметричности шифрования/дешифрования после восьми раундов над ветвями производится дополнительное преобразование. Общая схема алгоритма IDEA приведена на рис. 17.21. С помощью подобного дополнения создатели добились того, что для дешифрования в IDEA используется та же самая процедура, но в качестве 52 компонент материала ключа ей передается специальным образом модифицированные компоненты.

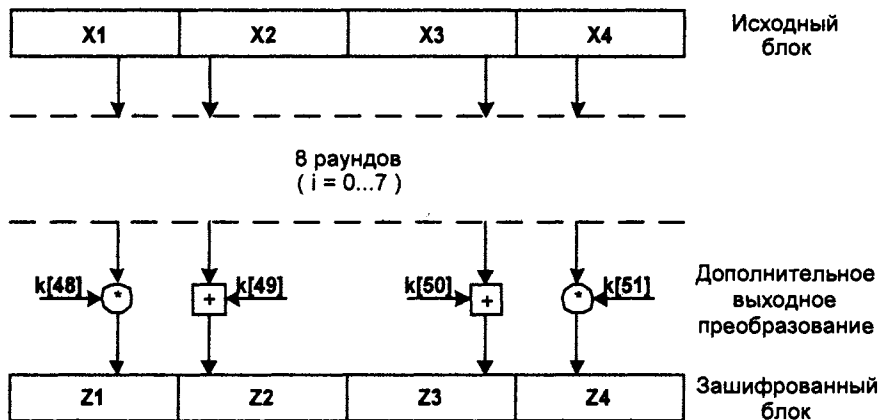


Рис. 17.21. Алгоритм IDEA

IDEA: расширение ключа

Получение из 128-битного ключа материала ключа в IDEA предельно просто. Первые 8 16-битных компонент $k[0]...k[7]$ материала ключа получаются непосредственно разбиением 128 бит ключа на блоки без каких-либо преобразований. Затем 6 раз повторяется следующая операция: ключ сдвигается циклически влево на 25 бит и очередные 8 компонент получают таким же разбиением уже сдвинутого ключа на 16-битные блоки.

В псевдокоде:

$k[0]...k[7] = \text{KEY}$

$\text{KEY} = \text{KEY} \text{ ROL } 25$

$k[8]...k[15] = \text{KEY}$

$\text{KEY} = \text{KEY} \text{ ROL } 25$

...

Для дешифрования в IDEA можно использовать тот же самый основной алгоритм, но предварительно специальным образом необходимо подготовить материал ключа. Если мы обозначим 52 компоненты материала ключа, используемые для шифрования через $e[0]e[51]$, а материал для дешифрования $d[0]d[51]$, то алгоритм создания вектора d по вектору e будет иметь следующий вид.

Если $(i \bmod 6 = 0)$, то $d[i] = \text{inv}(e[48-i])$.

Если $(i \bmod 6 = 3)$, то $d[i] = \text{inv}(e[54-i])$.

Если $(i \bmod 6 = 4)$, то $d[i] = e[50-i]$.

Если $(i \bmod 6 = 5)$, то $d[i] = e[52-i]$.

Если $(i \bmod 6 = 1 \text{ или } 2)$, то $d[i] = -e[51-i]$, за исключением:

$d[1] = -e[49]$, $d[2] = -e[50]$, $d[49] = -e[1]$, $d[50] = -e[2]$.

Операция $\text{inv}(x)$ представляет собой вычисление числа, обратного к x относительно операции модифицированного умножения, т. е. если некоторое число a будет сначала умножено на x , а затем на $\text{inv}(x)$, то в результате всегда получится снова число a . Операция умножения по модулю $(2^{16} + 1)$ была специально модифицирована авторами с тем, чтобы такое число $\text{inv}(x)$ существовало для любого x — т. е. операция умножения была обратима. Для нахождения $\text{inv}(x)$ на практике используется алгоритм Евклида.

Как видим, основная идея инвертирования материала ключа проста. Во-первых, все компоненты записываются в обратном порядке. Во-вторых, значения заменяются на такие числа, которые при повторном применении той же операции восстановят исходное значение ветви. Для арифметического сложения — это то же число, но со знаком минус, для умножения — величина $\text{inv}(x)$, а для операции XOR — само же наложенное при шифровании значение.

IDEA: анализ

Алгоритм интересен тем, что использует только три легко реализуемые на современных ЭВМ операции, не содержит таблиц подстановок и каких-либо констант. Все три операции являются обратимыми и реализуют различные схемы "перемешивания" данных с материалом ключа. Естественно, основной целью разработчиков было придумать такую последовательность их применения, чтобы получить наилучший коэффициент "стойкость/быстродействие". При проектировании подобной схемы авторы активно использовали теоретические исследования, посвященные свойствам в целом и криптостойкости в частности последовательного применения разнородных обратимых операций.

За полтора десятка лет с момента опубликования алгоритма наиболее серьезной найденной проблемой в его криптостойкости считается наличие набора из 2^{51} слабых ключей. При случайном попадании ключа на одно из таких значений злоумышленник может при определенных условиях восстановить значение ключа за 2^{17} операций. Бельгийский исследователь Джон Дэмен (*J. Daemen*), обнаруживший эту уязвимость, предложил и очень простое изменение в процедуре расширения ключа, решающее данную проблему. Однако, мировое сообщество посчитало шанс попадания ключа в группу слабых $2^{51} / 2^{128} = 1/2^{77}$ чрезвычайно низким и модификация не прижилась.

Группа исследователей в 1998 году представила на конференции схему криптоатаки на IDEA в том случае, если злоумышленник может измерять время, которое потратила ЭВМ на выполнение шифрования одного блока. Взлом ключа возможен при этом только в том случае, если процедура модифицированного умножения была написана по оптимизированной по времени исполнению схеме. Подобная схема широко применялась в различных реализациях IDEA до данного выступления на конференции. В ней в случае, если один из сомножителей был равен нулю, вычисления произведения не производилось (в результат просто записывался второй сомножитель со знаком минус — данное упрощение позволяет сделать математика). Исследование, представленное на конференции 1998 года показало, что подобная оптимизация несет больше вреда в стойкости, чем пользы в плане выигрыша нескольких тактов процессора.

В целом, IDEA считается одним из очень надежных и простых блочных криптоалгоритмов, но не удовлетворяет требованиям современной вычислительной техники (ВТ) из-за ориентированности на 16-битную архитектуру и, как следствие, из-за более низкого быстродействия, чем могло бы быть достигнуто 32-разрядным алгоритмом при "примерно той же" криптостойкости.

ГОСТ 28147-89: структура

Криптоалгоритм ГОСТ 28147-89 [ГОСТ 28147-89] является легендарным и, пожалуй, единственным из официально опубликованных советских блочных шифров. Он разрабатывался на заре блочного шифрования в недрах спецслужб Советского Союза. Постепенно шифр был рассекречен. Было ли

причиной тому раскрытие его иностранными разведками, или на смену ему повсеместно пришел другой более современный алгоритм — широкой общественности неизвестно. Как бы то ни было, в 1989 году шифр был принят государственным стандартом блочного шифрования и к нему получили доступ многие специалисты по защите информации. В течение долгого времени данный алгоритм был единственной крипторазработкой, удостоившейся "ГОСТирования". Вследствие этого за ним закрепилось "безциферное" имя "ГОСТ", под которым он известен и за рубежом (*GOST*). Алгоритм не запатентован, что делает его привлекательным для применения в коммерческих приложениях (в том числе за границей).

ГОСТ 28147-89 оперирует 64-битным блоком данных, шифруя его с помощью 256-битного ключа. Структура криптоалгоритма предельно проста — шифр реализует классическую сеть Файштеля из 32 раундов с двумя ветвями (рис. 17.22). Образующая функция алгоритма приведена на рис. 17.23.

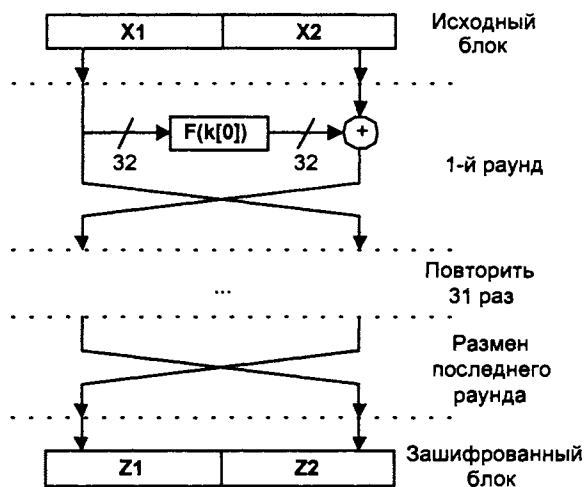


Рис. 17.22. Общий вид ГОСТ 28147-89

О табличных подстановках ГОСТа (в терминологии стандарта они именуются *узлы замен*) необходимо поговорить отдельно. Они — главный камень преткновения и причина давних споров. Дело в том, что в первоначальной реализации криптоалгоритма $8 \times 16 \times 4 = 512$ бит таблиц подстановок являлись также частью секретной информации. Можно ли называть эту информацию частью ключа? Возможно несколько вариантов применения информации для табличных подстановок — они могут быть едиными (однократно созданными) для всего ведомства, а могут быть созданы уникально для каждой пары приборов, обменивающихся информацией. Возможен и вариант однократного создания таблиц замен ради одного сеанса связи — вот в этом случае таблицы подстановок становятся полноправной частью ключевой информации.

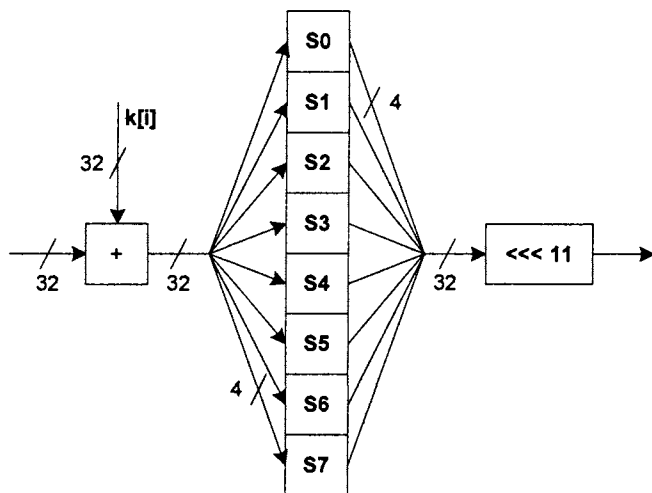


Рис. 17.23. Образующая функция ГОСТ 28147-89

Определив алгоритм подобным образом, авторы добились очень высокой потенциальной гибкости шифра и структуры ключей. Проблема возникла тогда, когда использовать шифр захотели многочисленные невоенные и даже негосударственные структуры. В тексте ГОСТ 28147-89 табличные подстановки не зафиксированы, более того, не приведено даже возможного их примера. А, следовательно, разработчики встают перед вопросом, где взять эти значения. Очевидно, что именно табличные подстановки вносят основной вклад в стойкость алгоритма: например, при наличии в них сильной линейной корреляции шифр становится совершенно беззащитным перед линейным криптоанализом — других нелинейных криптопримитивов, кроме табличных подстановок, в его структуре просто нет.

Единственным официальным учреждением, имеющим полномочия на создание и выдачу материала табличных подстановок, является ФАПСИ РФ¹. По-видимому, только этой организации доступен полный список правил для создания "стойких", "умеренно стойких" и "слабых" таблиц подстановок. Никаких публичных заявлений относительно правил построения таблиц официальными лицами сделано не было. Все это формирует очень специфическую ситуацию вокруг практического применения данного шифра в негосударственных структурах.

ГОСТ 28147-89: расширение ключа

Расширение ключа производится по очень простой схеме. 256-битный ключ разбивается на восемь 32-битных машинных слов. Они нумеруются с K_0 по K_7 .

¹ ФАПСИ РФ — Федеральное агентство правительственной связи и информации при Президенте Российской Федерации.

32 компоненты материала ключа получаются их применением в следующем порядке:

$$k[0]...k[31] = K_0 K_1 \dots K_6 K_7 K_0 K_1 \dots K_6 K_7 K_0 K_1 \dots K_6 K_7 K_7 K_6 \dots K_1 K_0$$

В последней четвертой секции 8 машинных слов ключа записываются в обратном порядке. Для дешифрования (по свойству симметричной сети Файштеля) используется тот же код, но с обратным порядком ключей раунда:

$$K_0 K_1 \dots K_6 K_7 K_7 K_6 \dots K_1 K_0 K_7 K_6 \dots K_1 K_0 K_7 K_6 \dots K_1 K_0$$

ГОСТ 28147-89: анализ

Криптостойкость данного алгоритма зависит на 95% от таблиц подстановок, поэтому обсуждение данной темы на страницах книги практически бесполезно. Несомненно, что существуют наборы таблиц, при использовании которых ГОСТ 28147-89 на несколько порядков устойчивее к криптоатакам, нежели DES и многие другие современные алгоритмы. Для этого у алгоритма есть все предпосылки: потенциальная емкость ключа в 256 бит, зарекомендовавшая себя структура сети Файштеля, взятое "с запасом" число раундов, не вызывающая нареканий образующая функция. Алгоритм оптимизирован под 32-разрядные процессоры и очень прост в реализации.

Конкурс AES

Цели и условия конкурса

Недостаточная длина ключа DES, его ориентированность на аппаратную реализацию и, как следствие, использование на практике огромного числа малоизвестных криптоалгоритмов побудило правительство Соединенных Штатов к объявлению открытого конкурса на криптостандарт блочного шифрования США начала нового века. Конкурс получил название *AES* (*Advanced Encryption Standard*) — "улучшенный стандарт шифрования" и был проведен под эгидой Национального Института Стандартизации и Техники (*National Institute of Standards and Technology* — *NIST*) в 1997–2000 годах. Лучшие претенденты на звание AES, несомненно, отражают современное состояние дел в открытом блочном шифровании и достойны изучения на страницах данной книги.

На первом этапе конкурса заявить о своей разработке могла любая группа исследователей или компания из любой страны мира. Требования к новому стандарту были чрезвычайно просты:

- ☐ шифр должен быть блочным;
- ☐ шифр должен иметь длину блока, равную 128 битам;
- ☐ шифр должен поддерживать ключи длиной 128, 192 и 256 бит.

К середине 1998 года свои разработки на конкурс представили 15 исследовательских групп. К сожалению, российские криптографы не пожелали участвовать в этом не имеющем аналогов проекте. В течение года организаторы проекта вели переписку с научными центрами всего мира — высказаться "за" или "против" того или иного претендента имел любой желающий. Все письма (как и сами алгоритмы-конкурсанты) сразу же публиковались на сервере NIST. В итоге за год было получено более ста электронных и печатных посланий как от научных центров с мировой известностью, так и от исследователей-специалистов в данной области.

Во второй половине 1999 года NIST представил общественности документ, в котором сравнивались все 15 претендентов и были объявлены 5 финалистов, прошедших во второй этап конкурса. Все критерии, по которым производилось сравнение алгоритмов, были разбиты сотрудниками Института на две категории: криптографические и практические характеристики.

Среди наиболее важных криптографических характеристик специалисты NIST указывали:

- ☐ стойкость к известным на момент проведения конкурса криптографическим атакам;
- ☐ максимально возможное упрощение алгоритма (в большинстве случаев — уменьшение количества раундов), при котором шифр все еще остается стойким к известным криптоатакам;
- ☐ отсутствие слабых ключей;
- ☐ стойкость к криптоатакам, специфичным для применяемых в шифре схем и криптоопераций;
- ☐ отсутствие корреляций между данными или ключом и временем, требуемым на операции шифрования/дешифрования;
- ☐ отсутствие корреляций между данными или ключом и мощностью, потребляемой аппаратным или программным устройством, реализующим шифр;
- ☐ отсутствие необоснованно сложных преобразований и "необъяснимых" констант (что дает некоторый шанс исключить недокументированные свойства шифра, известные только разработчикам);
- ☐ применение в структуре алгоритма традиционных криптосхем (парадигм) и частей уже опубликованных ранее алгоритмов, позволяющее принять во внимание результаты анализа этих схем за гораздо более длительный промежуток, чем год проведения конкурса.

К наиболее важным практическим аспектам жюри отнесло:

- ☐ оптимизацию по скорости выполнения кода (в первую очередь, на 32-, 8- и 16-разрядных архитектурах ЭВМ);
- ☐ активное использование для ускорения вычислений ресурсов оперативной памяти, если таковые имеются в наличии на исполняющем программой компьютере;

- ❑ возможность оптимизации по размеру кода (для хранения в ПЗУ "слабых" микросхем и в других запоминающих устройствах с ограниченными ресурсами);
- ❑ возможность распараллеливания кода (естественно, без потерь в криптостойкости);
- ❑ идентичность кода и времени исполнения процедур шифрования и дешифрования;
- ❑ сложность и затраты времени на процедуру создания материала ключа;
- ❑ характер зависимости скорости работы алгоритма от размера используемого ключа (128, 192 или 256 бит).

На первом этапе специалисты NIST отсеяли следующие 10 блочных шифров в табл. 17.5 они представлены в алфавитном порядке названий).

Таблица 17.5. Претенденты AES, не прошедшие в финал конкурса

Алгоритм	Коллектив авторов (для компаний — страна расположения штаб-квартиры)	Наиболее существенные замечания
CAST-256	<i>Entrust Technologies, Inc. (США)</i>	Очень большой объем таблиц подстановок, невысокая производительность
CRYPTON	<i>Future Systems, Inc. (Южная Корея)</i>	Недостаточно стойкое к криптоатакам количество раундов
DEAL	<i>R.Outerbridge, L.Knudsen</i>	Невысокая производительность, неудачная схема использования 128-битных ключей относительно 192-битных
DFC	<i>CNRS — Centre National pour la Recherche Scientifique (Франция)</i>	Недостаточно стойкое к криптоатакам количество раундов, применение 64-битного умножения (что вызовет сложности при реализации на "слабых" архитектурах и может привести к атакам по потребляемой мощности)
E2	<i>NTT — Nippon Telegraph and Telephone Corporation (Япония)</i>	Использование криптопримитивов, реализованных не на всех архитектурах ЭВМ, очень большой объем таблиц подстановок

Таблица 17.5 (окончание)

Алгоритм	Коллектив авторов (для компаний — страна расположения штаб-квартиры)	Наиболее существенные замечания
FROG	<i>TecApro Internacional S.A</i> (Коста-Рика)	Наличие слабых ключей (вероятность — 2^{-30}), высокие требования к оперативной памяти, очень медленный процесс расширения материала ключа
HPC	<i>R.Schroeppel</i>	Наличие эквивалентных ключей, высокие требования к оперативной памяти, медленный процесс расширения материала ключа
LOKI97	<i>L.Brown, J.Pieprzyk, J.Seberry</i>	Высокие требования к оперативной памяти, возможность криптоатаки на ключ в случае перехвата 2^{56} пар блоков "открытый зашифрованный текст"
MAGENTA	<i>Deutsche Telekom AG</i> (Германия)	Использование криптопримитивов, реализованных не на всех архитектурах ЭВМ, возможность криптоатаки на ключ в случае зашифровки им 2^{64} выбранных пар блоков "открытый/зашифрованный текст"
SAFER+	<i>Cylink Corporation</i> (США)	Использование криптопримитивов, реализованных не на всех архитектурах ЭВМ, возможность криптоатаки класса "встреча посередине"

Алгоритмы пяти участников, прошедших в финал, мы рассмотрим подробнее.

MARS: структура

Алгоритм MARS (<http://www.research.ibm.com/security/mars.html>) разработан специалистами фирмы IBM. Он представляет собой блочный шифр с размером блока, равным 128 битам, и переменной длиной ключа. Схема расширения материала ключа такова, что в качестве него может использоваться последовательность бит произвольной длины от 128 до 1248 бит. Сами авторы отмечают, что такая большая возможная длина ключа выбрана ими не с точки зрения стойкости шифра (очевидно, что барьер в 256 бит будет еще очень долго оставаться недостижимой величиной), а ради удобства использования шифра совместно с другими криптографическими примитивами. Так, например, во многих системах распространения ключей уникальная ключе-

вая информация, получаемая в результате выполнения всех положенных действий, имеет длину в 512, 768 или 1024 бита.

Алгоритм MARS можно классифицировать как среднего уровня сложности в реализации. Он представляет собой традиционную сеть Файштеля 3-го типа с 4 ветвями. Однако для повышения стойкости шифра к известным и, возможно, новым схемам криптоанализа разработчики значительно усложнили дополнительные входное и выходное преобразования сети Файштеля. В результате весь процесс шифрования удобно представить в виде 6 последовательных преобразований.

1. Входное сложение с материалом ключа (первичное "забеливание").
2. Входное перемешивание данных (8 раундов преобразования E_TRANS).
3. Первые 8 раундов сети Файштеля (первая схема наложения).
4. Завершающие 8 раундов сети Файштеля (вторая схема наложения).
5. Выходное перемешивание данных (8 раундов преобразования D_TRANS).
6. Выходное сложение с материалом ключа (выходное "забеливание").

Все шесть преобразований спроектированы авторами практически симметрично. За исключением двух центральных этапов исполнение шифра является почти зеркальным. Подобная структура продиктована желанием защитить шифр от криптоатак абсолютно симметрично как "со стороны" входных данных, так и "со стороны" выходных данных.

Схемы входного сложения с материалом ключа выходного "забеливания" приведены соответственно на рис. 17.24. и 17.25.

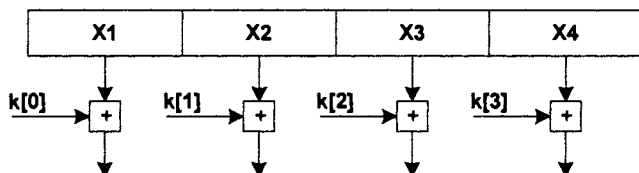


Рис. 17.24. Входное сложение алгоритма MARS

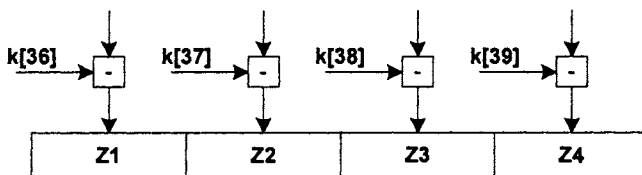


Рис. 17.25. Выходное "забеливание" алгоритма MARS

Схемы входного и выходного перемешивания приведены на рис. 17.26 и 17.27 соответственно. Обе процедуры состоят из 8-кратного повторения типовых схем E_TRANS и D_TRANS, приведенных на рис. 17.28 и 17.29 соответственно. "Разнообразят" данное восьмикратное повторение только операции арифметического сложения (в обратном проходе — вычитания), выполняемые после 1, 2, 5 и 6 применения схем перемешивания. Подобная "нестройность" алгоритма лежит в полном соответствии с общей концепцией авторов сделать шифр легко анализируемым и понятным, но не приводимым к простому аналитическому виду.

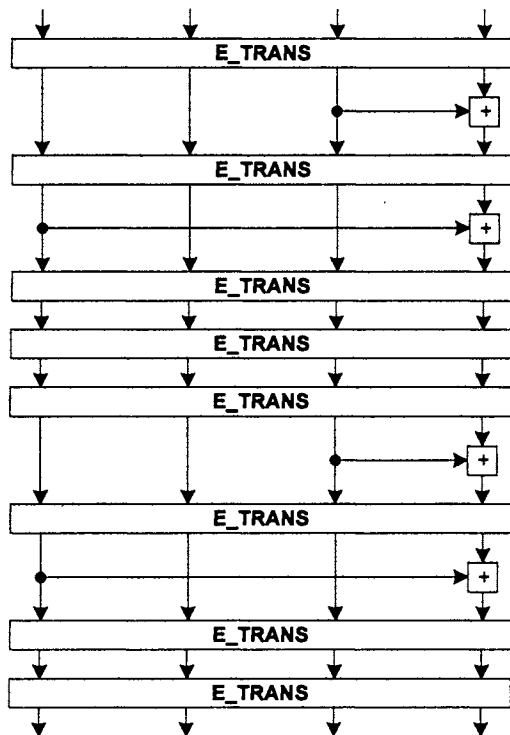


Рис. 17.26. Входное перемешивание MARS

Как видим, на данном этапе полностью отсутствует взаимодействие с материалом ключа. Главная цель перемешивания: подготовка информации к основному циклу криптопреобразований, улучшение статистических характеристик входного блока. Даже в том случае, если на вход криптоалгоритма подаются сильно коррелированные между собой данные, например, ASCII-текст, входное перемешивание (с учетом уже добавленной первой порции материала ключа) выравнивает все статистические характеристики блока. Кроме того, разработчики полагают, что использование в шифре разнородных этапов (перемешивание плюс сеть Файштеля) повышает потенциальную стойкость алгоритма к еще не открытым методам криптоанализа.

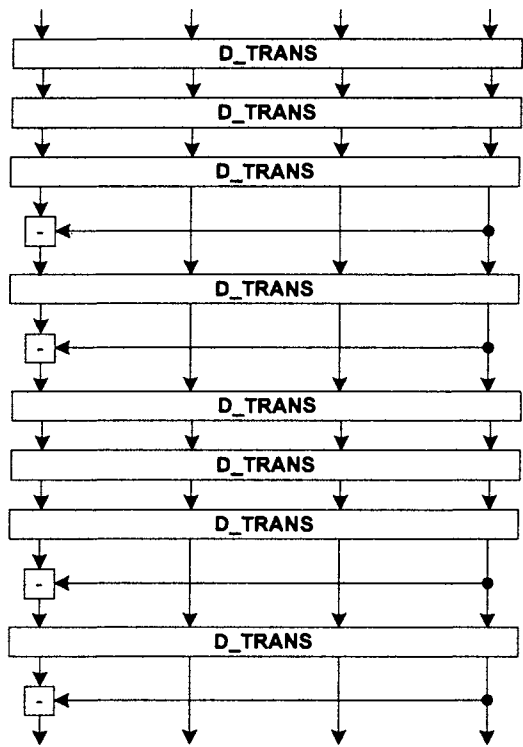


Рис. 17.27. Выходное перемешивание MARS

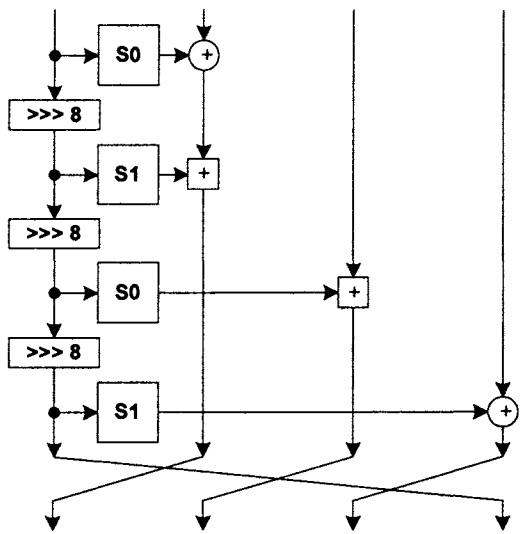


Рис. 17.28. Преобразование E_TRANS алгоритма MARS

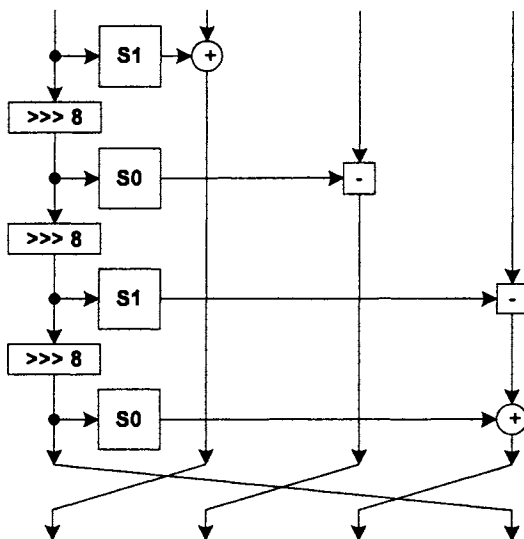


Рис. 17.29. Преобразование D_TRANS алгоритма MARS

Табличные подстановки S_0 и S_1 на этапе перемешивания имеют разрядность 8×32 бита (то есть в качестве индекса в табличной выборке используются только младшие 8 бит из каждой ветви). Эти же две таблицы, но объединенные вместе, образуют табличную подстановку S разрядностью 9×32 бита, используемую в основных раундах сети Файштейля.

Табличная подстановка S из 512 машинных слов каждое разрядностью в 32 бита создана авторами по следующей схеме:

цикл по i от 0 до 102

цикл по j от 0 до 4

$S[5*i+j] = (j - \text{е машинное слово от SHA-1}(i|c_1|c_2|c_3))$

То есть для создания каждого значения в таблице подстановок запускается стандартизованная хэш-функция SHA-1 (см. глава 19) от блока данных, представляющего собой соединение счетчика цикла i и трех констант c_1 , c_2 , c_3 . Из получившегося 160-битного результата хэш-функции берутся только 32 бита, соответствующие состоянию счетчика цикла j .

Выбор констант c_1 , c_2 , c_3 авторы описывают следующим образом: $c_1 = B7E15162_{16}$ (старшие 32 бита двоичной записи дробной части числа $e = 2,71828182845\dots$), $c_2 = 243F6A88_{16}$ (старшие 32 бита двоичной записи дробной части числа $\pi = 3,14159265359\dots$). Константа $c_3 = 02917B59_{16}$ была подобрана простым перебором до тех пор, пока получающаяся табличная подстановка не стала отвечать набору требований, гарантирующих стойкость к линейному и дифференциальному криптоанализу. Подобная схема создания таблиц подстановок (с использованием широко известных констант и небольшой подбираемой части) получила в последнее время широ-

кое распространение. Причина в том, что при достаточно хорошей рандомизации и гарантированной стойкости таблиц она позволяет убедить пользователей в отсутствии каких-либо недокументированных возможностей шифра. Получившаяся в результате исследований табличная подстановка 9×32 приведена в *приложении 5*.

Образующая функция F , используемая в алгоритме во всех 16 раундах сети Файштеля, приведена на рис. 17.30. Как видим, в ней разработчики использовали практически все применяемые на сегодняшний день криптопримитивы: арифметическое сложение, исключающее ИЛИ, 32-разрядное умножение, сдвиги на фиксированное и переменное число бит, табличные подстановки. Подобный "винегрет" авторы снова объясняют повышением потенциальной стойкости алгоритма к еще неизвестным методам криптоатак. Вероятность того, что какая-либо закономерность потока данных, используемая в подобной атаке, пройдет через все виды криптопреобразований, очень мала — уверяют они. Подобные взгляды составляют целую парадигму в современной криптографии в оппозицию принципам "чистых" математических преобразований (которые используют очень ограниченный набор операций с тщательно выбранными свойствами).

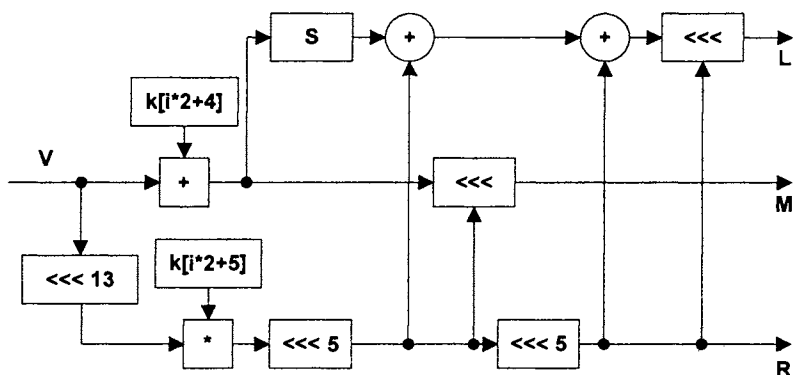


Рис. 17.30. Образующая функция MARS

Схема наложения трех результатов образующей функции на остальные ветви отличается в первых восьми (рис. 17.31) и во вторых восьми (рис. 17.32) раундах сети Файштеля.

Процедура дешифрования отличается от шифрования только заменой арифметического сложения на вычитание в схеме наложения результата образующей функции F на остальные ветви блока и инверсным направлением размена ветвей. Это позволяет при процедурной реализации шифра уместить обратный алгоритм (при наличии готовой "прямой" реализации) всего в несколько десятков байт кода.

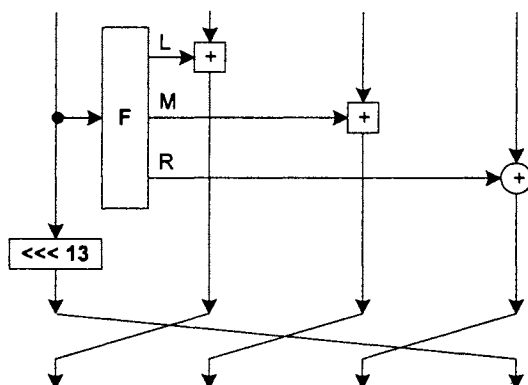


Рис. 17.31. Первая схема наложения MARS

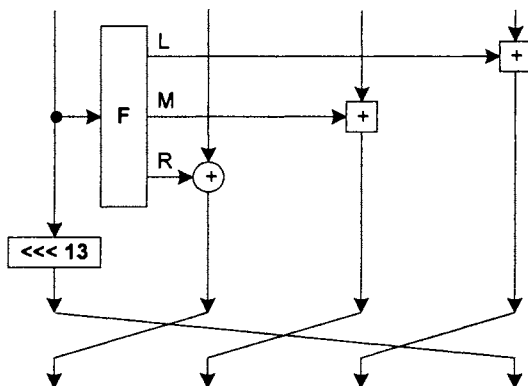


Рис. 17.32. Вторая схема наложения MARS

MARS: расширение ключа

Алгоритм обладает очень сложной процедурой расширения материала ключа, состоящей из 4 этапов.

1. Первоначальное расширение ключа произвольной длины до 1280 бит.
2. Перемешивание материала ключа.
3. Изменение порядка ключевого материала (пересортировка).
4. Проверка и коррекция материала, использующегося в операции 32-битного умножения.

Первый этап расширения создает из ключа K произвольной длины n (измеряется в 32-битных машинных словах) временный массив T из 47 машинных слов по следующей схеме. Массив T имеет нумерацию элементов с -7 по $+39$.

Элементы с -7 по -1 заполняются значениями из табличной подстановки S :

$T[-7]=S[0], \dots, T[-1]=S[7]$

Элементы с 0 по 38 заполняются по рекурсивной формуле, использующей как ключ, так и уже заполненные ранее значения T :

$T[i]=((T[i-7] \text{ XOR } T[i-2]) \text{ ROL } 3) \text{ XOR } k[i \bmod n] \text{ XOR } i \quad (i=0..38)$

В последнюю 39-ю ячейку массива T помещается исходная длина ключа n с той целью, чтобы два разных по длине ключа не могли создать одинаковый массив T , и как следствие, одинаковый материал ключа. Подобная ситуация может произойти, например, если первый ключ — 256-битный и состоит из двух одинаковых половинок, совпадающих со вторым 128-битным ключом.

По окончании заполнения массива T первые семь ячеек с отрицательными номерами отбрасываются. Их предназначение было только в исполнении рекурсивной формулы для малых значений счетчика i . В дальнейшем они использоваться не будут.

На втором этапе (перемешивании) весь массив T семикратно преобразуется по рекурсивной "волноподобной" формуле, очень напоминающей некоторые хэш-функции:

```
цикл по j от 0 до 7
  начало_j
  цикл по i от 1 до 39
    начало_i
     $T[i]=(T[i]+S[LOW\_9(T[i-1])]) \text{ ROL } 9$ 
     $T[0]=(T[0]+S[LOW\_9(T[39])]) \text{ ROL } 9$ 
    конец_i
  конец_j
```

Функция LOW_9 означает взятие только младших 9 бит результата (напомним, что табличная подстановка S имеет 9 бит на входе и 32 бита на выходе).

На третьем этапе производится перестановка, расставляющая далеко друг от друга машинные слова, находившиеся рядом в массиве T . В результате выполнения данного этапа создается массив материала ключа k из 40 32-битных машинных слов:

$k[(7*i) \bmod 40]=T[i] \quad (i=0..39)$

Четвертый (один из самых сложных) этапов расширения ключа отвечает за то, чтобы на нечетных позициях в массиве k не оказались "слабые" с точки зрения операции умножения числа. Преобразованию и проверке подвергаются все нечетные ячейки массива k с 5-й по 35-ю включительно. Первоначально два младших бита у каждого подобного элемента устанавливаются в "1" ($k[i] = k[i] \text{ OR } 3$), а их исходное значение (возможные значения 0, 1, 2, 3) запоминается перед этим в переменной $L[i]$ (оно потребуется на следующем

этапе). Это обеспечивает хотя бы минимально приемлемое перемешивание нескольких младших бит основного (неключевого) сомножителя в результате умножения.

Затем для каждого проверяемого элемента $k[i]$ создается битовая маска $M[i]$, в которой единицами отмечены те места в элементе $k[i]$, где идут подряд 10 или больше "нулей" или "единиц". Такие места в ключевом сомножителе $k[i]$ приведут к тому, что корреляция между некоторыми битами неключевого сомножителя и результатом умножения будет неприемлемо высока. В полученной маске $M[i]$ сбрасываются в "0" два младших бита, а также те биты, которые стояли на границе (в начале и в конце) длинных последовательностей нулей или единиц, например:

Ключевой элемент	$k[i]$	0101111111111110110000000001101
Первоначальная маска	$M[i]$	0001111111111110001111111110000
Маска после сброса бит	$M[i]$	000011111111111000001111111100000

Наконец, производится собственно коррекция элемента $k[i]$ по следующей формуле:

$$k[i] = k[i] \text{ XOR } (S[265+L[i]] \text{ ROL } LOW\_5(K[i+3]))$$

Функция LOW_5 означает взятие младших 5 бит от числа. Элементы табличной подстановки с номерами 265...268 (напомним, что $L[i]$ может принимать только значения 0, 1, 2, 3) выбраны авторами как обладающие наиболее удачными для подобной коррекции вероятностными характеристиками.

MARS: анализ

Основные преимущества алгоритма по сравнению с другими претендентами заключаются в специфичной процедуре расширения ключа и в парадигме "множество операций". Используемое расширение ключа позволяет, во-первых, применять алгоритм с ключом произвольной длины от 32 до 1248 бит (с шагом в 32 бита). Во-вторых, мощное криптографическое преобразование еще на этапе расширения материала ключа (по стойкости оно не уступает некоторым хэш-функциями) позволяет использовать в качестве ключа даже сильно коррелированные между собой данные, например, текст в формате ASCII (пароль — до 156 символов) или Unicode (до 78 символов). Правда, использование пароля непосредственно в качестве ключа не считается в криптографии хорошим тоном, но создатели алгоритма MARS похоже предусматривают и такую схему работы алгоритма.

Использование в самом ядре криптографического преобразования множества разноплановых арифметических и логических операций действительно с большой вероятностью поможет выстоять алгоритму против новых еще не открытых схем криптоанализа. Если, конечно, не будет реализован какой-либо качественный прорыв в криптотехнологиях.

К недостаткам алгоритма MARS исследователи относят достаточно большой размер табличных подстановок ($512 \times 4 = 2048$ байт), что может сказаться на реализации в системах с ограниченными ресурсами ПЗУ. Процедура расширения ключа и 32-разрядное умножение также достаточно сложны для слабых микрочипов, подобных смарт-картам и контроллерам. Из-за применения умножения и сдвигов на переменное число бит шифр потенциально нестойк к атакам по потребляемой мощности и времени исполнения. Алгоритм слабо поддается распараллеливанию.

RC6: структура

Шифр RC6 (<http://www.rsasecurity.com/rsalabs/rc6/>) представлен группой разработчиков во главе с мировой фигурой в области криптографии Рональдом Ривестом (*Ronald Rivest*). Неофициально, но скорее всего абсолютно верно название шифра, как и предыдущих разработок этого ученого, представляет собой аббревиатуру от Rivest's Cipher — шифр Ривеста. Алгоритм разработан за довольно короткий для криптографии период на основе предыдущего очень популярного блочного шифра RC5. Изменения коснулись длины ключа, требуемой от конкурсантов AES, размера блока и некоторых незначительных моментов в основной части.

Алгоритм RC6 исключительно прост. Его четкая краткая структура и отсутствие табличных подстановок делают его очень привлекательным для реализации как криптографической прослойки в программных продуктах. Шифр представляет собой сеть Файштеля из 20 раундов с 4 ветвями смешанного типа — результат образующих функций, вычисляемых от четных ветвей, накладывается на нечетные ветви, затем происходит обмен ветвей местами. Общий вид алгоритма (с входным и выходным забеливанием) приведен на рис. 17.33, структура одного раунда — на рис. 17.34.

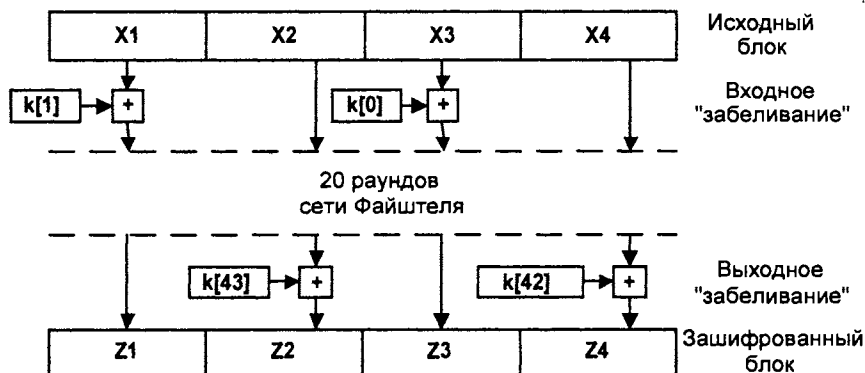


Рис. 17.33. Общий вид алгоритма RC6

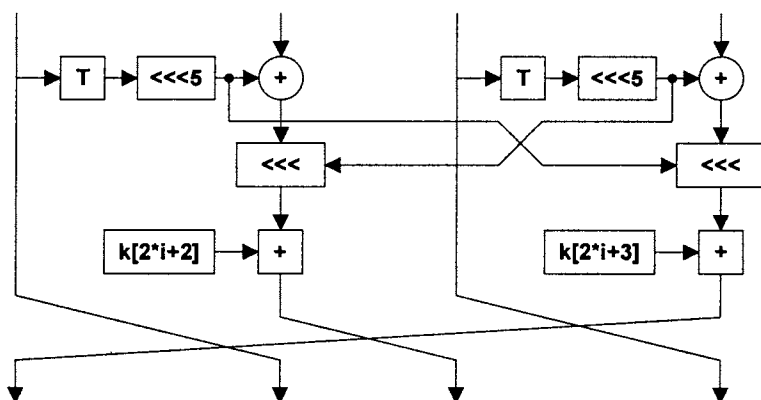


Рис. 17.34. Раунд сети Файштеля алгоритма RC6

Преобразование T представляет собой функцию

$$T(X) = \text{LOW\_32} (X \times (2 \times X + 1))$$

с 32-разрядным входом X и 32-разрядным результатом (возвращаются только 32 младших бита от результата умножения).

Операция умножения обладает хорошими перемешивающими свойствами, и в RC6 это свойство используется на максимальном уровне. Как видно из схемы, в качестве параметра циклического сдвига на переменное число бит поступает результат умножения, циклически смещенный влево на 5 бит, т. е. реально величину переменного сдвига определяют 27...31-й биты результата 32-битного умножения — а именно эти биты находятся в самом центре 64-битного произведения, и, следовательно, зависят от всех бит входного параметра X .

Необходимо отметить и такую особенность преобразования T как биективность, т. е. обратимость — взаимно однозначное соответствие между операндом и результатом. Это довольно неочевидный и редкий факт для операции умножения, да еще и с отсечением некоторой части результата. Однако, при некоторых наборах констант (и в данном случае тоже) подобная операция биективна, и следующее за ней наложение результата умножения операцией XOR на изменяемую ветвь "переносит" на нее все 32-бита информационной емкости неизменяемой ветви. Применение в качестве образующей функции сети Файштеля именно обратимого преобразования хотя и не обязательно, но считается хорошим тоном в криптографии.

Дешифрование алгоритма RC6 производится инвертированием порядка выполняемых в сети Файштеля действий и заменой операции сложения на вычитание. Из-за смешанного типа применяемой сети и простоты алгоритма в целом дешифрование RC6 проще реализовывать отдельным блоком программного кода.

RC6: расширение ключа

Алгоритм расширения ключа в RC6 сравним по простоте и скорости исполнения с основным преобразованием. Первоначально ключ (который теоретически может иметь произвольную длину) выравнивается по 32-битной границе нулями. Пусть полученный в результате массив K имеет длину в s 32-битных машинных слов и нумеруется соответственно с $K[0]$ по $K[s-1]$. Массив расширенного материала ключа назовем традиционно k , для 20 раундов и забеливания потребуется 44 элемента: $k[0]...k[43]$. В процедуре расширения используются две константы $P = B7E15163_{16}$ и $Q = 9E3779B9_{16}$, представляющие собой старшие 32 бита двоичной записи дробной части чисел $e = 2,71828182845...$ (основание натурального логарифма) и $\phi = 1,61803398875...$ (золотое сечение) соответственно.

На первом этапе массив k заполняется константами по формуле:

$$k[i] = P + i \times Q \quad (i = 0...43)$$

На втором этапе производится рекурсивное перемешивание между собой массивов K и k с помощью двух вспомогательных 32-битных переменных A и B . Алгоритм перемешивания выглядит следующим образом:

$A=0$; $B=0$; $i=0$; $j=0$

цикл по x от 1 до 44×3

 начало_х

$k[i] = (k[i] + A + B) \text{ ROL } 3$; $B = k[i]$;

$i = (i + 1) \bmod 44$

$K[j] = (K[j] + A + B) \text{ ROL } (A + B)$; $A = K[j]$

$j = (j + 1) \bmod c$

 конец_х

Схематично данный алгоритм можно изобразить, как показано на рис. 17.35, если представить, что массивы K и k замкнуты в виде колец (первое с периодом оборота c , второе — с периодом 44). Как видим, процедура перемешивания алгоритмически очень проста — в ней используются только арифметическое сложение и циклические сдвиги. Однако с точки зрения криптоанализа подобное трехкратное рекурсивное перемешивание материала расширения с ключом (а главное, при наличии сдвигов на переменное число бит) очень трудно анализируется.

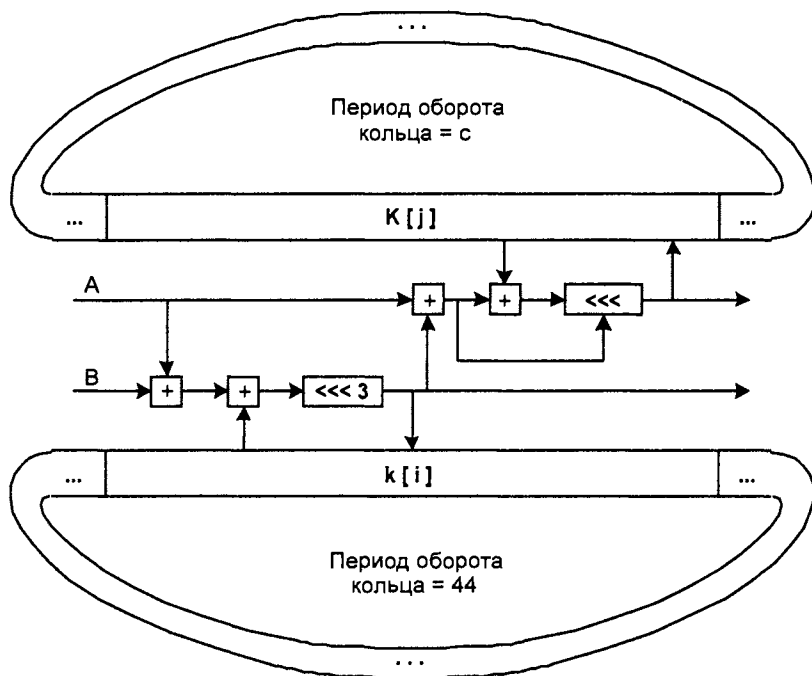


Рис. 17.35. Раунд расширения ключа RC6

RC6: анализ

Основными преимуществами RC6 над другими претендентами исследователи единодушно называли очень высокую скорость работы шифра на 32-разрядных процессорах, при том что криптостойкость алгоритма практически "гарантируется" его предшественником — RC5 (прошедшим гораздо больший период безуспешных попыток взлома, чем большинство других претендентов). Сочетание этих двух факторов формирует для RC6 очень высокий коэффициент "стойкость/быстродействие", а он, как уже упоминалось, является основополагающим в криптографии. Среди других преимуществ называется сверхмалый объем кода как основной части алгоритма, так и процедуры расширения ключа.

К потенциальным недостаткам шифра следует отнести использование 32-разрядного умножения, сложного для 8-битных архитектур, и слабую распараллеливаемость алгоритма.

Rijndael: структура

Данный блочный шифр (<http://www.esat.kuleuven.ac.be/~rijmen/rijndael/>) развивает нетрадиционную для последнего десятилетия структуру: прямо-

угольные KASLT-сети. Эта структура получила свое название из-за того, что шифруемый блок представляется в виде прямоугольника, например, 4×4 или 4×6 байт, и затем над ним построчно, по столбцам и побайтно производятся криптопреобразования. KASLT представляет собой аббревиатуру английских терминов *Key Addition* (добавление ключа), *Substitution* (табличная подстановка) и *Linear Transposition* (линейное перемешивание).

Rijndael (допускаются различные варианты произношения, например, *англ.* "Ра'йндэл", *интерн.* "Рийнда'эл") описан авторами (бельгийцами *Vincent Rijmen* и *Joan Daemen*) как шифр с вариативными размером блока и длиной ключа. Единственные налагаемые требования на эти два параметра — их кратность 32 битам. Размер блока далее определяется как $Nb \times 32$, длина ключа — $Nk \times 32$. Количество раундов сети, необходимое для стойкого шифрования, зависит от этих двух параметров. Для конкурса AES в соответствии с требованиями авторы зафиксировали $Nb = 4$ (то есть шифруемый блок представляет собой квадрат 4 на 4 байта) и $Nk = 4, 6$ и 8 . В этом случае количество раундов сети Nr соответственно равно 10, 12 и 14. Потенциально шифр может работать и с большими по размеру блоками.

Перед началом раундов KASLT-сети на блок накладывается операцией *XOR* первая порция материала ключа. Все раунды, кроме последнего, состоят из повторения четырех операций: табличной подстановки, циклического сдвига строк, линейного перемешивания столбцов и добавления материала ключа. В последнем раунде эти операции повторяются в том же порядке, но из них исключено линейное преобразование столбцов.

Наложение материала ключа производится операцией *XOR*. Материал извлекается из массива расширенного ключа k последовательно по четыре 32-битных блока на каждом раунде. При наложении ключа прямоугольная матрица рассматривается как 4 столбца с 32-битными записями в каждом (рис. 17.36).

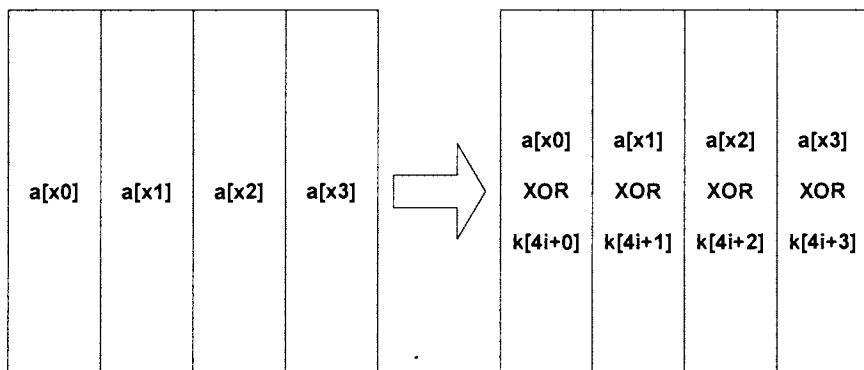


Рис. 17.36. Наложение ключа в Rijndael

Табличная подстановка 8×8 бит производится с помощью одной таблицы S над каждым из 16 байт независимо. После подстановки каждый байт помещается на свое исходное место в прямоугольной матрице (рис. 17.37). Подстановка S имеет алгоритмическую запись, однако в большинстве случаев ее удобнее хранить уже вычисленной в виде массива из 256 байт. Полностью таблица S и ее инверсия приведены в *приложении 6*.

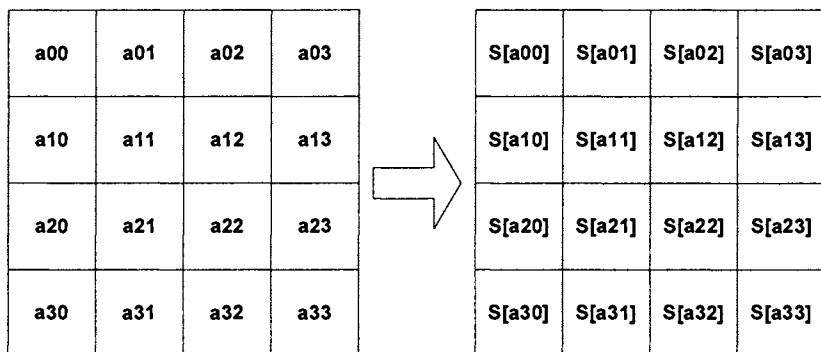


Рис. 17.37. Табличная подстановка в Rijndael

Циклический сдвиг строк выполняет функцию горизонтального распространения изменений в шифруемом блоке. Нулевая (самая верхняя) строка матрицы остается без изменений. Первая циклически сдвигается на 1 байт вправо — "вытолкнутый" за границу таблицы самый правый байт помещается на пустое место в самом начале строки. Вторая строка сдвигается циклически вправо на два байта, третья (самая нижняя) — на три байта (рис. 17.38).

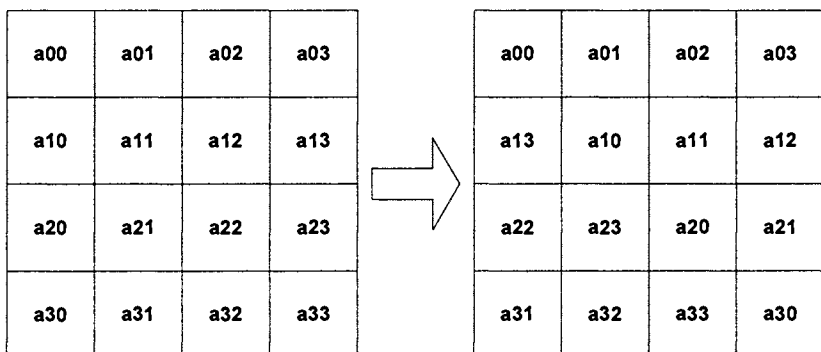


Рис. 17.38. Сдвиг строк в Rijndael

Линейное перемешивание (рис. 17.39) внутри столбцов производится с помощью следующего математического аппарата. Столбец представляется в виде полинома четвертой степени с коэффициентами из диапазона $0 \dots 255$,

затем производится его умножение на фиксированный полином $C(x) = 3x^3 + x^2 + x + 2$ по модулю полинома $(x^4 + 1)$. В привычном алгоритмическом виде это выглядит следующим образом ($a0, a1, a2, a3$ — байты столбца сверху вниз):

$$a0' = (a0*02) \text{ XOR } (a1*03) \text{ XOR } (a2*01) \text{ XOR } (a3*01)$$

$$a1' = (a0*01) \text{ XOR } (a1*02) \text{ XOR } (a2*03) \text{ XOR } (a3*01)$$

$$a2' = (a0*01) \text{ XOR } (a1*01) \text{ XOR } (a2*02) \text{ XOR } (a3*03)$$

$$a3' = (a0*03) \text{ XOR } (a1*01) \text{ XOR } (a2*01) \text{ XOR } (a3*02)$$

Умножение двух байт, записанное символом "*", отличается от привычного арифметического, — это побитное умножение в полиномиальной форме по модулю неприводимого полинома. Образующим неприводимым полиномом авторы выбрали $M(x) = x^8 + x^4 + x^3 + x + 1$. Так, например, умножение двух чисел $65_{16} * 4A_{16}$ будет выглядеть следующим образом:

$$65_{16} * 4A_{16} = (01100101_2) * (01001010_2) =$$

$$= ((x^6 + x^5 + x^2 + 1) * (x^6 + x^3 + x)) \bmod (x^8 + x^4 + x^3 + x + 1) =$$

$$= (x^{12} + x^{11} + x^8 + x^6 + x^9 + x^8 + x^5 + x^3 + x^7 + x^6 + x^3 + x) \bmod (x^8 + x^4 + x^3 + x + 1) =$$

// коэффициенты вычисляются по модулю 2, поэтому x^8, x^6 и x^3 сокращаются

$$= (x^{12} + x^{11} + x^9 + x^7 + x^5 + x) \bmod (x^8 + x^4 + x^3 + x + 1) =$$

$$= (x^7 + x^6 + x^5 + x^2 + x + 1) =$$

$$= (11100111_2) = E7_{16}$$

Для проведения подобного умножения на архитектурах с достаточным объемом ресурсов используется табличная подстановка, что гораздо быстрее выполнения всех операций поразрядно. На слабых вычислительных мощностях оптимизированная реализация полиномиального умножения двух байт выполняется за 7 сдвигов и максимум 14 операций *XOR* (для конкретных "маленьких" коэффициентов $C(x)$, выбранных в Rijndael, требуется максимум 2 сдвига и 4 *XOR*).

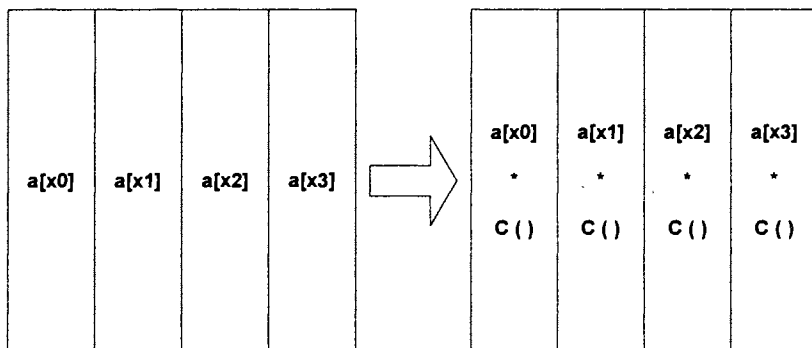


Рис. 17.39. Линейное перемешивание в Rijndael

Благодаря свойствам использованных криптопримитивов полную модификацию одного столбца в раунде Rijndael на ЭВМ, обладающих хотя бы 4 Кбайтами свободной оперативной памяти, можно представить в виде четырех табличных подстановок $T0...T3$ размером 8×32 бита с последующим объединением результатов операций XOR :

$$A_j' = T0[a[0,j]] \text{ XOR } T1[a[1,j-1]] \text{ XOR } T2[a[2,j-2]] \text{ XOR } T3[a[3,j-3]] \text{ XOR } k[j]$$

В табличной подстановке T "спрятаны" табличная подстановка S и линейное перемешивание столбца, выбор коэффициентов элементов $a[...]$ обеспечивает циклические сдвиги строк (рис. 17.40). В результате получается очень эффективная по скорости исполнения реализация. При этом можно отметить, что на современных процессорах все 4 Кбайта таблиц T при интенсивном использовании помещаются в кэше процессора первого уровня..

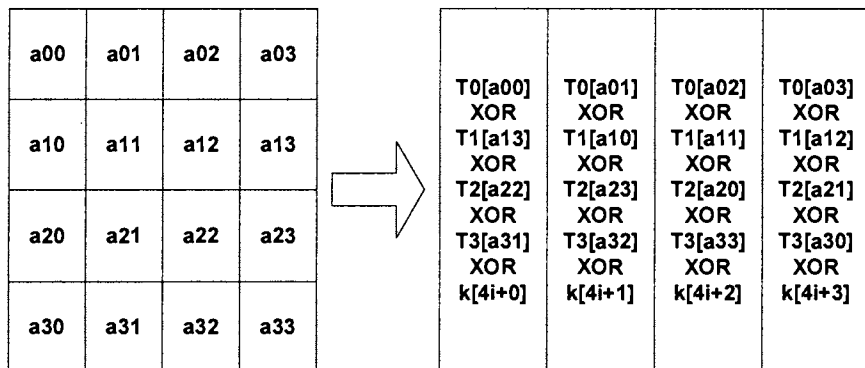


Рис. 17.40. Эффективная реализация раунда Rijndael

Дешифрование производится выполнением в обратном порядке криптопримитивов, инверсных к шифрующим. Наложение материала ключа обратимо самому себе. Сдвиг строк производится циклически в обратную сторону. Табличная подстановка S заменяется на инверсную ей S^{-1} . Полином линейного перемешивания $C(x) = 3x^3 + x^2 + x + 2$ выбран так, что к нему существует инверсный полином $D(x) = 0B_{16} \times x^3 + 0D_{16} \times x^2 + 09_{16} \times x + 0E_{16}$, $C(x) * D(x) = 1$. Таким образом, инверсия линейного перемешивания имеет вид (умножение "*", как и ранее, выполняется побитно в полиномиальной форме):

$$\begin{aligned} a0' &= (a0 * 0E) \text{ XOR } (a1 * 0B) \text{ XOR } (a2 * 0D) \text{ XOR } (a3 * 09) \\ a1' &= (a0 * 09) \text{ XOR } (a1 * 0E) \text{ XOR } (a2 * 0B) \text{ XOR } (a3 * 0D) \\ a2' &= (a0 * 0D) \text{ XOR } (a1 * 09) \text{ XOR } (a2 * 0E) \text{ XOR } (a3 * 0B) \\ a3' &= (a0 * 0B) \text{ XOR } (a1 * 0D) \text{ XOR } (a2 * 09) \text{ XOR } (a3 * 0E) \end{aligned}$$

Rijndael: расширение ключа

Расширение ключа производится по рекурсивной схеме. В первые Nk 32-битных ячеек в массиве материала ключа k заносятся байты ключа K без изменений. Всего необходимо получить $Nb \times (Nr + 1)$ значений в массиве k . Все недостающие элементы вычисляются следующим алгоритмом:

□ при длине ключа, меньшей либо равной 192 битам

$j = Nk$

цикл пока $j < Nb \times (Nr + 1)$

$k[j] = k[j - Nk] \text{ XOR } S(k[j - 1] \text{ ROL } 1) \text{ XOR } R[j / Nk]$

цикл по i от 1 до $Nk - 1$

$k[j + i] = k[j + i - Nk] \text{ XOR } k[j + i - 1]$

$j = j + Nk$

конец_пока

□ при длине ключа, большей 192 бит

$j = Nk$

цикл пока $j < Nb \times (Nr + 1)$

$k[j] = k[j - Nk] \text{ XOR } S(k[j - 1] \text{ ROL } 1) \text{ XOR } R[j / Nk]$

цикл по i от 1 до 3

$k[j + i] = k[j + i - Nk] \text{ XOR } k[j + i - 1]$

$k[j + 4] = k[j + 4 - Nk] \text{ XOR } S(k[j + 3])$

цикл по i от 5 до $Nk - 1$

$k[j + i] = k[j + i - Nk] \text{ XOR } k[j + i - 1]$

$j = j + Nk$

конец_пока

В алгоритме используется та же самая табличная подстановка S , что и в основном блоке шифра. Массив констант R получается по следующей формуле: $R[x] = (r[x], 0, 0, 0)$, т. е. младший байт 32-разрядного числа вычисляется, а три старших байта всегда равны нулю. Константы $r[x]$ получаются рекуррентно выражением $r[0] = 1$, $r[x] = r[x - 1] * 02$ — умножение двух байт производится в полиномиальной форме, также как в этапе линейного перемешивания (то есть начало массива $r[x]$ имеет вид $01_{16}, 02_{16}, 04_{16}, 08_{16}, 10_{16}, 20_{16}, 40_{16}, 80_{16}, 1B_{16}, 36_{16}, 6C_{16}, D8_{16}, AB_{16}$).

Rijndael: анализ

К достоинствам алгоритма необходимо отнести очень хорошее быстродействие на всех платформах от 8- до 64-битных. "Обгоняют" алгоритм по скорости исполнения только Serpent на аппаратной реализации и RC6 на 32-битной платформе — причина этого в очень сильной оптимизации данных двух алгоритмов под конкретные архитектуры ЭВМ.

Структура шифра позволяет использовать его при любых комбинациях размера блока и длины ключа — единственным необходимым изменением будет только смена количества раундов.

Возможность распараллеливания у Rijndael очень высока по сравнению с другими конкурсантами. Коэффициент ускорения вычислений при наличии неограниченного (но разумного) числа процессоров в системе у Rijndael достигает 6–8 раз, что более чем в два раза выше ближайшего соперника.

Несомненно, что все перечисленные преимущества, — заслуга в большой степени архитектуры KASLT по сравнению с ограниченными возможностями сети Файштеля. Однако ни в коем случае нельзя недооценивать и вклад авторов, т. е. конкретную специфику шифра. Во-первых, без оригинального построения операций внутри раунда оптимизация 32-битными табличными подстановками и высокое распараллеливание были бы невозможны. А во-вторых, огромная заслуга авторов в том, что на конкурс был представлен один из первых KASLT-шифров с надежно доказанной криптостойкостью при таком малом количестве раундов, т. е. был найден вариант достижения такого же соотношения "стойкость/быстродействие", что и у традиционной сети Файштеля, но на другой, гораздо более универсальной для вычислительной техники, архитектуре KASLT.

К недостаткам шифра исследователи относят необходимость разработки двух независимых процедур — шифрования и дешифрования (объединить их, как, например, в сети Файштеля, уже невозможно). Кроме того, генерация материала ключа "на лету" при дешифровании гораздо сложнее, чем при шифровании.

Serpent: структура

Алгоритм Serpent (<http://www.cl.cam.ac.uk/~rja14/serpent.html>) представлен на конкурс тремя очень известными исследователями в области криптографии — англичанином Россом Андерсеном (Ross Anderson), ученым из Израиля Эли Бихамом (Eli Biham) и датчанином, работающим в настоящее время в Норвегии, Ларсом Кнудсеном (Lars Knudsen). Шифр является продолжателем семейства SP-сетей, наиболее ярким представителем которого является DES. SP-сети получили свое название от первых букв двух основных слоев, выполняемых поочередно в течение нескольких раундов. Первым слоем является табличная подстановка (англ. *substitution*), второй — перемешивание данных внутри блока (англ. *permutation*). В целом Serpent представляет собой очень консервативный по своей структуре шифр, чего и добивались авторы, так как за десятки лет криптоатак на DES так и не было найдено действительно эффективного метода его взлома. Очень низкую скорость исполнения битовых выборов и перестановок в DES авторы обошли с помощью недавно опубликованной очень интересной технологии *bitslice operating* — послойных битовых операций.

Алгоритм состоит из 32 раундов. Предварительно из ключа произвольной длины от 1 до 256 бит создается 33 блока материала ключа $k[0]...k[32]$, каждый размером в 128 бит. Перед основным ядром алгоритма производится перестановка бит по схеме, приведенной в *приложении 7*. Никакой криптографической нагрузки она не несет.

Циклы с 0 по 30 состоят из:

- ❑ наложения на блок материала ключа $k[i]$ операцией XOR ;
- ❑ 32-х табличных подстановок 4×4 бита;
- ❑ линейного преобразования, действующего на все биты блока.

В алгоритме используется 8 табличных подстановок $S[0]...S[7]$, каждая 4×4 бита (они приведены в *приложении 8*). В нулевом раунде используется 32 раза (над каждой 4-битной группой блока) подстановка $S[0]$, в первом — $S[1]$, ..., в седьмом — $S[7]$, в восьмом — снова $S[0]$ и так далее по циклу. Линейное преобразование задано перечислением бит входного блока до преобразования, от которых зависит каждый конкретный бит блока после преобразования. Например, нулевой бит результата преобразования получается применением операции XOR над 16, 52, 56, 70, 83, 94 и 105-м битами исходного блока, первый бит — применением операции XOR над 72, 114 и 125-м битами и т. д. Полная таблица зависимостей приведена в *приложении 9*.

Последний 31-й раунд отличается тем, что вместо линейного преобразования на результат 31-й табличной подстановки накладывается операцией XOR 32-й блок материала ключа ($k[32]$). Вслед за последним раундом производится перестановка бит, обратная той, которая выполнялась в самом начале шифрования.

Очевидно, что в целом структура и основные криптопримитивы алгоритма Serpent копируют DES, однако в некоторых моментах авторы произвели ряд изменений, которые позволили им наряду с приведенной (очень медленной для современных процессоров) реализацией описать Serpent в виде хорошо оптимизированного под 32-разрядные ЭВМ алгоритма.

В первую очередь оптимизация коснулась табличных подстановок 4×4 бита. Производимая в самом начале алгоритма перестановка бит введена только для того, чтобы, если ее не производить, соответствующие биты всех 4-битных блоков "сошлись" вместе, т. е. нулевые биты всех 4-битных блоков собираются в нулевом машинном слове (с 0 по 31 биты блока), первые биты — в первом машинном слове, и т. д., причем в соответствующем порядке. Выполнение всех 32 табличных подстановок производится одновременно путем вычисления нескольких логических операций над этими четырьмя машинными словами. Подобная технология и носит название *bitslice operating*. Например, нулевая табличная подстановка Serpent

3, 8, 15, 1, 10, 6, 5, 11, 14, 13, 4, 2, 7, 0, 9, 12

вычисляется за 15 битовых операций следующим образом (a, b, c, d — входные биты, e, f, g, h — биты результата, $t1-t14$ — вспомогательные биты):

```

t1  = a   XOR   d;
t2  = a   AND   d;
t3  = c   XOR   t1;
t4  = b   XOR   t3;
h   = t2   XOR   t4;
t6  = b   AND   t1;
t7  = a   XOR   t6;
t8  = c    OR   t7;
g   = t4   XOR   t8;
t10 =      NOT   t3;
t11 = t3   XOR   t7;
t12 = h    AND   t11;
f   = t10 XOR t12;
t14 =      NOT   t7;
e   = t12 XOR t14

```

Все восемь алгоритмов, на которые заменимы табличные подстановки в шифре Serpent, приведены в *приложении 10*.

Вторым компонентом шифра, для которого возможна оптимизированная под 32-разрядные процессоры реализация — это линейное преобразование. Если преобразуемый блок представить в виде четырех машинных слов $X1, X2, X3, X4$, то линейное преобразование можно записать следующим алгоритмом:

```

X1 = X1 ROL 13
X3 = X3 ROL 3
X2 = X2 XOR X1 XOR X3
X4 = X4 XOR X3 XOR (X1 SHL 3)
X2 = X2 ROL 1
X4 = X4 ROL 7
X1 = X1 XOR X2 XOR X4
X3 = X3 XOR X4 XOR (X2 SHL 7)
X1 = X1 ROL 5
X3 = X3 ROL 22

```

Общая схема алгоритма Serpent приведена на рис. 17.41.

Дешифрование алгоритма производится инвертированием порядка операций, использованием материала ключа в обратном порядке и применением обратных табличных подстановок $S'[0]...S'[7]$.

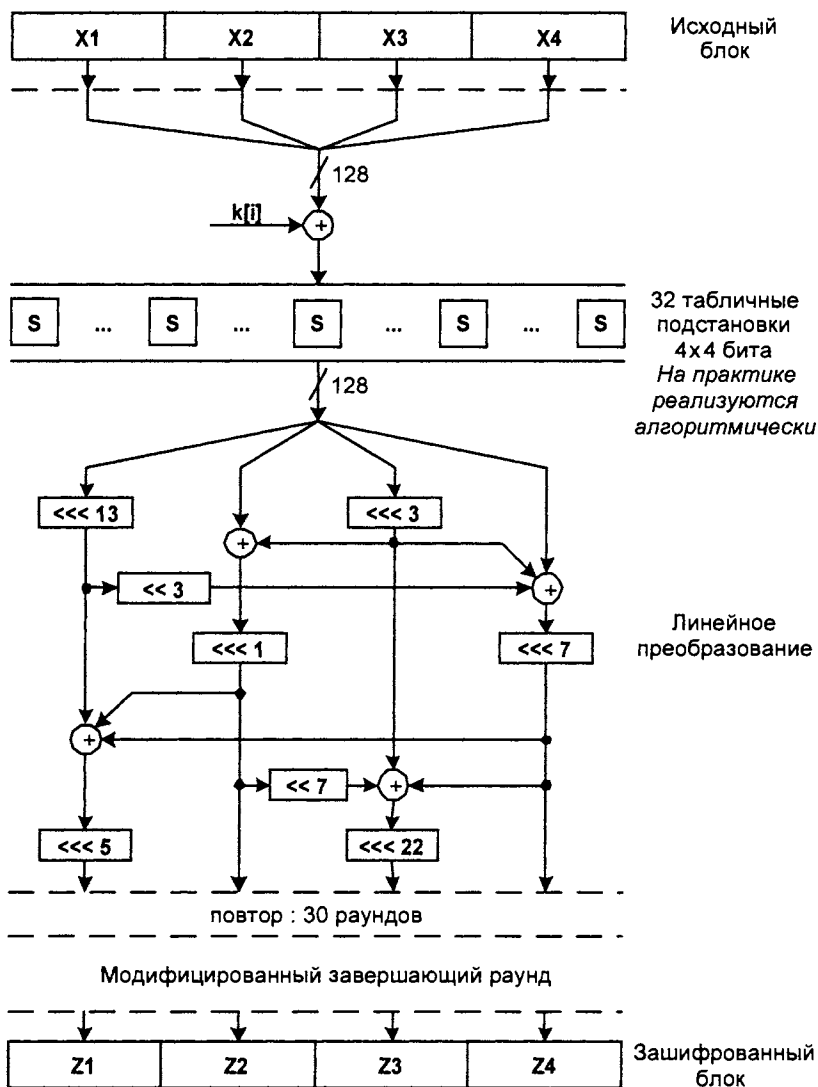


Рис. 17.41. Алгоритм Serpent

Serpent: расширение ключа

Расширение ключа в алгоритме производится в три этапа. Первоначально к ключу произвольной длины добавляется справа бит "1", а затем столько битов "0", сколько требуется до получения 256-битного массива (добавление единицы гарантирует, что никакие два разных неполных ключа не превратятся после расширения в одинаковые битовые массивы).

На втором этапе получившиеся биты записываются в массив 32-битных чисел $W[i]$ под индексами от (-8) до (-1) . Последующие значения массива W с 0-го по 131-й вычисляются по формуле:

$$W[i] = (W[i-8] \text{ XOR } W[i-5] \text{ XOR } W[i-3] \text{ XOR } W[i-1] \text{ XOR } F \text{ XOR } i) \text{ ROL } 11,$$

где $F = 9\text{E}3779\text{B}9_{16}$ — двоичная запись старших бит дробной части золотого сечения — ϕ .

На третьем этапе над 132 машинными словами массива W производится табличная подстановка (опять же в режиме *bitslice operating*). Подряд стоящие блоки по 4 машинных слова и составляют одно 128-битное значение для материала ключа. Таблицы подстановок используются, начиная с 3-й в обратном порядке:

$$k[0] = S[3](W[0], W[1], W[2], W[3])$$

$$k[1] = S[2](W[4], W[5], W[6], W[7])$$

$$k[2] = S[1](W[8], W[9], W[10], W[11])$$

$$k[3] = S[0](W[12], W[13], W[14], W[15])$$

$$k[4] = S[7](W[16], W[17], W[18], W[19])$$

. . .

$$k[31] = S[4](W[124], W[125], W[126], W[127])$$

$$k[32] = S[3](W[128], W[129], W[130], W[131])$$

Serpent: анализ

Алгоритм имеет самую высокую среди всех алгоритмов, представленных в финале, скорость в полностью аппаратной реализации (на простейших электронных элементах). Это неудивительно, если принять во внимание первый (DES-подобный) вариант его описания. В программной реализации алгоритм использует только логические операции и сдвиги на фиксированное количество бит — это делает его стойким к атакам по потребляемой мощности и времени.

Среди недостатков алгоритма следует назвать самую низкую среди алгоритмов скорость исполнения на 32-разрядных платформах. Несомненно, авторы добились многократного уменьшения времени работы шифра, по сравнению с другими DES-подобными алгоритмами, но все-таки парадигма побитовых операций проигрывает более современным криптопримитивам в скорости исполнения. Вторым недостатком является слабая возможность распараллеливания алгоритма.

TwoFish: структура

Алгоритм TwoFish (<http://www.counterpane.com/twofish.html>), представленный на конкурс AES американской фирмой *Counterpane Systems* во главе с известным криптоаналитиком Брюсом Шнайером (*Bruce Schneier*), представляет собой сеть Файштеля смешанного типа. В каждом из 16 раундов образующая функция F , взятая от первой и второй ветвей, накладывается на третью и четвертую ветви. Наложение производится операцией *XOR*, в дополнение к

классической схеме сети Файштеля на каждом этапе разработчики производят циклический сдвиг третьей ветви вправо, а четвертой ветви влево на 1 бит. После каждого раунда, кроме последнего, производится размен ветвей: меняются между собой ветви первая с третьей, а вторая с четвертой. Общая структура шифра и схема образующей функции приведены на рис. 17.42.

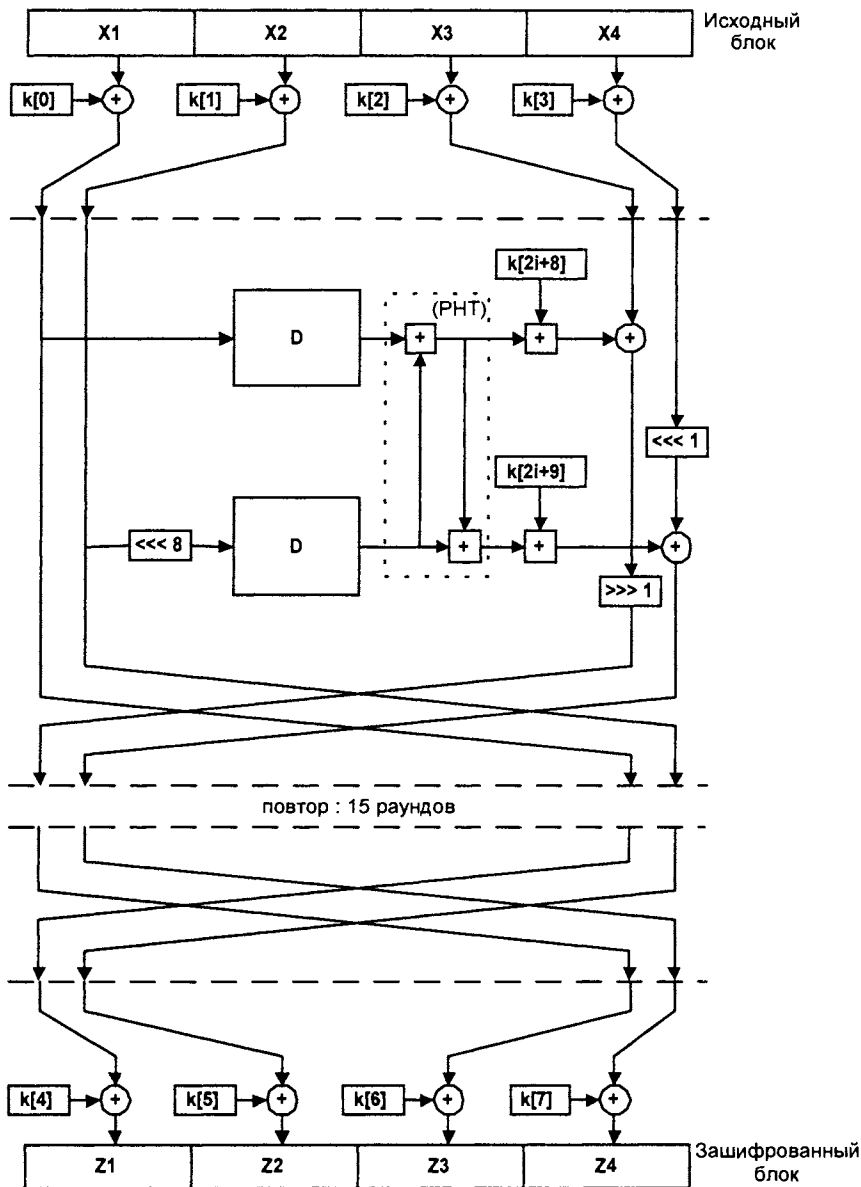


Рис. 17.42. Алгоритм TwoFish

Как видно из схемы, основными перемешивающими элементами являются табличные подстановки 8×8 бит и умножение на матрицу. Оба эти компонента достаточно сложны по своей структуре. Во-первых, в алгоритме используется табличная подстановка, зависящая от ключа (*key-dependent S-box*). Данный термин немного некорректен, хотя и широко используется. Иначе говоря, над данными производится сложное преобразование, зависящее от ключа, которое для ускорения вычислений можно представить в виде табличной подстановки. По причине добавления ключа процесс создания табличных подстановок рассмотрен в разд. "TwoFish: расширение ключа" данной главы.

Второй сложной перемешивающей операцией в TwoFish является умножение на матрицу MDS. Такое название авторы дали ей от сокращения английского термина Maximum Distance Separable, поскольку выбор ее коэффициентов был сделан на основе некоторых идей из теории помехоустойчивого кодирования. Сама матрица представлена на рис. 17.43.

01	EF	5B	5B
5B	EF	EF	01
EF	5B	01	EF
EF	01	EF	5B

Рис. 17.43. Матрица MDS алгоритма TwoFish

Умножение производится в полиномиальной форме по модулю 2 (подробно эта операция рассмотрена в алгоритме Rijndael). В качестве образующего полинома для операции полиномиального умножения выбран $V(x) = x^8 + x^6 + x^5 + x^3 + 1$.

TwoFish: расширение ключа

Алгоритм имеет самую сложную из всех претендентов процедуру расширения ключа. Кроме того, на этапе расширения ключа создаются 4 табличные подстановки S_0, S_1, S_2, S_3 каждая разрядностью 8×8 бит. На начальном этапе ключ произвольной длины дополняется нулями до ближайшей большей длины: 128, 192 или 256 бит. Величина j определяется как $N/64$ (N — длина ключа в битах), т. е. j может принимать значения 2, 3 или 4.

На следующем этапе формируются три вектора длины j из 32-разрядных машинных слов из ключа:

$$M0 = (K[0], K[2], \dots)$$

$$M1 = (K[1], K[3], \dots)$$

Для получения массива S ключ рассматривается уже как массив из j 64-разрядных значений. Каждое из j значений умножается в полиномиальной форме по модулю 2 на трансформирующую матрицу RS разрядности 4×8 байт, созданную на основе кода Риды-Соломона (рис. 17.44):

Полученный после каждого умножения 4-байтный вектор расценивается как 32-битное значение и помещается в массив S . Значения размещаются в массиве S в обратном порядке:

$$S = (\dots, RS*(K[2]K[3]), RS*(K[0]K[1]))$$

Векторы $M0$, $M1$, S составляют базис процедуры расширения ключа.

Основной операцией расширения ключа и одновременно табличной подстановкой, зависящей от ключа, является трансформация D , изображенная на рис. 17.45. Внутри раунда сети Файштеля преобразование D с параметром S используется как зависящая от ключа табличная подстановка.

На этапе расширения ключа это же преобразование D , но с параметрами $M0$ и $M1$ используется для получения четных и нечетных машинных слов массива материала ключа $k[i]$ соответственно:

цикл по i от 0 до 19

начало_i

$$k[2*i] = (D((2*i)*01010101_{16}, M0)*MDS)$$

$$k[2*i+1] = (D((2*i+1)*01010101_{16}, M1)*MDS) \text{ ROL } 8$$

$$k[2*i] = k[2*i] + k[2*i+1]$$

$$k[2*i+1] = (k[2*i+1] + k[2*i]) \text{ ROL } 9$$

конец_i

Преобразования $q0$ и $q1$, используемые в D , являются уже "полноценными", т. е. не зависящими от ключа табличными подстановками 8×8 бит. Их можно привести в табличной форме, однако, памятуя о системах с ограниченными ресурсами для хранения табличных подстановок, авторы выразили их в аналитическом виде через 4 табличные подстановки 4×4 бита — $t0$, $t1$, $t2$, $t3$. Алгоритм получения таблиц $q0$ и $q1$ приведен на рис. 17.44. Табличные подстановки $t0..t3$ для преобразования $q0$ имеют вид:

$$t0 = (8, 1, 7, D, 6, F, 3, 2, 0, B, 5, 9, E, C, A, 4)$$

$$t1 = (E, C, B, 8, 1, 2, 3, 5, F, 4, A, 6, 7, 0, 9, D)$$

$t2 = (B, A, 5, E, 6, D, 9, 0, C, 8, F, 3, 2, 4, 7, 1)$

$t3 = (D, 7, F, 4, 1, 2, 6, E, 9, B, 3, 0, 8, 5, C, A);$

для преобразования $q1$ они другие:

$t0 = (2, 8, B, D, F, 7, 6, E, 3, 1, 9, 4, 0, A, C, 5)$

$t1 = (1, E, 2, B, 4, C, 3, 7, 6, D, A, 5, F, 9, 0, 8)$

$t2 = (4, C, 7, 5, 1, 6, 9, A, 0, E, D, 8, 2, B, 3, F)$

$t3 = (B, 9, 5, 1, C, 3, D, E, 6, 4, 7, F, 2, 0, 8, A).$

01	A4	55	87	5A	58	DB	9E
A4	56	82	F3	1E	C6	68	E5
02	A1	FC	C1	47	AE	3D	19
A4	55	87	5A	58	DB	9E	03

Рис. 17.44. Матрица RS алгоритма TwoFish

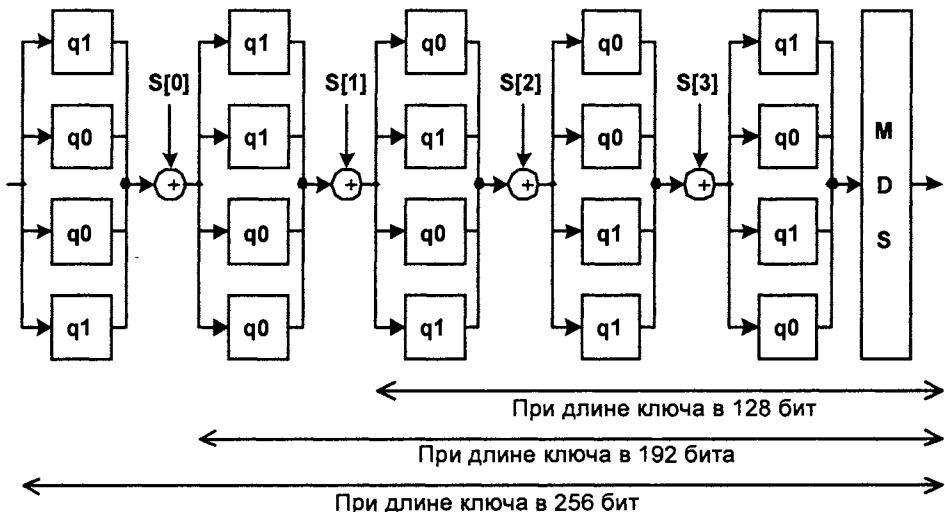


Рис. 17.45. Преобразование D алгоритма TwoFish

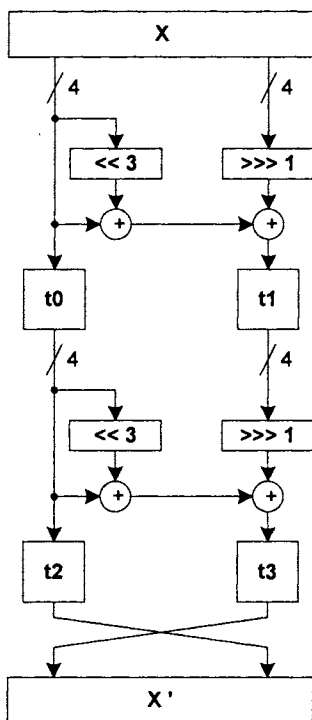


Рис. 17.46. Алгоритм получения таблиц q_0 и q_1 в алгоритме TwoFish

TwoFish: анализ

К преимуществам алгоритма исследователи относят процедуру расширения ключа, не связывающую между собой соседние элементы массива. Это позволяет генерировать материал ключа "на лету" как при шифровании, так и при дешифровании, практически не занимая ресурсов оперативной памяти системы. Среди недостатков называются слабая распараллеливаемость алгоритма, потенциальная уязвимость к атаке по потребляемой мощности, высокая криптосложность и низкая скорость процедуры расширения ключа.

Итоги конкурса AES

По сравнению с первым этапом конкурса, где оценивались в основном качественные характеристики шифров, при выборе победителя во втором этапе основное внимание было уделено уже сравнению количественных характеристик алгоритмов. Преимущества, определившие места на пьедестале, а также количество голосов, отданное за каждый шифр, сведены в табл. 17.6.

Таблица 17.6. Сравнительные характеристики алгоритмов — финалистов AES

Преимущество	Алгоритмы				
	RIJNDAEL	SERPENT	TWOFISH	RC6	MARS
Быстродействие: аппаратная реализация	+	+			
программная реализация: на слабых ВС	+			+	
на мощных ВС		+		+	
Этап расширения ключа	+		+		
Распараллели- ваемость	+				
Этап дешифрования			+	+	+
Запас криптостойкости	Оптимум	Завышен	Завышен	Оптимум	Завышен
Количество голосов	86	59	31	23	13

2 октября 2000 года алгоритм RIJNDAEL был официально оглашен победителем конкурса и получил второе наименование — AES — *Advanced Encryption Standard*.

Основы криптоанализа симметричных шифров



Обычно целью нарушителя в сфере информационной безопасности является получение доступа к информации или управлению системой. Как уже неоднократно упоминалось ранее, путей получить такой доступ у злоумышленника может быть очень много. Общепринято, что целью при "взломе" непосредственно блочного шифра как части защищенной системы является выяснение ключа шифрования. При этом любые другие несанкционированные действия, например, раскрытие части открытого текста, если при этом не был определен ключ, либо фальсификация сообщения считаются взломом уже не самого блочного шифра, а криптосистемы в целом. Здесь ошиб-

ка допущена на этапе взаимодействия блочного шифра с другими компонентами, а не в самом шифре.

Большинство современных методов "взлома" симметричных криптоалгоритмов укладывается в следующую общепринятую классификацию.

Криптоатаки делятся на атаки:

- ❑ По степени наличия информации о ключе:
 - *при полностью неизвестном ключе;*
 - *при частично известном ключе.*
- ❑ По объему и характеру обрабатываемой (перехваченной) информации на атаки:
 - *при неизвестном открытом тексте;*
 - *по известному открытому тексту;*
 - *по выбранному открытому тексту.*
- ❑ По объекту воздействия криптоатаки на атаки:
 - *непосредственно на алгоритм* — "взлом" применим к любой реализации шифра (уязвимость заложена в самой "криптографической" его части);
 - *на конкретную реализацию алгоритма* — успех "взлома" зависит от того, как именно программисты реализовали идеи авторов шифра.

Рассмотрим более подробно криптоатаки по объекту воздействия.

Криптоатаки непосредственно на алгоритмы

Среди множества атак этого класса (как универсальных, так и узкоспецифических — ориентированных на определенные структуру или свойства шифров) мы приведем три наиболее известные. Для них может быть уязвим любой блочный шифр, если при его проектировании не было уделено внимание защите от них.

Метод встречи посередине

Атака *методом встречи посередине* является успешной, если удастся представить процесс шифрования блока $Y = \text{CRYPT}(X, KEY)$ в виде двух независимых алгоритмов CRYPT_1 и CRYPT_2 , оперирующих независимыми половинками ключа KEY_1 и KEY_2 :

$$Y = \text{CRYPT}_2(\text{CRYPT}_1(X, KEY_1), KEY_2), \text{ где } KEY = KEY_1 || KEY_2$$

При этом никаких ограничений на вид функций CRYPT_1 и CRYPT_2 и на то, как именно разбит ключ KEY на две половинки не существует.

Если подобное разделение найдено и у криптоаналитика имеется одна пара $(X^*; Y^*)$ "открытый / зашифрованный текст" (алгоритм "взлома" принадлежит классу атак при известном открытом тексте), то выполняются следующие действия:

- перебирая все варианты KEY_1 , создается массив значений $P = \text{CRYPT}_1(X^*, KEY_1)$;
- перебирая все варианты KEY_2 , вычисляются значения $Q = \text{DECRYPT}_2(Y^*, KEY_2)$.

Равенство $\text{CRYPT}_1(X^*, KEY_1) = \text{DECRYPT}_2(Y^*, KEY_2)$ тождественно выражению $Y^* = \text{CRYPT}_2(\text{CRYPT}_1(X^*, KEY_1), KEY_2)$, т. е. в тех случаях, когда значения P и Q совпадают, пара половинок $(KEY_1; KEY_2)$ является потенциальным искомым ключом KEY . Почему потенциальным? Если размер ключа в битах больше длины блока, то существует довольно много значений ключа, которые превращают X^* в Y^* , но правильным является только один — тот, который использовали отправитель и получатель шифрограммы. Поэтому для однозначного определения ключа необходимо иметь столько бит "открытый / зашифрованный текст", сколько содержится в ключе шифрования (обычно это составляет два, три или четыре блока).

С помощью данной атаки время перебора всех значений ключа сократилось с 2^N (N — разрядность ключа KEY) условных операций, до $2^{N_1} + 2^{N_2}$ (N_1 и N_2 — разрядность половинок KEY_1 и KEY_2 соответственно). В лучшем для злоумышленника случае, когда ключ удалось поделить примерно на равные по длине половинки, формула сложности перебора принимает вид $2^{N/2} + 2^{N/2} = 2^{(N/2)+1}$, т. е. условная длина ключа сократилась примерно вдвое. Например, вместо 2^{64} вариантов перебора 64-битного ключа достаточно произвести 2^{33} операций — а это уже по производительности доступно даже домашнему персональному компьютеру.

Правда, пока не упомянут второй ресурс, очень важный практически для всех криптоатак — оперативная память. Для данной атаки необходимо либо $W \times 2^{N_1}$ (для хранения значений P) либо $W \times 2^{N_2}$ (для хранения значений Q) байт оперативной памяти, где W — размер блока криптоалгоритма. Очевидно, что совершенно безразлично, вычислить ли предварительно массив P и затем по очереди вычислять значения Q , либо, наоборот, хранить все Q , а динамически вычислять P . Поэтому решение принимается исходя из того, которая из половинок KEY_1 или KEY_2 оказалась меньше разрядностью, и, следовательно, потребует меньше оперативной памяти для хранения либо P , либо Q .

Защитой от атаки методом встречи посередине является тщательная и плановая схема расширения ключа, многократное равномерное использование битов ключа на всех раундах (как начальных, так и завершающих).

Линейный криптоанализ

Атака класса *линейный криптоанализ* впервые опубликована *M. Matsui* в 1993 году. Термин "опубликовано" (а не "открыто") применяется очень часто в данной области в связи с огромной дистанцией между засекреченными разработками и т. н. "гражданской" (общедоступной) криптографией. В этой связи примечателен следующий факт: после "открытия" в 1987 году дифференциального криптоанализа, независимые исследователи убедительно показали, что алгоритм DES, спроектированный за 15 лет до этой публикации, уже содержит в своей структуре элементы, целенаправленно противостоящие данному виду криптоанализа.

Линейный криптоанализ классифицируется как атака по известному открытому тексту. Его основой является попытка выделить в блоке исходного текста X , ключе K и блоке шифротекста Y такие биты, чтобы их сумма по модулю 2 на большом объеме зашифрованной информации имела неравномерное распределение. То есть прослеживалось *отклонение* E (англ. *bias*) от закономерных вероятностей $p(0) = 0,5$; $p(1) = 0,5$. Набор выбранных бит может быть самый произвольный, например,

$$p((X_3 \text{ XOR } X_{20} \text{ XOR } K_{12} \text{ XOR } K_{23} \text{ XOR } K_{40} \text{ XOR } Y_{10} \text{ XOR } Y_{37}) = 0) = 0.5 \pm E$$

(подстрочные символы означают номер бита).

При обнаружении подобной зависимости криптоаналитику требуется перехватить порядка $(1/E)^2$ пар "открытый / зашифрованный текст", чтобы с определенной достоверностью выяснить, равна ли для данного ключа выявленная линейная зависимость "0" или "1". Таким образом, во-первых, одна линейная зависимость раскрывает злоумышленнику один бит ключа (явно, либо чаще, в виде связывающей биты формулы) — для успешной атаки потребуется несколько десятков таких зависимостей. Во-вторых, чем больше по модулю отклонение E , тем меньше требуется перехватить информации для раскрытия бита ключа.

Защитой от линейного криптоанализа является активное внедрение в криптоалгоритм нелинейных криптопримитивов, тщательная проверка последовательности выполняемых операций внутри раунда и между раундами на предмет "просачивания" линейных зависимостей.

Дифференциальный криптоанализ

Атака класса *дифференциальный криптоанализ* впервые была опубликована *Eli Biham* и *Adi Shamir* в 1991 году. Данный метод требует наличия у взломщика возможности зашифровать на искомом ключе любое, даже абсолютно бессмысленное, сообщение, т. е. является атакой по выбранному открытому тексту. Алгоритм требует поиска такого изменения входного блока данных, чтобы после шифрования вероятность изменения выходных данных имела закон распределения, отличный от равномерного. Изменение представляет

собой либо инвертирование небольшого количества бит входного блока, т. е. операцию *XOR* с некоторой маской — оно наиболее эффективно, когда материал ключа накладывается на шифруемые данные внутри раунда операцией *XOR*. Либо изменение имеет вид сложения входного блока с некоторой константой — такая модификация метода чаще приводит к успешной криптоатаке, если материал ключа смешивается внутри раунда сложением или умножением.

Если подобное изменение найдено, то путем шифрования и последующего анализа специальным образом подбираемых значений криптоаналитик может в итоге построить систему из достаточного количества линейных уравнений, описывающих материал ключа. Алгоритм подбора шифруемых значений очень сильно варьируется в зависимости от атакуемого шифра. Целью всегда ставится предугадать или получить какую-либо информацию о промежуточных значениях данных внутри раундов шифрования. Правильно подобранные табличные подстановки, размножающие изменение любого входного бита равновероятно на половину своих выходных битов, сводят на нет вероятность дифференциальной криптоатаки уже после 5—8 раундов сети. Такого же эффекта можно добиться алгоритмическими или аналитическими нелинейными примитивами, добавляя их в каждый раунд блочного шифра.

Криптоатаки на реализации алгоритмов

Атаки на реализации алгоритмов зависят также только лишь от их структуры, т. е. повод к ним дают какие-либо особенности шифра, однако при правильно продуманной реализации их можно избежать. Этим они отличаются от перечисленных в *разд. "Криптоатаки непосредственно на алгоритмы"* данной главы от универсальных "математических" атак. И конечно же, явные ошибки (опечатки) в программировании алгоритмов не являются объектом данного параграфа.

Большая часть изложенного далее материала в англоязычной литературе попадает под термин "*side-channel attacks*", вольный перевод — *утечки по неосновным каналам*. Этим указывается на то, что атаки на реализацию алгоритмов обычно получают кроме исходного и/или зашифрованного текста дополнительную информацию о процессе шифрования из третьих источников.

Атака по времени исполнения

Атака по времени исполнения криптоалгоритма основывается на возможности злоумышленника замерять с очень высокой точностью время шифрования или дешифрования блока. В том случае если в шифре используются умножение, сдвиги на зависящее от ключа число бит или, забегая вперед, возведение в высокую степень (это основная операция асимметричной криптографии), становится возможным из замеров времени получить некоторую информацию о материале ключа. В различных разновидностях эта

технология может применяться в классе атака при неизвестном открытом тексте, либо в классе атака с выбранным открытым текстом, в одиночку или в сочетании с рассмотренной далее атакой по потребляемой мощности.

Одна из нашумевших криптоатак с помощью замеров времени связана с шифром IDEA (см. разд. "Эпоха пост-DES" данной главы). В том случае если код алгоритма написан по оптимизированной схеме (умножение на 0, если таковое встречается, не производится вовсе), криптоаналитик может выяснить большую часть материала ключа. Первый вариант атаки может быть произведен по схеме с неизвестным открытым текстом, т. е. при отсутствии возможности зашифровать свои произвольные данные. В этом варианте возможно выяснение до 80 бит ключа при достаточно большом объеме перехваченного зашифрованного трафика (чуть меньше 1 Мбайта) и точных замеров времени его шифрования. Оставшиеся 48 бит ключа определяются перебором. Второй вариант атаки предполагает наличие у злоумышленника возможности шифровать свои данные на искомом ключе с замером времени исполнения. Этот вариант атаки, также определяющий большое количество бит ключа, позволяет замерять время шифрования с гораздо меньшей точностью.

Атака по потребляемой мощности

Атака по потребляемой мощности применима к слабым вычислительным системам, потребляемый ток которых напрямую зависит либо от количества нулей и единиц в оперативной памяти (так называемая *атака по весу Хэмминга*), либо от сложности выполняемых процессором операций. Чрезвычайно эффективно комбинировать данную атаку с замером временных интервалов, что в отношении большого числа реализаций шифров на слабых вычислительных системах дает почти моментальное вскрытие ключа, даже если известен только зашифрованный текст. Реальные взломы, основанные на такой схеме, были продемонстрированы криптоаналитиками в отношении пластиковых карт на микрочипах (смарт-карты). Потребляемый ток и временные интервалы измерялись с помощью специальной накладки, располагавшейся между контактами смарт-карты и считывающим устройством.

Атака по частоте попадания в кэш

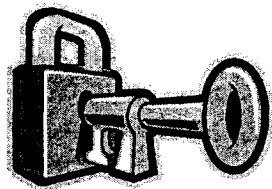
Атака по частоте попадания в кэш современных микропроцессоров была недавно продемонстрирована в отношении блочных шифров, использующих большие табличные подстановки. Одними из наиболее известных подобных шифров являются алгоритмы CAST и BlowFish. Исследователи показали, что по характеру обращения к кэшу оперативной памяти можно определить некоторую часть материала ключа в процессе шифрования блока данных. Анализ работы процессора при шифровании на одном ключе достаточно большого объема данных позволяет построить реальную криптоатаку.

Атаки на отказах

Многочисленные виды *атак на отказах*, предложенные в 1996–1997 годах, основываются на предположении, что злоумышленник может каким-либо образом изменить (инвертировать) один или несколько бит на различных этапах процесса шифрования. Несмотря на специфичность подобных условий, авторы все же продемонстрировали несколько криптосистем, реально применяемых на практике, против которых атака на отказах успешно раскрыла достаточную для взлома часть ключа алгоритма DES.

Подводя итоги атакам на реализацию криптоалгоритмов, необходимо отметить, что на современном этапе развития криптоанализа для построения действительно надежной системы недостаточно только одного "теоретического" уровня проработки алгоритма. Необходимо тесное взаимодействие и многоплановые исследования как со стороны математиков-теоретиков, так и со стороны инженеров-программистов и специалистов по аппаратному обеспечению систем.

Глава 18



Асимметричная криптография и цифровая подпись

Основные принципы



Асимметричная криптография, изобретенная и развившаяся за последние два десятилетия прошлого века, заняла за это время почти такое же положение, как и блочное симметричное шифрование, насчитывающее полувековую историю. Асимметричное шифрование или, как его еще называют, *шифрование на открытом ключе* представляет собой совершенно иную идеологию, уверенно занявшую свою специфическую нишу среди систем защиты информации.

Рассмотренная в *главе 17* симметричная криптография обладает, казалось бы, очень простой и понятной схемой ключей — отправитель и получатель знают один общий секрет, и третьим лицам он недоступен. Но эта на первый взгляд удобная схема скрывает в себе подводные камни при значительном увеличении числа абонентов, желающих шифровать взаимную переписку. Ведь если пользователь хочет конфиденциально вести дела с N партнерами, ему необходимо иметь N секретных ключей, а общее количество ключей в системе растет от числа абонентов по формуле $N \times (N - 1) / 2$. Кроме того, появляются вопросы доверенной доставки ключей абонентам и аппелируемости, т. е. ситуации, когда получатель, дешифровав сообщение, может внести в него изменения, а затем сослаться на документ, якобы уже полученный в таком виде.

Асимметричная криптография решает этот вопрос кардинально иначе — в первых, количество ключей сокращается до $2 \times N$. Кроме данного факта асимметричная криптография обладает еще целым рядом свойств, которые сделали ее идеальной технологией для шифрования общения — именно общения — двух удаленных друг от друга абонентов. Асимметричную криптографию применять для конфиденциального хранения информации на своем собственном жестком диске просто невозможно (если не считать опцией хранения возможность восстановления ключей, как в случае с EFS).

Что же предложила эта новая область науки? Каждый абонент, участвующий в общении в едином информационном поле, обладает двумя ключами — *открытым* (известным всем остальным абонентам) и *закрытым* (известным только ему и хранящимся им в секрете). Открытый ключ используется для отправки сообщений: с его помощью на основе специального алгоритма любой желающий может произвести *асимметричное шифрование* — необратимое без знания закрытого ключа преобразование документа. А вот дешифровать сообщение сможет только владелец закрытого ключа, т. е. законный получатель.

Математика, лежащая в основе обоих алгоритмов (как шифрующего, так и дешифрующего), заботится и том, чтобы по известному открытому ключу невозможно было узнать закрытый. Эта операция во всех схемах асимметричного шифрования требует таких же ресурсов процессорного времени, как и попытка непосредственно обратить процесс шифрования конкретного сообщения. Потенциально исходный текст не может прочесть даже отправитель, если он вдруг "забыл", что написал в этом письме день или месяц назад (если, конечно, у него не осталось копии в незашифрованном виде). Схемы ключей симметричной и асимметричной криптографии для сравнения приведены на рис. 18.1 и 18.2.

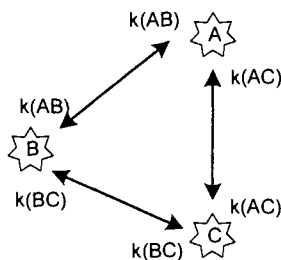


Рис. 18.1. Ключи в симметричной криптографии

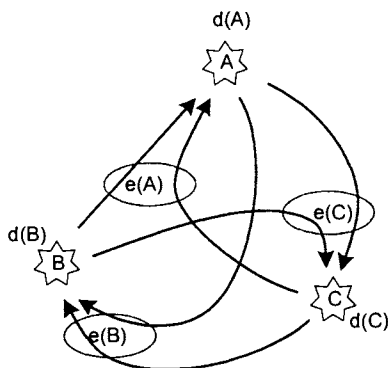


Рис. 18.2. Ключи в асимметричной криптографии

Более того, эта же асимметричная схема, будучи использованной наоборот (когда для шифрования задействован закрытый ключ), с небольшими дополнениями породила еще одну громадную отрасль современной криптографии — *электронную цифровую подпись (ЭЦП)*. При формировании электронной подписи под сообщением, отправляемым неким абонентом в той же единой информационной системе, отправитель *подписывает* послание своим секретным ключом. На самом деле пользователь вычисляет контрольную сумму сообщения, шифрует ее секретным ключом и приклеивает шифрограмму к сообщению, однако глагол "подписывает" очень удачно вписался в новое значение и теперь неотделим от понятия ЭЦП.

При создании электронной подписи под документом в нее закладывается достаточно информации, чтобы любой получатель мог удостовериться с помощью открытого ключа отправителя, что только он мог подписать, а, следовательно, и отправить это сообщение. Но при этом в подписи должно быть недостаточно информации, чтобы извлечь из нее сам секретный ключ отправителя — иначе после первого подписания любой перехвативший письмо мог бы подписывать от имени отправителя свои послания, т. е. технология ЭЦП очень напоминает асимметричный шифр, только наоборот.

Заметим, что асимметричное шифрование и ЭЦП решают совершенно разные задачи: первое — обеспечение конфиденциальности послания, вторая — аутентичность отправителя и целостность сообщения. Поэтому вполне возможно и часто используется шифрование сообщения, а затем его подписание с помощью ЭЦП. В случае симметричной криптографии секретный ключ играл роль и гаранта конфиденциальности и гаранта аутентичности, но за это приходилось платить квадратичным ростом количества ключей в сообществе.

Как же удалось теоретикам добиться такого интересного набора свойств? И здесь на помощь криптографии снова пришла математика. В основе любой схемы асимметричного шифрования либо инверсной ей схемы цифровой подписи лежит определенная трудноразрешимая математическая задача. При этом к данной задаче на самом деле есть способ найти решение, но для этого нужно обладать некоторой дополнительной информацией — в англоязычной литературе ее называют *trapdoor* — *потайная дверь*.

В качестве открытого ключа в асимметричной криптографии выбирается какое-либо частное уравнение, которое и является этой трудноразрешимой для всех задач. Но при составлении этого уравнения оно разрабатывалось так, что лицо, знающее некоторую дополнительную информацию об этом уравнении, может решить его за разумный временной интервал. Эта дополнительная информация и является закрытым ключом. В идеальном случае без закрытого ключа задача должна решаться только за экспоненциальное время работы ЭВМ, а вот при наличии такового — за полиномиальное. Это позволяет выбрать такой размер всех параметров задачи, что процедуры шифрова-

ния или, скажем, подписания документа делятся доли секунды, а вот на взлом их без знания секретного ключа требуются десятилетия.

При собственно асимметричном шифровании получатель сообщений публикует в качестве открытого ключа часть параметров уравнения, которое только он, зная закрытый ключ, сможет разрешить. Отправитель сообщения дробит сообщение на блоки необходимой длины, преобразует их в большие натуральные числа и тем самым заканчивает формирование уравнения. В качестве шифровки посылаются некоторые параметры уравнения или его значения относительно какого-либо вектора, чего недостаточно для злоумышленника, чтобы восстановить исходное послание. Однако их достаточно для получателя, чтобы решить с помощью закрытого ключа уравнение и восстановить добавленный отправителем параметр, который и является исходным текстом.

При формировании ЭЦП процесс идет в несколько ином порядке. Отправитель письма добавляет к письму некоторую часть закрытого ключа в таком виде, чтобы по ней невозможно было полностью восстановить закрытый ключ. Однако, этой информации достаточно, чтобы помочь любому желающему (проверяющему ЭЦП) решить то самое уравнение, на базе которого построена данная схема ЭЦП. В качестве других параметров уравнения проверяющий подставляет контрольную сумму полученного письма и значения из открытого ключа отправителя. Если уравнение успешно разрешилось, т. е. при подстановке всех данных оно превратилось в равенство, значит, письмо с данной контрольной суммой отправил именно тот абонент, чья подпись под ним стоит. Если же равенство не получилось, то на каком-то из этапов произошел сбой и необходимо уже более тщательно выяснять, был ли это случайный сбой на канале связи либо же умышленное увеличение банковского платежа в десять раз.

Рассказывая об открытом ключе, мы умышленно ничего не упомянули о конкретном виде пресловутого уравнения в асимметричной криптографии. Дело в том что, во-первых, оно не так просто, как обычное школьное $a \times x + b \times y = c$, а во-вторых, именно видом уравнения и тем, какие его параметры являются секретными, а какие общеизвестными, и характеризуется та или иная схема асимметричного шифрования. Все они будут рассмотрены в *данной главе*. Необходимо отметить, что задач с подобными специфическими свойствами очень немного.

Отдельное внимание уделим тому факту, что в асимметричной криптографии и ЭЦП ключ является более сложным составным компонентом, нежели просто 128-битный блок данных симметричной криптографии. Чаше всего и открытый и закрытый ключи состоят из двух или трех очень больших (сотни и даже тысячи бит) натуральных чисел, однако встречаются и случаи, когда ключ состоит из сотен натуральных чисел — определенной последовательности или двумерной матрицы. Поэтому не нужно удивляться

тому, что в качестве ключа или, скажем, цифровой подписи далее встретятся наборы чисел — все это неразделимый ключ, имеющий смысл только в сочетании всех его компонент. Не стоит удивляться и их размерам (есть схемы с ключами под сотни килобайт) — просто не очень часто математикам встречаются задачи, которые удовлетворяли бы десятку требований, перечисленных в начале данного раздела, да еще и имели бы небольшое число маленьких по величине параметров-коэффициентов.

Забегая немного вперед, уточним, что на самом деле процедуры асимметричного шифрования еще и очень медлительны. Этот недостаток, правда, успешно нейтрализуется объединением асимметричного шифрования с блочными шифрами. Весь текст сообщения преобразуется обычным блочным шифром (намного более быстрым), но с использованием случайного "только что созданного" *ключа сеанса*, а вот сам ключ сеанса шифруется как раз асимметричным алгоритмом с помощью открытого ключа получателя и помещается в начало шифрограммы. Но подробная информация об этом будет в *главе 20*.

Практические схемы асимметричного шифрования



Схема RSA: структура

Базовые идеи о шифровании сообщений двумя различными ключами были высказаны *W. Diffie* и *M. E. Hellman* в 1976 году. Но практически применимой схемы асимметричного шифрования эти авторы не предложили. Их идеи нашли применение позднее в другой области криптографии — системах управления ключами (см. *главу 20*). А первой реально действующей ассиметричной схемой стал алгоритм RSA, изобретенный тремя исследователями-математиками *Рональдом Ривестом (Ronald Rivest)*, *Ади Шамиром (Adi Shamir)* и *Леонардом Адльманом (Leonard Adleman)* в 1978 году. Схема получила наименование по их инициалам.

Первый этап любого асимметричного алгоритма — создание пары ключей — состоит для схемы RSA из следующих операций.

1. Выбираются два больших простых числа p и q (напомним, что простым называется число, делящееся нацело только на единицу и на само себя).
2. Вычисляется n , равное $(p \times q)$.
3. Выбирается произвольное число e ($e < n$), такое, что наибольший общий делитель $\text{НОД}(e, (p-1) \times (q-1)) = 1$, т. е. e должно быть взаимно простым с числом $(p-1) \times (q-1)$.

4. Методом Евклида решается в целых числах уравнение $e \times d + (p - 1) \times (q - 1) \times y = 1$. Здесь неизвестными являются переменные d и y — метод Евклида как раз и находит множество пар (d, y) , каждая из которых является решением уравнения в целых числах.
5. Пара чисел (e, n) — публикуется как открытый ключ. Число d хранится в строжайшем секрете — это и есть закрытый ключ, который позволит читать все послания, зашифрованные с помощью пары чисел (e, n) .

Второй этап — собственно шифрование с помощью открытого ключа.

1. Отправитель разбивает свое сообщение на блоки, равные $k = \lfloor \log_2(n) \rfloor$ бит, где квадратные скобки обозначают взятие целой части от дробного числа. Подобный блок может быть интерпретирован как число из диапазона $(0; 2^k - 1)$.
2. Для каждого такого числа (назовем его m_i) вычисляется выражение $c_i = (m_i)^e \bmod n$. Блоки c_i и есть зашифрованное сообщение. Их можно без опасения передавать по открытому каналу, поскольку операция возведения в степень по модулю простого числа является той самой трудноразрешимой математической задачей.

Третий этап — дешифрование послания с помощью секретного ключа. Частный случай теоремы Эйлера утверждает, что если число n представимо в виде произведения двух простых чисел p и q , то для любого x имеет место равенство:

$$(x^{(p-1) \times (q-1)}) \bmod n = 1.$$

Для дешифрования RSA-сообщений воспользуемся этой формулой. Возведем обе ее части в степень $(-y)$: $(x^{(-y) \times (p-1) \times (q-1)}) \bmod n = 1^{(-y)} = 1$. После умножения обеих частей равенства на x получим

$$(x^{(-y) \times (p-1) \times (q-1) + 1}) \bmod n = 1 \times x = x.$$

А теперь вспомним, как создавались открытый и закрытый ключи. Величина d была подобрана с помощью алгоритма Евклида так, что $e \times d + (p - 1) \times (q - 1) \times y = 1$, т. е. $e \times d = 1 + (-y) \times (p - 1) \times (q - 1)$. Следовательно, в последнем выражении предыдущего абзаца мы можем заменить показатель степени на число $(e \times d)$. Получаем

$$(x^{(e \times d)}) \bmod n = (x^{(-y) \times (p-1) \times (q-1) + 1}) \bmod n = 1 \times x = x.$$

То есть для того чтобы прочесть сообщение $c_i = ((m_i)^e) \bmod n$, достаточно возвести его в степень d по модулю n :

$$((c_i)^d) \bmod n = ((m_i)^{(e \times d)}) \bmod n = m_i.$$

Общий вид системы приведен на рис. 18.3.

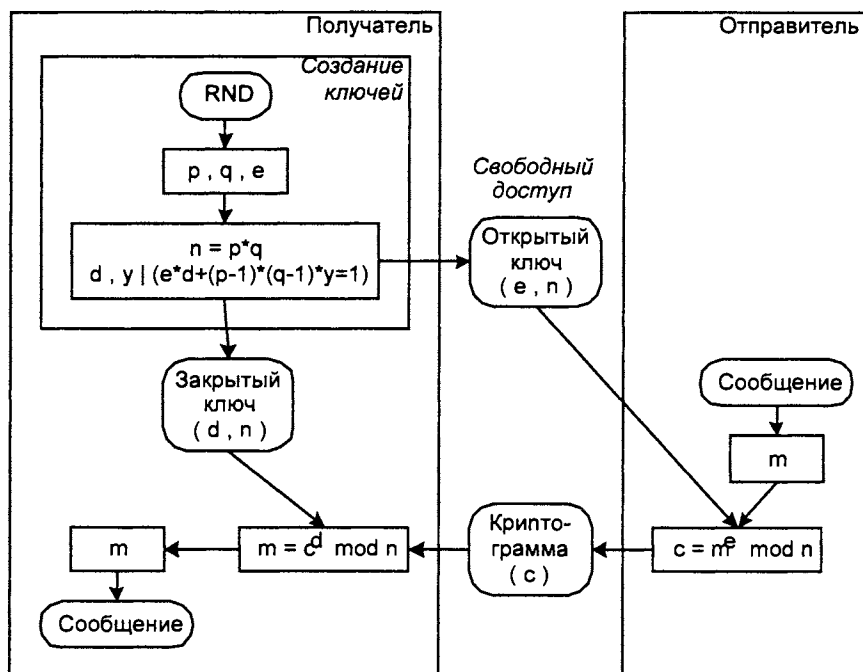


Рис. 18.3. Криптосистема RSA

Схема RSA: анализ

Двумя необходимыми свойствами для любой асимметричной схемы являются:

1. Невозможность прочесть послание, зная только открытый ключ.
2. Невозможность вычислить закрытый ключ по открытому.

Для схемы RSA той самой трудноразрешимой математической задачей, которая стоит на страже первого требования, является проблема *извлечения корней высоких степеней в конечном поле*, т. е., зная в какую степень возводилось число и что получилось в результате, отыскать исходное число очень трудно (обычные методы извлечения корней не работают из-за операции $(\bmod n)$, примененной над результатом).

Второе требование выполняется исходя из вычислительной сложности другой математической задачи — *разложения на множители (факторизации) больших чисел*. Если бы этот процесс занимал полиномиальное время, то злоумышленник мог бы, разложив на делители n , вычислить самостоятельно число $(p-1) \times (q-1)$ и тем самым, зная e , вычислить секретный ключ d . К счастью, и первая и вторая математические задачи пока не разрешимы за полиномиальное время. Но если новые гении математики найдут полино-

миальное решение хотя бы одной из этих задач, все криптографические системы, использующие схему RSA, будут взломаны за несколько дней. В этой "завязанности" с математической наукой заключается некоторый риск почти всех без исключения схем асимметричного шифрования и цифровой подписи.

За два десятилетия криптоанализа RSA были обнаружены алгоритмы взлома схемы при частных случаях его параметров. Поэтому в настоящее время принцип выбора простых чисел p и q — самой основы криптосхемы — пополнился рядом дополнительных ограничений.

- ❑ Числа $(p + 1)$ и $(q + 1)$ должны содержать в своем разложении на множители большие простые делители (назовем их p_{sup} и q_{sup}).
- ❑ Числа $(p - 1)$ и $(q - 1)$ также должны содержать в своем разложении на множители большие простые делители (назовем их p_{sub} и q_{sub}), причем такие, что бы числа $(p_{sub} - 1)$ и $(q_{sub} - 1)$ также содержали в своем разложении достаточно большие делители.
- ❑ Числа p и q должны очень близко совпадать по порядку длины (это очевидное требование в плане атаки полным перебором — потенциально злоумышленник в этом случае первым отыщет обязательно меньший делитель, следовательно, стойкость всей схемы зависит от размера именно меньшего делителя). Однако нежелательно, чтобы они оказались рядом друг с другом по значению.
- ❑ Секретный ключ d не должен в результате процедуры создания ключей оказаться очень маленьким. Если подобное условие случайно выполнилось, придется "перевыбрать" случайное число e и повторить процедуру вычисления d .
- ❑ Каждый абонент сети должен использовать уникальное значение n . Недопустимо использование схемы, при которой супервизор ключей раздает абонентам пары (e, d) , соответствующие одному и тому же n , в надежде, что абонент не сможет разложить n на множители без знания p и q . Это не так — по известным e, d и n последнее раскладывается на множители за полиномиальное время, т. е. у любого абонента подобной сети появляется возможность читать сообщения всех других пользователей.

В современной криптографии порогом раскрываемости схемы RSA считается длина n в 512 бит. Эта величина находится на "границе доверия". Кроме нее используются криптосхемы с ключами в 768, 1024, 2048 бит. По состоянию на начало 2003 года специализированный чип стоимостью в 10 000 000 долларов, описанный в [Shamir], вскрывает RSA ключ длиной в 1024 бита за 1 год, а подобный ему чип стоимостью в 10 000 долларов — ключ длиной в 512 бит за 10 минут.

В качестве уловки, ускоряющей процесс шифрования (но не дешифрования), используются небольшие по абсолютной величине и по весу Хемминга

(количеству "1" в двоичной записи числа) значения e . При наличии в двоичной записи числа e небольшого количества единиц процедура возведения в степень по модулю простого числа значительно ускорится.

Схема RSA: пример

Продemonстрируем работу схемы RSA на небольших числах. Естественно, из-за малых значений p и q дополнительные требования на их выбор не выдержаны. Пусть $p = 5$, а $q = 11$, тогда значение $n = 55$. В качестве открытого ключа e выберем число 7, таким образом весь открытый ключ имеет вид $(e = 7, n = 55)$. Вычислим закрытый ключ d : уравнение $e \times d + (p - 1) \times (q - 1) \times y = 1$ приобретает вид $7 \times d + 40 \times y = 1$ и имеет в целых числах решение $d = 23, y = -4 = 51$. Таким образом, закрытым ключом является число 23.

Пусть произвольный отправитель хочет передать абоненту комбинацию бит 100111_2 , ее числовой эквивалент: 39_{10} . Возводим 39 в степень открытого ключа $e = 7$ по модулю $n = 55$: $(39^7 \bmod 55) = 19$. Число 19 является шифрограммой и передается по каналу связи. Получатель по приходу сообщения возводит его в степень $d = 23$: $(19^{23} \bmod 55) = 39$. Исходное послание восстановлено.

Схема Рабина

Схема Рабина (*Michael Rabin*) очень похожа на алгоритм RSA, но возводит сообщение m не в высокую степень e , а только лишь в квадрат: $c = (m^2 \bmod n)$ (число n также как и в схеме RSA является произведением двух больших простых чисел p и q). На приемной стороне возможно декодирование сообщения и по схеме дешифрования RSA, но чаще применяется несколько более быстрое извлечение квадратного корня по модулю n .

Процедура извлечения квадратного корня при известном разложении n на сомножители требует полиномиальное время, а при отсутствии подобной информации — гораздо более трудоемкая. Если p и q выбраны так, что остаток от деления на 4 и у первого и у второго числа равен трем, то вычисление квадратного корня производится по следующим формулам:

- алгоритмом Евклида решается в целых числах уравнение $a \times p + b \times q = 1$;
- $r = c^{(p+1)/4} \bmod p$;
- $s = c^{(q+1)/4} \bmod q$;
- $x = (a \times p \times s + b \times q \times r) \bmod n$;

$$\square y = (a \times p \times s - b \times q \times r) \bmod n ;$$

□ корнями числа c являются 4 значения: $(\pm x \bmod n)$ и $(\pm y \bmod n)$.

В качестве преимущества схемы Рабина можно назвать уменьшение открытого ключа: он состоит из одного числа n , что значительно ускоряет процесс шифрования. Недостатком является получение в результате извлечения квадратного корня на приемной стороне четырех различных значений: m_1, m_2, m_3, m_4 . Одно из этих значений действительно является зашифрованным текстом, три других — издержки модульной арифметики. Это свойство требует добавлять перед шифрованием в сообщение какие-либо метки, т. е. применять избыточное кодирование, с тем чтобы алгоритм на приемной стороне с высокой вероятностью мог распознать среди четырех значений действительно верный вариант. Схема Рабина представлена на рис. 18.4.

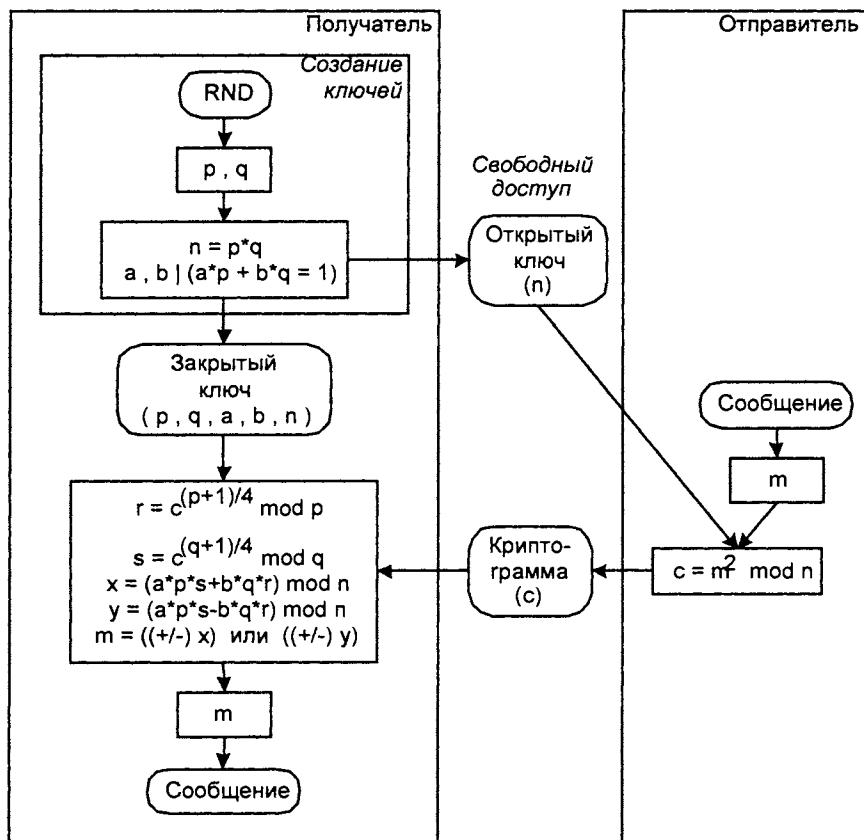


Рис. 18.4. Криптосхема Рабина

Схема Эль Гамаль: структура

Асимметричная схема Эль Гамаль предложена автором (*El Gamal*) как логичное продолжение алгоритма обмена ключей Диффи-Хеллмана, см. главу 20. Она использует ту же операцию возведения в степень по модулю простого числа, но трудноразрешимой для злоумышленника задачей ставит отыскание не числа, которое возведено в степень, а то, в какую степень возведено известное число. Эта задача носит название проблемы *дискретного логарифма*.

На этапе выработки ключей:

1. Выбирается произвольное (правда, достаточно большое) простое число p .
2. Для этого простого числа определяется любой *образующий элемент* (англ. *primitive root*) — т. е. такое число a , при многократном возведении которого в степень по модулю p ($a^1 \bmod p$, $a^2 \bmod p$, ...) будут перебираться (в произвольном порядке, но обязательно только по одному разу) все числа от 1 до $(p - 1)$ включительно.
3. Генерируется произвольное случайное число x ($0 < x < p$) — это и есть закрытый ключ.
4. Вычисляется значение $b = a^x \bmod p$ — комбинация (a, p, b) представляет собой открытый ключ получателя.

На этапе шифрования:

1. Отправитель генерирует произвольное случайное число y ($0 < y < p$).
2. Помещает в начало шифрограммы число $(a^y \bmod p)$.
3. Вычисляет величину $k = (b^y \bmod p) = ((a^x \bmod p)^y \bmod p)$.
4. Используя некоторую, заранее оговоренную в данной реализации, часть k в качестве симметричного ключа для любого блочного шифра шифрует отправляемое сообщение.
5. "Напрочь забывает" числа y и k — надежно стирает их из оперативной памяти и других мест, куда они могли случайно попасть.

На этапе дешифрования:

1. По приходу зашифрованного сообщения получатель отделяет от пакета величину $(a^y \bmod p)$ и вычисляет на ее основе $((a^y \bmod p)^x \bmod p)$ — математика доказывает, что полученное число будет равно тому самому k , которое вычислил отправитель, так как в данной формуле операнды x и y можно менять местами.
2. Выделив из k ту же самую часть, что и отправитель, получатель дешифрует весь идущий далее пакет симметричным алгоритмом.

Схема Эль Гамаль приведена на рис. 18.5.

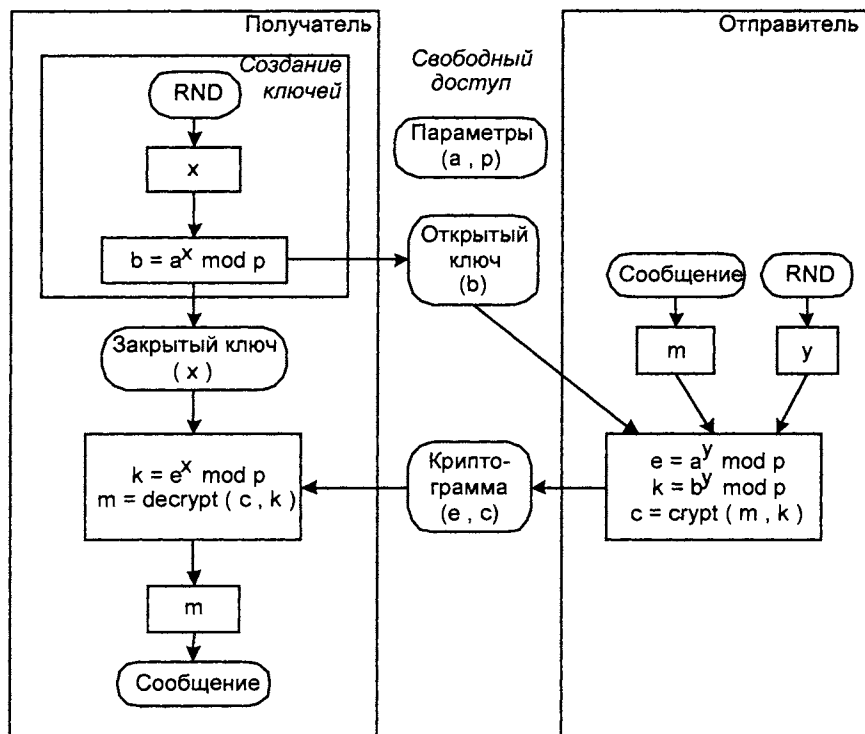


Рис. 18.5. Криптосистема Эль Гамаль

Схема Эль Гамаль: анализ

Проблема дискретного логарифма состоит в том, что, зная основание степени и получившийся после возведения результат по модулю простого числа, невозможно за полиномиальное время определить, в какую именно степень было возведено основание. В схеме Эль Гамаль потенциальный злоумышленник может получить значения a , p , $(a^x \mod p)$ и $(a^y \mod p)$. Однако из-за сложности определения чисел x и y "в чистом виде" у него не оказывается возможности вычислить значение $k = (a^{xy} \mod p)$, которое ему так необходимо для прочтения шифровки.

Относительно числа p криптоанализ выдвигает следующее требование. Число $(p - 1)$ должно содержать в разложении на множители большой простой делитель. В некоторых протоколах с участием схемы Эль Гамаль p выбирается как $(q \times 2 + 1)$, где q — простое число (число p в этом случае называется *простым Софи Жермен*). При выборе x и y получателем и отправителем соответственно, естественно, должно выполняться требование к их информационной емкости. Для генерации этих чисел должен использоваться либо

ГСЧ — генератор "действительно" случайных чисел (не ГПСЧ), либо криптостойкий генератор псевдослучайных чисел (КГПСЧ). В противном случае злоумышленник просто определит x или y полным перебором. Проблема генерации простых чисел рассмотрена в *главе 19*.

Все пользователи информационного пространства могут использовать одну и ту же пару основания конечного поля p и порождающего элемента a — это позволяет не включать эти параметры в открытый ключ, а также использовать предвычисления (кэширование) для значительного ускорения алгоритма. Однако для каждого сообщения должно использоваться новое значение y , в противном случае появляется возможность по одному известному сообщению прочесть все остальные.

По криптостойкости в схеме Эль Гамаль 512-битное число p приравнивается к 56-битному симметричному ключу, о котором уже сложилось однозначное мнение. Поэтому на практике применяются p длиной в 768, 1024 и 1536 бит. Три значения простого числа p и образующие элементы к ним, позаимствованные из стандарта [RFC-2412], приведены в *приложении 11*.

Схема Эль Гамаль: пример

В качестве простого числа, порождающего циклическую группу, выберем $p = 11$, за образующий элемент примем число $a = 7$ (при возведении 7 в степень 1, 2, 3 и т. д. по модулю 11 последовательно проходят все 10 значений [7, 5, 2, 3, 10, 4, 6, 9, 8, 1]). Секретным ключом x выберем 6, параметр b принимает значение $b = (a^x \bmod p) = (7^6 \bmod 11) = 4$. В целом открытый ключ принимает вид $(a = 7, p = 11, b = 4)$.

Предположим, что некий абонент хочет передать сообщение. Он выбрал случайное число, не превосходящее p , например, $y = 9$. В начало шифрограммы помещается число $(a^y \bmod p) = (7^9 \bmod 11) = 8$. Кроме того, на основе y и открытого ключа отправитель вычисляет $k = b^y \bmod p = 4^9 \bmod 11 = 3$. Выбрав значение 3 или какие-либо его биты в качестве симметричного ключа, отправитель шифрует передаваемые данные и стирает величины 9 и 3 со своих накопителей.

Получатель по приходу пакета для вычисления $k = (a^y \bmod p)^x \bmod p$ возводит число 8 из заголовка шифрограммы в степень секретного ключа и получает $k = 8^6 \bmod 11 = 3$ — то же самое значение, которое использовал отправитель, шифруя собственно данные.

Схемы на основе эллиптических кривых

Асимметричное шифрование, использующее *эллиптические кривые*, не является отдельной схемой. Оно представляет собой модификацию других схем, увеличивающую скорость работы алгоритмов и одновременно уменьшающую размеры ключей. Проблема "деления" в специфичной конечной группе, порождаемой эллиптическими кривыми, очень похожа на проблему

дискретного логарифма. В связи с этим асимметричная криптография на эллиптических кривых больше всего напоминает криптосистему Эль-Гамаль.

Эллиптической кривой, используемой в данной схеме, является выражение вида $y^2 = (x^3 + a \times x + b) \bmod p$, где p — большое простое число. Обе координаты x и y , а также параметры a и b являются натуральными числами из диапазона $[0; p - 1]$, т. е. все вычисления производятся по модулю p . Пары чисел (x, y) , удовлетворяющие приведенному равенству, называются *точками эллиптической кривой*.

Над точками определена операция сложения следующим образом. Если абсциссы точек $Q_1(x_1, y_1)$ и $Q_2(x_2, y_2)$ различны, то точка $S = Q_1 + Q_2$ имеет координаты (x_s, y_s) , определяемые формулами:

$$k = ((y_2 - y_1) / (x_2 - x_1)) \bmod p$$

$$x_s = (k^2 - x_1 - x_2) \bmod p$$

$$y_s = (k \times (x_1 - x_s) - y_1) \bmod p.$$

Если же точки Q_1 и Q_2 совпадают, т. е. идет речь об "удвоении точки", то применяются следующие формулы:

$$k = ((3 \times x_1^2 + a) / (2 \times y_1)) \bmod p$$

$$x_s = (k^2 - 2 \times x_1) \bmod p$$

$$y_s = (k \times (x_1 - x_s) - y_1) \bmod p.$$

Подобные формулы позволяют ввести над точками эллиптической кривой операцию *умножения на число*: $R = n \times P$ — n -кратное сложение точки P с самой собой. Данная операция по свойствам тождественна операции возведения в степень в конечном поле простого числа. Само умножение (шифрование) характеризуется полиномиальной скоростью вычислений, а вот попытка по известным P и R определить число n уже не укладывается в полиномиальные рамки.

Схема Мак-Элиса

Схема Мак-Элиса (McEliece), предложенная в 1978 году, основана на проблеме декодирования кода с исправлением ошибок. В общем случае линейный код не имеет полиномиального алгоритма декодирования, в то время как некоторые частные коды имеют эффективные схемы декодирования. В качестве секретного ключа выбирается матрица G разрядности $(k \times n)$, порождающая код с исправлением t ошибок, и две маскирующие матрицы (без знания их для злоумышленника код становится общим случаем линейного кода). Первая маскирующая матрица S разрядности $(k \times n)$ должна обладать свойством невырожденности. Вторая матрица P разрядности $(n \times n)$ является перестановочной, т. е. содержит закон перестановки из n бит. В качестве открытого ключа вычисляется матрица $W = S \times G \times P$, все або-

ненты сети знают в отношении получателя только матрицу W и то, какой степени t ошибку может исправить эта матрица.

При отправке сообщения абонент представляет сообщение M в виде двоичного числа разрядности k бит и выбирает случайный вектор ошибки Z , содержащий в двоичной записи не более t единиц. Шифрограмма получается по формуле $C = M \times W + Z$. Злоумышленник не может прочесть сообщение из-за сложности инвертирования матричного умножения при дополнительном наличии вектора ошибки Z .

На приемной стороне получатель, зная по отдельности P , G и S , сначала выполняет умножение $C = C \times (P^{-1})$, затем над вектором C производит декодирование помехоустойчивого кода, удаляя тем самым внесенную случайную ошибку. Полученный после декодирования двоичный вектор M умножается на обратную матрицу S : $M = M \times (S^{-1})$. Сообщение дешифровано. Криптосхема Мак-Элиса приведена на рис. 18.6.

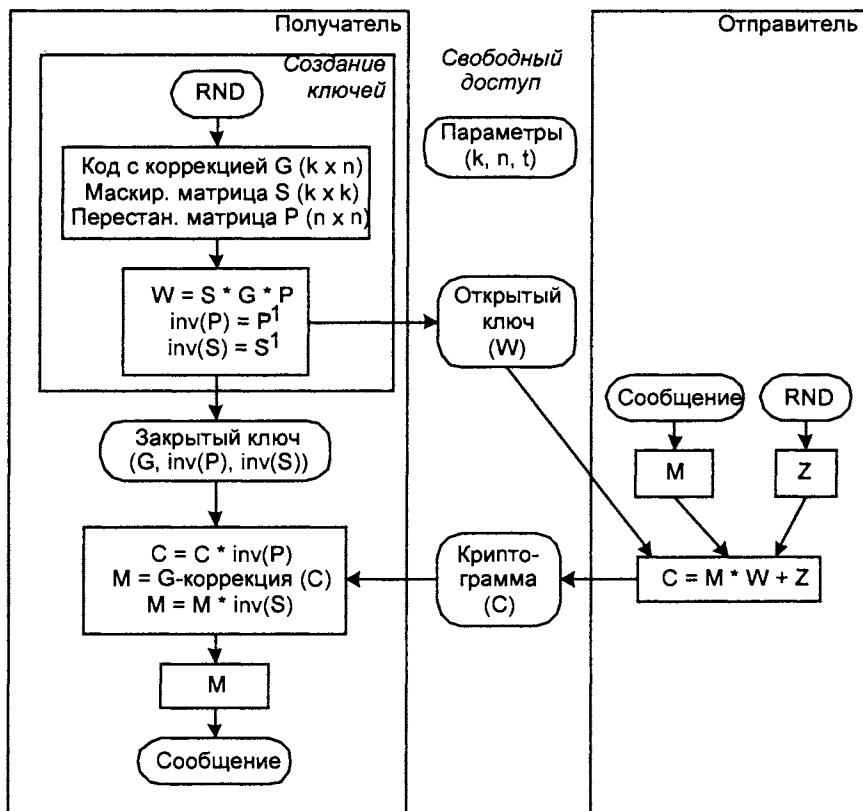


Рис. 18.6. Криптосхема Мак-Элиса

Схема Мак-Элиса обладает очень высокой скоростью как шифрования, так и дешифрования, однако размер открытого ключа у нее очень велик. При достаточно криптостойких размерах порождающей матрицы ($n = 1024$, $t = 38$) размер W составляет порядка 80 Кбайт. Это и сказалось в первую очередь на малом практическом применении данной интересной схемы.

Электронная цифровая подпись



Схема RSA

Инверсия асимметричного алгоритма RSA для получения схемы цифровой подписи является, пожалуй, самой простой из всех алгоритмов ЭЦП. Для формирования электронной подписи отправитель выполняет над контрольной суммой документа h те же самые действия, что и при шифровании, но использует не открытый ключ получателя, а свой собственный закрытый ключ, т. е. $sign_i = (h_i^d \bmod n)$. Открытый и закрытый ключи просто меняются местами. На приемной стороне получатель возводит подпись в степень открытого ключа e отправителя и получает $(sign_i^e \bmod n) = (h_i^{de} \bmod n) = h_i$ (согласно тем же формулам, что и в асимметричном шифровании RSA).

Если получившееся после возведения в степень значение совпадает с вычисленной независимо на приемной стороне контрольной суммой документа, то проверка считается выполненной, а документ — подлинным. Никто, кроме отправителя, не зная d , не может вычислить такую подпись $sign_i$, чтобы возведение ее в степень открытого ключа e дало требуемую контрольную сумму — это та же самая трудноразрешимая задача, что и в асимметричном шифровании RSA. Следовательно, снабдить документ такой подписью $sign_i$ мог только истинный владелец закрытого ключа. Схема ЭЦП приведена на рис. 18.7.

Использование секретного ключа для цифровой подписи, т. е. раскрытие некоторой информации о нем не приносит ему ущерба. Даже перехватив большое количество подписанных сообщений, злоумышленник не сможет с их помощью вычислить d за полиномиальное время, а применяемой на сегодняшний день длины ключа хватит на несколько десятков лет стойкости.

В такой простой схеме ЭЦП есть на самом деле другая потенциальная уязвимость — если злоумышленник активно переписывается с атакуемым абонентом, то умелым сочетанием шифрования и подписей со стороны абонента он сможет читать сообщения, адресованные атакуемым абонентом всем своим корреспондентам. Например, добившись под каким-либо предлогом подписания абонентом нескольких байт абракадабры u , он сможет прочесть одно послание этому абоненту. Произойдет это, если злоумышленник поместил в эту кажущуюся бессмысленной последовательность байт перехваченный ключ сеанса, зашифрованный открытым ключом:

$u = k_i^e \bmod n$. Тогда, будучи подписанной, $(u^d \bmod n)$ представляет собой секретный ключ блочного шифра перехваченного послания в открытом виде: $(u^d \bmod n) = (k_i^e \times d \bmod n) = k_i$.

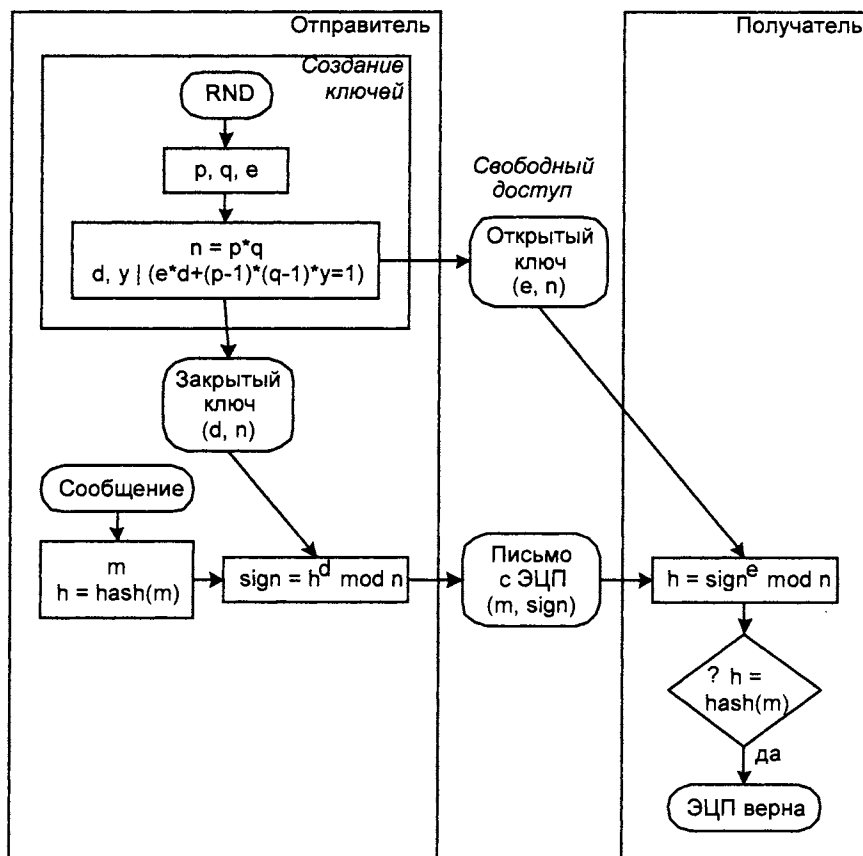


Рис. 18.7. ЭЦП RSA

Частное решение этой проблемы — никогда не подписывать то, чего не создавал сам. Общее — принять за правило подписывать только контрольную сумму интересующего сообщения, даже если все оно настолько мало, что поместится в один блок преобразования RSA. При этом для подобных целей повсеместно используются только криптостойкие (необратимые) контрольные суммы — *хэш-функции*, речь о которых пойдет в *главе 19*. Они не дают злоумышленнику возможности сформировать документ так, чтобы его контрольная сумма совпала с заранее желаемым значением (u), а, следовательно, абонент всегда будет подписывать уникальный битовый вектор, снимая возможность описанной атаки.

Схема Рабина

В цифровой подписи по схеме Рабина отправитель выполняет над контрольной суммой такие действия, которые сделать может только он сам, зная как раскладывается на сомножители p и q его открытый ключ n . Это действие — извлечение квадратного корня в поле натуральных чисел $[0; n - 1]$ (по модулю n). При отправке абонент вычисляет квадратный корень из контрольной суммы, на принимающей стороне получателю остается только возвести ее в квадрат и проверить, совпало ли полученное значение с контрольной суммой подписанного ею документа. Схема ЭЦП по методу Рабина приведена на рис. 18.8.

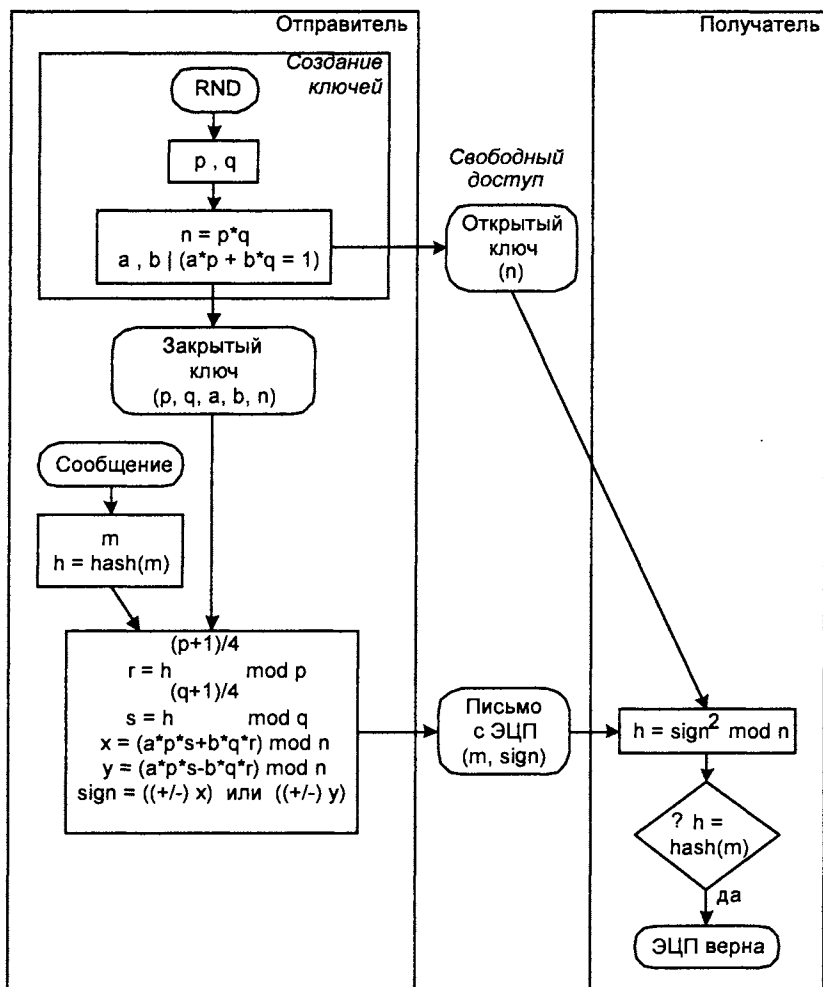


Рис. 18.8. ЭЦП Рабина

Проблема с этой, казалось бы, очень простой схемой в том, что не всякое число имеет квадратный корень в поле по модулю простого числа. Наличие квадратного корня высоко вероятно, но не обязательно. Для решения этой задачи существуют два способа. Согласно первому в подписываемом значении h резервируют несколько бит и перебирают все возможные их комбинации, добиваясь наличия квадратного корня у сгенерированного таким образом числа. По теории вероятностей из 4 попыток хотя бы одно из чисел будет иметь квадратный корень с очень большой вероятностью. На приемной стороне добавленные биты игнорируются. Второй способ носит название *модифицированная ЭЦП по схеме Рабина* и гарантирует наличие квадратного корня у подписываемого числа. Однако по сложности он далеко выходит за рамки данной книги.

Схема Эль Гамаль

Схема Эль Гамаль является одной из самых распространенных схем ЭЦП. Этому послужило, во-первых, то, что при надлежащей и достаточно хорошо проверенной стойкости схема имеет хорошую скорость вычисления. Во-вторых, схема имеет достаточно много модификаций, что в принципе мало свойственно асимметричным шифрам. В схеме Эль Гамаль абонент, подписывающий документ, доказывает всем желающим проверить подпись, что знает секретный ключ x — степень, в которую был возведен образующий элемент a , чтобы получить открытый ключ b . Явным образом продемонстрировать x , естественно, нельзя, иначе любой другой участник информационного обмена начнет подписывать им документы. Поэтому приходится идти на математические уловки — демонстрировать не само число x , а результат некоей математической формулы с участием x . При проверке само число x ни на каком этапе не раскрывается, но проверяющий на основе этой формулы удостоверится, что отправитель сообщения действительно знает x .

Базовый вариант ЭЦП по схеме Эль Гамаль выглядит следующим образом. Обозначения для открытого и закрытого ключей, простого числа и образующего элемента такие же, как и в асимметричном шифровании по схеме Эль Гамаль. На этапе подписания отправитель:

1. Генерирует случайное число k , уникальное для каждого подписываемого документа, взаимно простое с числом $(p - 1)$.
2. Вычисляет $r = (a^k \bmod p)$.
3. Вычисляет обратный элемент поля к числу k — его обозначают $(k^{-1} \bmod (p - 1))$ или $1/k$, в дальнейшем операция "деление на k " равнозначна умножению на $1/k$.
4. Вычисляет $s = ((h - x \times r) / k) \bmod (p - 1)$, где h — контрольная сумма подписываемого документа.

Пара чисел (r, s) является цифровой подписью для документа, имеющего контрольную сумму h . На приемной стороне получатель:

1. Вычисляет $u = ((b^r) \times (r^s) \bmod p)$.

3. Вычисляет $v = (a^h \bmod p)$.
4. Проверяет равенство значений $u = v$ (если равенство выполняется, то подпись верна, в противном случае документ сфальсифицирован).

Почему данное равенство является тождеством в случае корректности ЭЦП? Произведем несложные преобразования над u (подразумеваем, что все действия выполняются по модулю числа p):

$$u = (b^r) \times (r^s) = ((a^x)^r) \times ((a^k)^s) = a^{(xr + h - xr)} = a^h = v.$$

Как видим, во-первых, получилось тождество, а во-вторых, ни на одном этапе преобразований (а все они доступны проверяющему) число x не фигурировало в открытом виде — его как бы "экранируют" две величины r и k , причем вторая не известна получателю. Схема ЭЦП Эль Гамаль приведена на рис. 18.9.

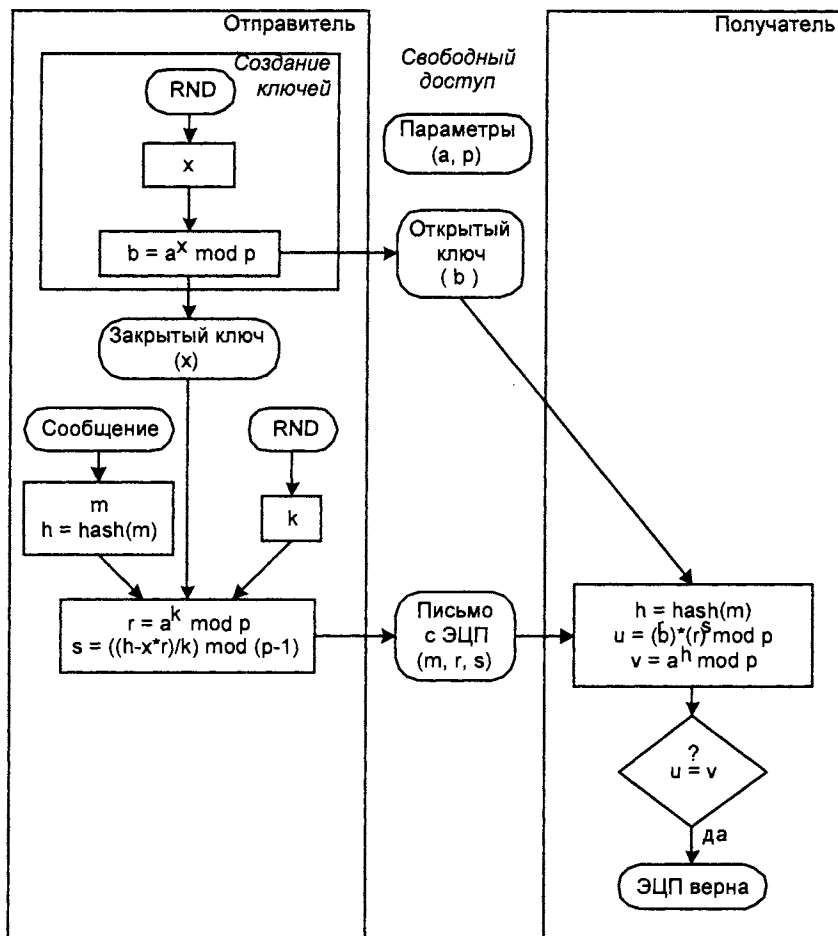


Рис. 18.9. ЭЦП Эль Гамаль

В качестве особенностей ЭЦП схемы Эль Гамаль необходимо упомянуть требование проверять на приемной стороне неравенство ($r < p$). Если такой проверки не делать, то злоумышленник сможет подделать подпись, имея перехваченным хотя бы один документ, подписанный легальным пользователем. Правда, по этому алгоритму значение r получается достаточно большим, что и позволяет защищаться от него сравнением r с простым числом p . Вторым требованием является наличие у числа $(p - 1)$ большого простого делителя. Обычно первоначально выбирается большое простое число q разрядности на один бит меньше требуемой, а затем p вычисляется как $p = 2 \times q + 1$ и проверяется на простоту.

Многочисленные модификации алгоритма Эль Гамаль связаны с несколькими шагами алгоритма. Во-первых, существует возможность при небольшом изменении формул выполнять все вычисления не по модулю p , а по модулю q , являющемуся наибольшим простым делителем p (как и определено ранее). Во-вторых, формула, вовлекающая секретный ключ и контрольную сумму документа и выглядящая в базовом варианте как $h = ks + xr$ (это шаг 4 выработки подписи: k перенесено из знаменателя в числитель, а затем слагаемые поменяны местами), является не единственно возможным соотношением. Другие формулы имеют вид: $h = x \times s + k \times r$, $s = x \times r + k \times h$, $s = x \times h + k \times r$, $r = x \times s + k \times h$, $r = x \times h + k \times s$. Естественно, в каждом из этих случаев и формула проверки меняет свой вид, но основная идея остается неизменной. Третий "пласт" модификации схемы Эль-Гамаль связан с возможностью переноса вычислений в группу, образованную эллиптическими кривыми. Стандарт США электронной цифровой подписи *DSS (Digital Signature Standard)*, принятый в 1992 году, является одной из модификаций схемы Эль Гамаль и приведен на рис. 18.10. Государственный стандарт Российской Федерации на ЭЦП ГОСТ Р 34.10–2001 также основан на схеме Эль Гамаль.

ГОСТ Р 34.10–2001

Алгоритм Государственного стандарта РФ по ЭЦП (полное название *Процессы формирования и проверки электронной цифровой подписи*) является переложением схемы Эль Гамаль в область эллиптических кривых. Как уже указывалось в разд. "Схемы на основе эллиптических кривых" настоящей главы, общеизвестными параметрами являются сама эллиптическая кривая и точка P на ней, превращающаяся после q -кратного сложения с самой собой в "нулевую" точку кривой. Секретным ключом отправителя d является случайным образом сгенерированное число, а открытым — точка Q , определяемая по формуле ($Q = d \times P$).

На этапе выработки подписи отправитель:

1. Генерирует случайное число k ($0 < k < q$).
2. Вычисляет x -координату точки ($k \times P$), назовем это число r .
3. Вычисляет значение $s = (rd + kh) \bmod q$, где h — контрольная сумма подписываемого сообщения.

Подписью является пара чисел (r, s) .

На этапе проверки выполняются следующие действия:

1. Вычисляются значения $(z_1 = (s \times (h^{-1})) \bmod q)$ и $(z_2 = ((-r) \times (h^{-1})) \bmod q)$.
2. Определяется x-координата точки $(z_1 \times P + z_2 \times Q)$, назовем это число R .
3. Проверяется равенство $(r = R)$ (если оно верно, подпись корректна).

Разрядность числа q фиксируется Стандартом в 255 бит. Это означает, что при заданной стойкости открытый и закрытый ключи имеют гораздо меньшую длину и требуют при вычислениях меньших затрат ресурсов — все это достоинства эллиптических кривых. Схема ЭЦП по ГОСТ Р 34.10–2001 приведена на рис. 18.11.

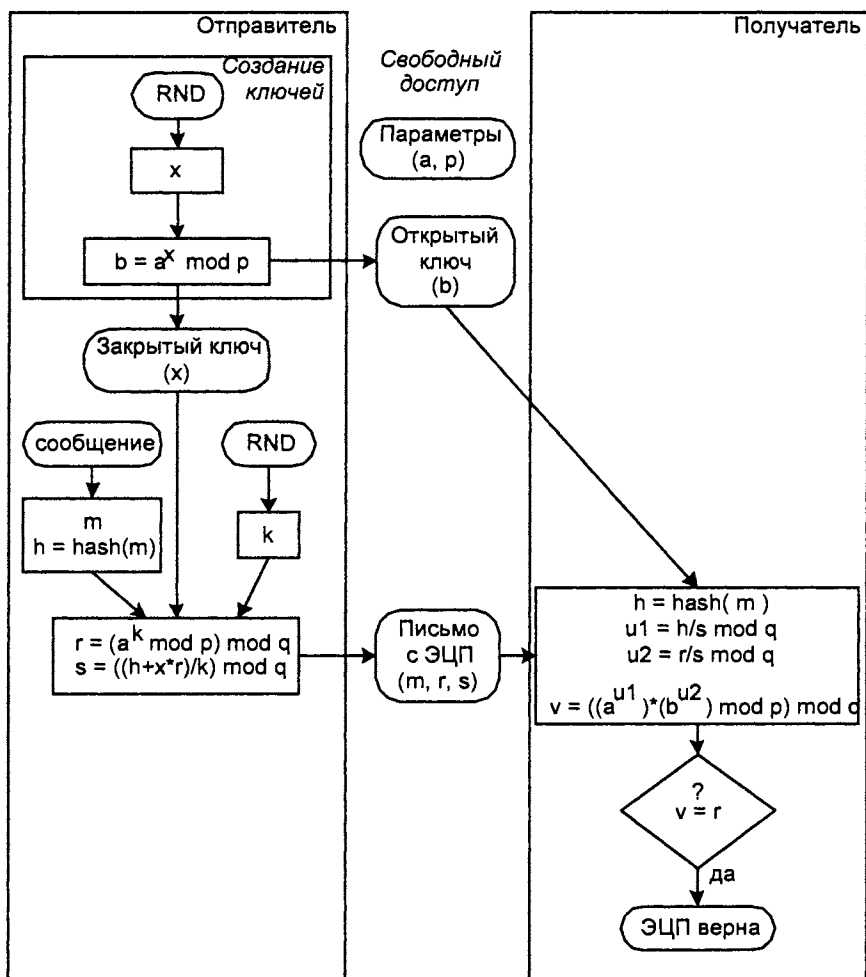


Рис. 18.10. Алгоритм ЭЦП DSS

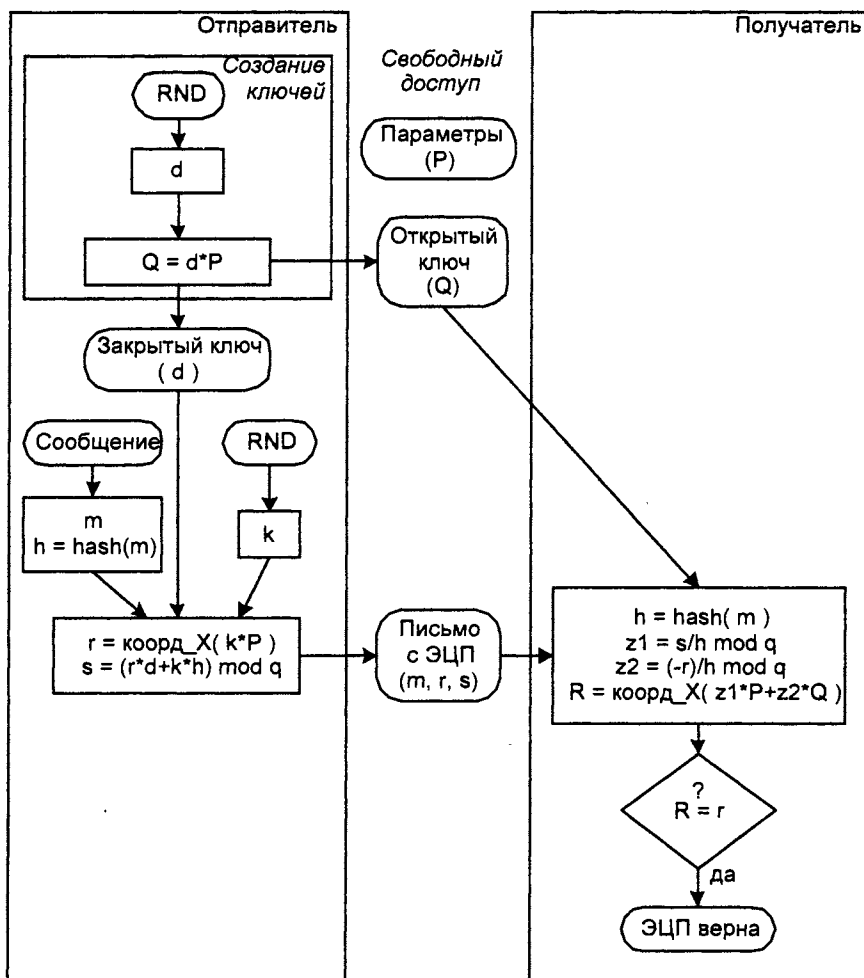
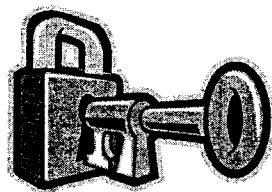


Рис. 18.11. ЭЦП по ГОСТ Р 34.10–2001

Глава 19



Прочие криптографические алгоритмы

Рассматриваемые в *данной главе* алгоритмы носят в криптосистемах не основной, а вспомогательный характер — ни один из них не может сам по себе обеспечить ни одну из категорий информационной безопасности. Однако все они стали неотъемлемыми "кирпичиками" сложных криптопротоколов, в которых ведущую роль играет симметричный или асимметричный шифр. Более того, за несколько десятилетий криптоаналитики выяснили, что проектирование описываемых в данной главе алгоритмов является таким же важным для криптопротокола и непростым этапом, как и проектирование его основного шифра.

Хэш-функции

Свойства и назначение хэш-функций



Хэш-функцией (англ. *hash* — мелко измельчать и перемешивать) называется необратимое преобразование данных, обладающее следующими свойствами:

- ❑ на вход алгоритма преобразования может поступать двоичный блок данных произвольной длины;
- ❑ на выходе алгоритма получается двоичный блок данных фиксированной длины;
- ❑ значения на выходе алгоритма распределяются по равномерному закону по всему диапазону возможных результатов;
- ❑ при изменении хотя бы одного бита на входе алгоритма его выход значительно меняется: в идеальном случае инвертируется произвольная половина бит.

Хэш-функции получили широкое распространение в разнообразных алгоритмах быстрого поиска информации. Однако с появлением криптографии у них появилась вторая, ничуть не меньшая, область применения. Хэш-

функция называется *криптографически стойкой*, если в дополнение к перечисленным свойствам она удовлетворяет еще двум требованиям:

- ❑ зная результат хэш-функции, невозможно подобрать, кроме как полным перебором, какой-либо входной блок данных, дающий такое же значение на выходе;
- ❑ невозможно подобрать, кроме как полным перебором, пару различных входных блоков, дающих на выходе произвольный, но одинаковый результат.

Если для целей поиска данных разрядность результата хэш-алгоритма обычно выбирается от 10 до 20 бит, то для криптографии алгоритм проектируется так, чтобы разрядность результата равнялась 128, 160, 192, а то и 256 битам. В дальнейшем все обсуждение будет посвящено только криптостойким хэш-функциям, поэтому сам термин "криптостойкая" будет опускаться.

Основное, но не единственное, предназначение хэш-функций в криптографии — вычисление "неподделываемых" контрольных сумм документов. Действительно, если для алгоритма хэширования выполняются требования невозможности подобрать иной документ с той же хэш-суммой и невозможности подобрать два документа с произвольной одинаковой хэш-суммой, то хэш-сумма становится уникальной характеристикой документа. Во-первых, в документ нельзя внести изменения так, чтобы хэш-сумма осталась неизменной. А во-вторых, разрядность в 128 или в 160 бит гарантирует, что на сегодняшнее время в мире не существует двух разных документов, имеющих одинаковую хэш-сумму.

Несомненно, теоретически возможно наличие неограниченного количества различных входных данных, имеющих одинаковую хэш-сумму, ведь хэширование преобразует блок неограниченной длины в фиксированную запись. Инцидент появления двух или более различных документов с одинаковой суммой называется *хэш-коллизия*. Из теоремы, носящей название *парадокс дней рождения*, следует, что для того, чтобы создать одну коллизию для 128-битной хэш-суммы, необходимо в среднем 2^{64} документов. Много это или мало? Оказывается, каждый житель планеты должен создавать в течение всей своей жизни ежесекундно по документу, чтобы с вероятностью 1/2 на Земле появилось два документа с одинаковыми 128-битными хэш-суммами.

Получающееся таким образом "практически" взаимно однозначное соответствие между документом и хэш-суммой позволяет защищать целостность не самого документа, который может иметь произвольно большой объем, а только лишь 16-байтного блока данных, т. е. с помощью хэш-функций проблема защиты большого блока данных сводится к проблеме защиты маленького блока данных, да к тому же еще и заранее известной длины. Так, в системах электронной цифровой подписи (ЭЦП) подписывается не сам до-

кумент, а только его хэш-сумма. Если на приемной стороне с помощью ЭЦП проверена целостность хэш-суммы, а вычисленное получателем самостоятельно хэш-сумма документа совпадает с присланным, то и весь документ признается аутентичным — внести в него изменения, не изменив значение хэш-суммы, было невозможно.

Хэш-сумму можно применить и для создания *симметричной электронной подписи*. Это может быть необходимо в том случае, когда отправитель и получатель имеют общий секретный ключ, но не хотят шифровать само сообщение (в этом случае проблема аутентификации исчезла бы сама с собой), а только лишь подтвердить друг перед другом его истинность. Например, сообщение после отправки необходимо будет продемонстрировать в открытом виде множеству лиц. Симметричная электронная подпись вычисляется очень просто: к самому документу добавляется секретный ключ (или функция от него) и производится хэширование объединенного блока данных. По хэш-сумме нет возможности восстановить исходный текст, даже если известна некоторая его часть, как и получается в этом случае. Единственный вариант действий злоумышленника — перебирать все варианты "добавки" до тех пор, пока не получится известное ему значение хэш-функции. Таким образом, единственное требование к симметричной электронной подписи — добавлять достаточно большой блок данных, недоступный для полного перебора. Наибольшее распространение получила схема, по которой до и после документа (назовем его *text*) добавляется значение секретного ключа *key*, затем производится хэширование. Полностью формула электронной подписи по этой схеме выглядит так:

$$\text{sign} = \text{hash}(\text{key} \mid \text{text} \mid \text{key}).$$

Полученное значение прикрепляется к подписываемому документу:
 $\text{text} = (\text{text} \mid \text{sign}).$

Основы построения хэш-функций



Общепринятым принципом построения хэш-функций является *итеративная последовательная схема*. По этой методике ядром алгоритма является преобразование k бит в n бит. Величина n — разрядность результата хэш-функции, а k — произвольное число, большее n . Базовое преобразование должно обладать всеми свойствами криптостойкой хэш-функции, т. е. необратимостью и невозможностью инвариантного изменения входных данных. О самом базовом преобразовании речь пойдет в последующих двух разделах, а пока рассмотрим как с его помощью строится полноценная хэш-функция.

Хэширование производится с помощью промежуточной вспомогательной переменной разрядностью в n бит. В качестве ее начального значения выбирается произвольное известное всем сторонам значение, например, 0.

Входные данные разбиваются на блоки по $(k - n)$ бит. На каждой итерации хэширования со значением промежуточной величины, полученной на предыдущей итерации, объединяется очередная $(k - n)$ -битная порция входных данных, и над получившимся k -битным блоком производится базовое преобразование. В результате весь входной текст оказывается "перемешанным" с начальным значением вспомогательной величины. Из-за характера преобразования базовую функцию еще часто называют *сжимающей*. Значение вспомогательной величины после финальной итерации поступает на выход хэш-функции (рис. 19.1). Иногда над получившимся значением производят дополнительные преобразования. Но в том случае, если сжимающая функция спроектирована с достаточной степенью стойкости, это преобразование излишне.

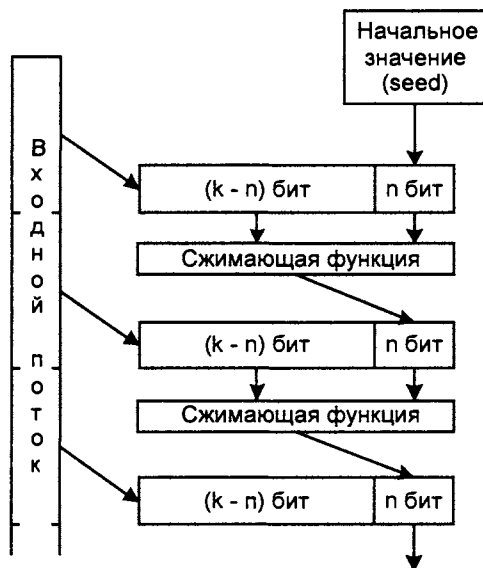


Рис. 19.1. Итеративные хэш-функции (сжимающая функция)

При проектировании хэш-функции по итеративной схеме возникают два взаимосвязанных вопроса: как поступать с данными, не кратными числу $(k - n)$, и как добавлять в хэш-сумму длину документа, если это требуется. Получили распространение два алгоритма решения этих вопросов. В первом варианте в начало документа перед хэшированием добавляется поле фиксированной длины (например, 32 бита), в котором в двоичном коде записывается исходная длина текста. Затем объединенный блок данных дополняется нулями до ближайшего кратного $(k - n)$ бит размера. Второй вариант дополняет документ справа одним битом "1", а затем до кратного $(k - n)$ бит

размера битами "0". В этом случае необходимость в поле длины отпадает — никакие два разных документа после выравнивания по границе порций не станут одинаковыми.

Кроме более популярных однопроходных алгоритмов хэширования существуют и схемы многопроходных алгоритмов. В этом случае входной блок данных на этапе расширения неоднократно повторяется, а уже затем дополняется до ближайшей границы порции.

Хэш-функции на основе блочных шифров



В качестве сжимающей функции алгоритма хэширования может быть использован обычный блочный шифр. Действительно, если учесть, что любой блочный шифр имеет два входа: открытый текст и ключ, а на выходе генерирует зашифрованный текст, то обнаружится, что это идеальный претендент на ядро хэш-функции. Тем более, необходимо отметить тот факт, что блочные шифры необратимы по ключу шифрования, т. е., если порции хэшируемых данных подавать именно на вход "ключ" шифра, то автоматически выполняется условие необратимости хэширования.

В тех случаях, когда требуемая разрядность хэш-функции n совпадает с размером блока шифра, алгоритм сжимающей функции становится предельно простым. Величина $(k - n)$ полагается равной длине ключа блочного шифра, т. е. k равняется сумме размера блока и длины ключа шифра. Существует три основных схемы превращения блочного шифра в сжимающую функцию хэш-алгоритма. Все они имеют примерно равную стойкость и приведены на рис. 19.2.

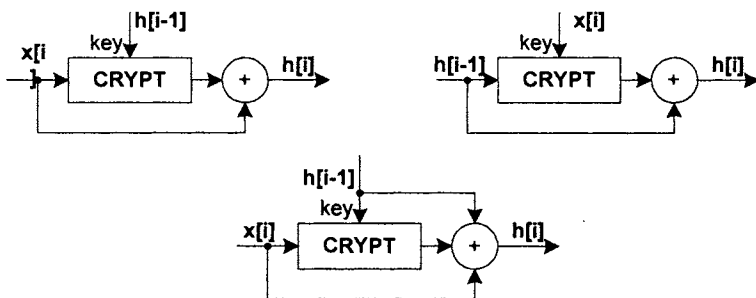


Рис. 19.2. Хэш-функции на блочных шифрах одинарной размерности

Для тех случаев, когда результат хэширования превосходит размер блока шифра ровно в два раза, также предложено множество различных типовых схем. Приведем алгоритм *TandemDM* (*Tandem Davies-Meyer*) (рис. 19.3), предложенный Лаи (*X. Lai*) и Мессэй (*J. L. Massey*) на основе схемы одинарной размерности Дэвиса-Майера (*D. Davies, C. Meyer*).

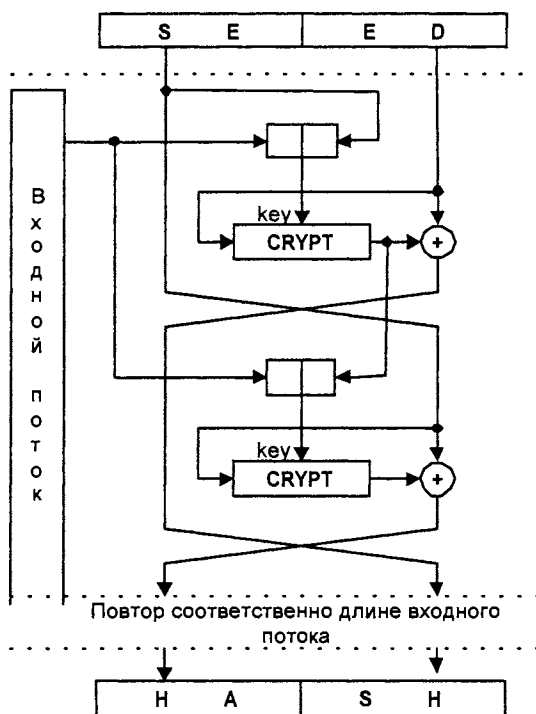


Рис. 19.3. Алгоритм TandemDM

Самостоятельные хэш-алгоритмы



Недостатком хэш-функций, спроектированных на основе блочных шифров, является несколько заниженная скорость работы. Дело в том, что ту же самую стойкость относительно двух основных требований к хэш-функции можно обеспечить за гораздо меньшее количество операций над входными данными. Но для этого алгоритм необходимо изначально проектировать специально, исходя из тандема требований (стойкость, скорость). Рассмотрим два самостоятельных алгоритма криптостойкого хэширования, получивших наибольшее распространение на сегодняшний день.

Алгоритм *MD5* (*Message Digest № 5*), разработан Роналдом Ривестом (*Ronald Rivest*) на основе своего же предыдущего алгоритма *MD4*. Алгоритм использует четыре многократно повторяющиеся преобразования над тремя 32-битными величинами U , V и W :

$$f(U, V, W) = (U \text{ AND } V) \text{ OR } ((\text{NOT } U) \text{ AND } W)$$

$$g(U, V, W) = (U \text{ AND } W) \text{ OR } (V \text{ AND } (\text{NOT } W))$$

$$h(U, V, W) = U \text{ XOR } V \text{ XOR } W$$

$$k(U, V, W) = V \text{ XOR } (U \text{ OR } (\text{NOT } W)).$$

Также в алгоритме используются следующие константы:

□ начальные константы промежуточных величин –

$H[0]=67452301_{16}$, $H[1]=\text{EFCDAB89}_{16}$, $H[2]=98\text{BADCFE}_{16}$, $H[3]=10325476_{16}$;

□ константы сложения в раундах –

цикл по j от 0 до 63

$y[j]=\text{HIGHEST\_32\_BITS}(\text{ABS}(\sin(j+1)))$,

где функция $\text{HIGHEST\_32\_BITS}(x)$ отделяет 32 самых старших бита из двоичной записи дробного числа x , функция $\text{ABS}(x)$ — взятие арифметического модуля числа x , а операнд $\sin(j+1)$ считается взятым в радианах;

□ массив порядка выбора ячеек в раундах:

$z[0\dots63] = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,$
 $1, 6, 11, 0, 5, 10, 15, 4, 9, 14, 3, 8, 13, 2, 7, 12,$
 $5, 8, 11, 4, 1, 4, 7, 10, 13, 0, 3, 6, 9, 12, 15, 2,$
 $0, 7, 14, 5, 12, 3, 10, 1, 8, 15, 6, 13, 4, 11, 2, 9);$

□ массив величины битовых циклических сдвигов влево:

$s[0\dots63] = 7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22,$
 $5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20,$
 $4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23,$
 $6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21).$

На первоначальном этапе входной блок данных дополняется одним битом "1". Затем к нему добавляется такое количество битов "0", чтобы остаток от деления длины блока на 512 составлял 448. Наконец, к блоку добавляется 64-битная величина, хранящая первоначальную длину документа. Получившийся входной поток имеет длину кратную 512 битам.

Каждый 512-битный блок проходит через сжимающую функцию, которая производит перемешивание вспомогательного блока $H0\dots H3$ с входным 512-битным, представленным в виде 16 32-битных значений $X[0]...X[15]$:

$(A, B, C, D) = (H[0], H[1], H[2], H[3])$

цикл по j от 0 до 15

начало_цикла

$T = (A + f(B, C, D) + X[z[j]] + y[j]) \text{ ROL } s[j]$

$(A, B, C, D) = (D, B+T, B, C)$

конец_цикла

цикл по j от 16 до 31

начало_цикла

$T = (A + g(B, C, D) + X[z[j]] + y[j]) \text{ ROL } s[j]$

$(A, B, C, D) = (D, B+T, B, C)$

конец_цикла

цикл по j от 32 до 47

начало_цикла

$T = (A + h(B, C, D) + X[z[j]] + y[j]) \text{ ROL } s[j]$

$(A, B, C, D) = (D, B+T, B, C)$

конец_цикла

цикл по j от 48 до 63

начало_цикла

$T = (A + k(B, C, D) + X[z[j]] + y[j]) \text{ ROL } s[j]$

$(A, B, C, D) = (D, B+T, B, C)$

конец_цикла

$(H[0], H[1], H[2], H[3]) = (H[0]+A, H[1]+B, H[2]+C, H[3]+D)$

После того как все 512-битные блоки прошли через процедуру перемешивания, временные переменные $H[0]$, $H[1]$, $H[2]$, $H[3]$ объединяются, и 128-битное значение подается на выход хэш-функции.

Алгоритм *SHA-1* (*Secure Hash Algorithm*) предложен Институтом Стандартизации США NIST как стандарт хэширования в гражданской криптографии. Этот алгоритм, как и *MD5*, основан на предыдущей разработке Рональда Ривеста *MD4* и также был призван дать еще больший запас прочности к криптоатакам.

Делая небольшой исторический экскурс, отметим, что *MD4* очень похож на *MD5* — отличие составляют только константы: в *MD4* 48 проходов основного преобразования, а не 64, как в *MD5*, и простейшие изменения в алгоритме наложения данных. Несмотря на большую популярность, *MD4* "медленно, но верно" был взломан. Сначала появились публикации об атаках на упрощенный алгоритм. Затем было заявлено о возможности найти два входных блока сжимающей функции *MD4*, которые порождают одинаковый выход. Наконец, в 1995 году было показано, что найти коллизию, т. е. "хэш-двойник" к произвольному документу, можно менее чем за минуту, а добиться "осмысленности" фальшивого документа (то есть наличия в нем только печатных ASCII-символов с определенными "разумными" законами расположения) — всего лишь за несколько дней.

Отличия *SHA-1* от *MD4*, а, следовательно, и от *MD5*, состоят в следующем. Разрядность выхода алгоритма увеличена до 160 бит — это дает дополнительный запас стойкости к хэш-коллизиям. Сжимающая функция состоит из 80 шагов, а не из 48 или 64 как в семействе MD. Расширение входных данных из 16 32-битных слов в 80 производится не простым их повторением в другом порядке (как это делает массив z в *MD4* и *MD5*), а рекуррентной формулой, т. е. значения ячеек входных данных X с номерами от 16 до 79 зависят от

многих предыдущих (основных) ячеек. Процесс перемешивания величин A , B , C , D , E с результатом функций $f(U, V, W)$, $g(U, V, W)$, $h(U, V, W)$ также усложнен.

Несмотря на уже найденные частичные атаки (либо на сжимающую функцию, либо на упрощенный алгоритм) на MD5 и SHA-1, полноценных алгоритмов создания хэш-коллизий к ним не продемонстрировано. Эти два алгоритма получили наиболее широкое распространение — на их долю приходится более 80% хэширования в гражданской криптографии.

Генерация случайных чисел



Проблема генерации случайных чисел на электронно-вычислительной технике стояла давно, но с развитием криптографии она получила мощный дополнительный стимул к развитию. Специфика криптографии, кроме того, наложила на генераторы случайных чисел (также как и на хэш-функции) свои дополнительные требования. Если спроектированные ранее 16 или 32-битные генераторы псевдослучайных чисел вполне устраивали как математиков, работавших на ЭВМ, так и создателей компьютерных игр, то криптографов они не удовлетворяли совершенно.

Случайные числа используются в каждой второй криптосхеме. Они применяются при создании ключей сеанса и закрытых ключей асимметричных схем, при подписании документов и в схемах разделения секрета. При этом сам по себе вопрос генерации случайных чисел на ЭВМ не так уж прост. Несмотря на кажущийся огромный объем случайных величин в компьютере, подавляющее их число детерминировано и иногда очень жестко. Нельзя говорить, что в ЭВМ совсем нет случайных величин, но объем действительно случайной и непредсказуемой информации в этих величинах гораздо меньше, чем кажется на первый взгляд. Поэтому при генерации криптостойких случайных величин часто говорят об их *информационной емкости* — энтропии действительно случайного сигнала в них.

Примечание

Понятие энтропии имеет множество определений, но в данном случае мы используем ее как меру неупорядоченности или непредсказуемости информационных элементов. Чем она выше, тем более хаотично — непредсказуемо заранее — распределена информация.

Генерации псевдослучайных чисел, к которым относятся всем известные ГПСЧ языков высокого уровня random, и их стойкости будет посвящен *разд. "Генерация псевдослучайных чисел" настоящей главы*, а пока рассмотрим возможность получения действительно случайных чисел на ЭВМ. Все методы, позволяющие это сделать, делятся на *автоматические* (без участия человека-оператора) и *автоматизированные* (с участием оператора).

Автоматические генераторы случайных чисел (ГСЧ) пытаются извлечь случайную информацию из технических или физических процессов, происходящих в окружающей среде без участия человека. Во многих ситуациях автоматические ГСЧ — единственный возможный путь работы системы. Примеры причины этого:

- огромный объем требуемых случайных величин;
- физическая невозможность размещения человека в точке создания случайных величин.

Первый (и один из наименее стойких) классов автоматических ГСЧ пытается извлечь случайные компоненты из самой ЭВМ. Обработываемыми параметрами являются в различных комбинациях количество, время и адреса обращения системы к своим внутренним модулям, портам ввода/вывода, процессору, памяти и сетевым ресурсам. Несмотря на кажущийся большой объем параметров, каждый из них в отдельности можно подвергнуть жесткой критике. В том случае когда ЭВМ находится в практически изолированных от внешней среды условиях, почти все процессы в ней начинают повторяться с завидной периодичностью, и доля случайности во всех параметрах резко падает. Проблему решил бы человек-оператор как гарантированный источник случайной составляющей, но это уже другой тип ГСЧ.

Второй класс автоматических ГСЧ извлекает информацию из физических процессов, не связанных с ЭВМ. Здесь уже гораздо шире поле для деятельности, и некоторые схемы в состоянии генерировать невероятно качественный (большой) случайный поток данных. Основными физическими устройствами, используемыми для этих целей, являются высокоточные измерители теплового шума полупроводниковых устройств, цепи с автоколебаниями, конденсаторы, датчики радиоактивности, фотодатчики, устройства для оцифровки аудио- и видеопотоков, радиоприемники.

Проблемой является то, что для обычного типового компьютера, как видим, практически единственным параметром, который можно оцифровать и использовать, является аудиофон помещения. Схемы ГСЧ на основе звуковой карты позволяют создавать поток с энтропией от 10 до 50 Кбит/с (все зависит от чувствительности микрофона). Радиотюнер, если таковой имеется в ЭВМ, может генерировать поток большей мощности при приеме на частоте, свободной от радиовещания. Все остальные физические величины требуют добавления в ЭВМ дополнительных устройств.

Светом в конце туннеля может служить намерение многих производителей на фоне растущего веса криптографии в электронном мире встраивать физические ГСЧ в свое оборудование. Первой "масштабной" ласточкой стал проект Intel, встроивший ГСЧ в некоторые версии чипсета из семейств Intel 810 и Intel 820. Получить доступ к генератору в операционных системах семейства Microsoft Windows можно через технологию *CSP (Crypto Service Provider)* — после установки драйвера от Intel на ЭВМ, снабженных

ГСЧ, появляется CSP с возможностью создания случайных чисел. В драйвер встроена возможность самотеста по стандарту FIPS 140-1 (подробнее о нем см. в *разд. "Генерация псевдослучайных чисел" данной главы*).

Основными проблемами автоматических ГСЧ является повышенная уязвимость к различным злонамеренным действиям. Во-первых, злоумышленник очень часто может иметь доступ к тем же параметрам, что и генератор. Например, радиотракт и компьютерную сеть можно прослушать, на аудио- и радио-поток навести шумы с известными параметрами. Кроме того, возможна и физическая атака на устройство, порождающее случайный параметр, если оно размещено вне ЭВМ, например, на радиоантенну или на счетчик Гейгера.

Негативной стороной любого автоматического ГСЧ является то, что "на глаз" невозможно проверить, насколько высока энтропия генерируемого им потока, не отпаялся ли контакт у микрофона или фотодатчика, не отломлена ли со злым умыслом антенна. Поэтому неотъемлемым требованием к автоматическим ГСЧ является наличие процедуры самотеста, которую необходимо выполнять ежедневно, а во многих случаях и гораздо чаще. Наиболее отказостойкими и помехоустойчивыми физическими ГСЧ принято считать размещаемые внутри корпуса ЭВМ полупроводниковые устройства: датчики теплового шума, автоколебательные цепи и измерители заряда конденсаторов.

Автоматизированные генераторы случайных чисел не могут обойтись без человека-оператора — именно он является случайным процессом в них. Схема генерации с участием человека довольно проста: анализируются входные данные или временные характеристики (что гораздо лучше) с любого устройства ввода. Наиболее часто в этом качестве используются клавиатура и манипулятор мышь. В случае применения клавиатуры пользователя просят набирать произвольный текст, при использовании мыши — делать хаотические движения ею. Автоматизированный ГСЧ анализирует либо собственно ввод (нажатые клавиши или относительные координаты мыши), либо временные задержки между нажатиями или фиксированными изменениями координат, измеряемые с высокой точностью. Современные процессоры содержат внутренний счетчик выполненных команд — у процессоров семейства Intel это команда RDTSC. Естественно, что все данные, полученные описанными способами, требуют тщательной пост-обработки, поскольку корреляция между ними очень высока.

Все криптостойкие алгоритмы ГСЧ как автоматические, так и автоматизированные уделяют огромное внимание финальным шагам обработки случайных данных. Не существует ни одного случайного процесса, в выходном потоке которого не было бы корреляций на том или ином уровне. Но любая корреляция — это уже неслучайная информация, это снижение энтропии сообщения. Если блок информации поддается компрессии хотя бы одним,

пусть даже самым изысканным, алгоритмом сжатия данных, значит, его информационная емкость меньше его объема. А в применении к криптографии это грозит тем, что после использования подобного "неполноценного" 128-битного ключа сеанса, злоумышленнику для взлома сессии будет достаточно перебрать, скажем, всего 2^{60} возможных значений.

Очевидный путь решения этой проблемы — выбирать в качестве выходного значения ГСЧ результат того алгоритма сжатия, который породил самый маленький (а, следовательно, наиболее близкий к энтропии) результат компрессии. Однако существует и гораздо более универсальный метод. Если проектировщики ГСЧ заранее знают, какой "ширины" случайный поток они действительно могут извлечь из данного физического или технического процесса, то достаточно над коррелированными данными выполнить сжимающую операцию хэширования, рассчитав требуемый коэффициент сжатия, т. е. данные от случайного процесса разбиваются для хэша на блоки такой длины, чтобы после хэширования поток имел "в ширину" чуть меньше энтропии случайного процесса.

Например, предположим, что аудиофон помещения дает 20 случайных килобит в секунду, а поток со звуковой карты идет с характеристиками 44 кГц, 16 бит (то есть примерно 720 Кбит/с), таким образом, необходимо сжатие в 36 раз. Пусть применяется алгоритм хэширования MD5, генерирующий на выходе 128-битное значение. Тогда нарезка данных аудиопотока для хэш-функции должна составлять по $128 \times 720/20 = 4600$ бит = 576 байт. Пример для ввода с клавиатуры: при случайном "бряцании" по клавиатуре одно нажатие несет не более 3 бит случайной информации. Следовательно, на вход хэш-функции должны поступать блоки по $128 \times 8/3 = 340$ бит = 43 символа, только тогда 16 байт, созданные алгоритмом MD5, будут иметь энтропию близкую к 100%. Пример для измерения времени между нажатиями на клавиатуре: одно нажатие несет около 10 бит случайной информации о временном интервале. Если использовать 64-битный счетчик команд RDTSC, то один хэшируемый блок должен иметь размер $128 \times 64/10 = 820$ бит = 13 временных интервалов.

Генерация псевдослучайных чисел



Целью генерации псевдослучайных чисел (ГПСЧ) является формирование на основе некоторого блока данных небольшой длины (и обычно действительно случайного) значительно большей по размеру последовательности чисел, которые по виду "невозможно отличить" от действительно случайных. Термин *невозможно отличить* на сегодняшний день воспринимается в следующем смысле: не существует алгоритма с полиномиальной скоростью работы, который на основе достаточно большой последовательной выборки смог бы

определить, является ли генератор случайным или псевдослучайным. Случайный блок небольшого объема, порождающий псевдослучайную последовательность, называется *начальным*, или *инициализирующим, вектором* (англ. *seed* — зерно, семя), его размерность N (в битах) имеет очень важное значение для всего ГПСЧ.

Очевидно, что вне зависимости от объема сгенерированного псевдослучайного потока его информационная емкость равна N бит (или — для плохих генераторов — еще меньше). Так как ГПСЧ является конечным автоматом, то через 2^N созданных им бит он обязательно заиклится, некачественные генераторы могут иметь циклы еще меньшей длины.

Основными алгоритмами ГПСЧ, применяемыми для общетехнических целей, являются:

- простые рекуррентные арифметические формулы (сложение, умножение) с отбрасыванием нескольких первых и последних бит результата;
- те же формулы, но взятые по модулю простых чисел;
- разложение в десятичную или иную дробь иррациональных чисел.

Разрядность начального вектора обычно в таких алгоритмах составляет 16 или 32 бита — 4 млрд неповторяющихся значений вполне достаточно для большинства прикладных задач.

Для поверхностной проверки ГПСЧ на соответствие своему предназначению были разработаны простейшие статистические тесты. Проверка с их помощью, конечно, не гарантирует того факта, что перед нами хороший ГПСЧ, но все таки помогает в автоматическом режиме выявлять многие ошибки в проектировании или сбои в момент работы в модулях ГПСЧ. В США в качестве стандарта принят набор из 4 тестов, получивший номенклатурный номер FIPS 140-1.

ГПСЧ считается прошедшим тестирование по стандарту FIPS 140-1, если произвольная последовательность из 20000 бит, сгенерированная им, удовлетворяет всем четырем требованиям.

1. Количество единиц X_1 в тестовом блоке должно лежать в пределах $9654 < X_1 < 10346$.
2. Последовательность разбивается на 5000 блоков по 4 бита каждый, подсчитываются частоты $q[0]...q[15]$ появления блоков от "0000₂" до "1111₂" соответственно, вычисляется сумма квадратов частот $Q = q[0]^2 + q[15]^2$, наконец, вычисляется критерий $X_2 = ((Q \times 2^4/5000) - 5000)$ — он имеет, т. н. *статистическое распределение "χ-квадрат"* с 2^4-1 степенями свободы и должен лежать при таком построении выборки в диапазоне $1,03 < X_2 < 57,4$.
3. Количество непрерывных последовательностей (англ. *run*) нулей R_0 и единиц R_1 длины K должно лежать в диапазонах, сведенных в табл. 19.1 (таким образом, данный тест состоит из 12 независимых требований).

Таблица 19.1. Диапазоны R_0 и R_1 теста № 3 стандарта FIPS 140-1

К, бит	R_0, R_1
1	2267—2733
2	1079—1421
3	502—748
4	223—402
5	90—223
6 и более	90—223

4. В тестовом блоке не должно встречаться непрерывных последовательностей нулей либо единиц, длиннее 33 бит.

Генератор псевдослучайных чисел называется *криптостойким* (КГПСЧ), если по неограниченному объему сгенерированных им данных невозможно предсказать с вероятностью, отличной от 0,5, очередной создаваемый им бит. Подобное требование приводит, во-первых, к тому что размер начального вектора ГПСЧ должен быть значительно выше возможностей полного перебора современных ЭВМ (на практике используются векторы в 128 бит и более). Во-вторых, начальный вектор никогда не должен попадать в открытом виде в выходной поток алгоритма, иначе станет возможным воссоздать цепочку генерации бит. Как видим, "общетехнические" ГПСЧ не удовлетворяют ни первому, ни второму условиям.

Современной криптографией разработаны КГПСЧ на основе блочных шифров и криптостойких хэш-функций. Свойство необратимости хэш-функции в целом и шифра относительно ключа гарантирует выполнение второго требования к криптостойкому ГПСЧ — начальный вектор используется в качестве секретного ключа. Первое же требование выполнить также нетрудно, так как длины ключей современных блочных шифров уже сравнялись с величиной в 128 бит и даже превзошли ее.

За рубежом получили широкое распространение стандартизованные схемы КГПСЧ FIPS 186 и X9.17. Но мы остановимся на отечественном стандарте, основанном на блочном шифре ГОСТ 28147-89. Начальным вектором для него является 320-битное значение, которое состоит из 256-битного ключа K и дополнительного 64-битного элемента V (в случае использования только 256 случайных бит блок V можно заполнить любым фиксированным значением или, например, текущим временем). Блок V по ходу ГПСЧ используется как "монокитно", так и в виде двух 32-битных слов $V[0]$ (младшее) и

$V[1]$ (старшее). Алгоритм генерации псевдослучайных блоков выглядит следующим образом:

$V = \text{Encrypt}(V, K)$

цикл по i от 0 до необходимого количества псевдослучайных чисел

 начало_ i

$V[0] = (V[0] + 01010101_{16}) \bmod 2^{32}$

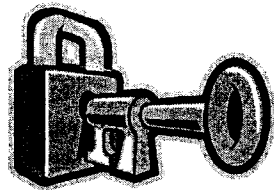
$V[1] = (V[1] + 01010104_{16}) \bmod (2^{32} - 1)$

$RND[i] = \text{EnCrypt}(V, K)$

 конец_ i

Таким образом, на каждом проходе алгоритм генерирует 64 псевдослучайных бита, период повторения последовательности $2^{32} \times (2^{32} - 1) = 2^{64}$. Стойкость к полному перебору начального вектора — 2^{320} . Отметим, что ни на каком этапе ни величина K ни величина V не появляются на выходе алгоритма в открытом виде. Полученную по подобному правилу последовательность можно применять для симметричного шифрования по схеме *гаммирования* точно так же, как для этого применяется выход с поточных шифров-скремблеров.

Глава 20



Сложные криптографические протоколы

Практически во всех случаях применения симметричных или асимметричных криптоалгоритмов они используются не "в одиночку", а как основная часть — ядро — более сложной системы защиты информации. О них и пойдет речь в данной главе.

Криптосистемы

Общие сведения

Криптосистемой именуется законченный программный, аппаратный или программно-аппаратный продукт, целью которого является объединение конкретного набора криптосхем, шифров и иных систем обеспечения защиты информации в единое целое. В функции криптосистемы может входить и дополнительная защита от атак, не связанных непосредственно с криптоанализом, а также согласование системы с внешними программными интерфейсами и обеспечение удобства пользователя. В состав криптосистемы могут входить:

- ☐ алгоритм сжатия данных перед шифрованием;
- ☐ система управления ключами и цифровыми сертификатами;
- ☐ алгоритм преобразования парольной фразы в ключ;
- ☐ алгоритм создания цепочек;
- ☐ схемы электронной цифровой подписи;
- ☐ штампы времени или порядковых номеров;
- ☐ алгоритмы транспортного и/или помехоустойчивого кодирования.

Неотъемлемой частью — ядром — любой криптосистемы является шифр: *симметричные криптосистемы* основаны на блочных или поточных шифрах, *асимметричные криптосистемы* — на асимметричных криптосхемах. Исклю-

чением являются криптосистемы, обеспечивающие только аутентичность, но не конфиденциальность сообщения — их ядром являются алгоритмы ЭЦП. Как правило, криптосистема не увеличивает стойкость самого использованного шифра, она только лишь закрывает возможность обходных атак злоумышленника — например, повторной отправки зашифрованных команд (англ. — *replay attack*) или угадывания открытого текста по повторяющимся зашифрованным блокам.

Общий вид симметричной криптосистемы



Некоторые возможные варианты симметричной криптосистемы приведены на рис. 20.1. Рассмотрим подробнее составные части этой схемы.

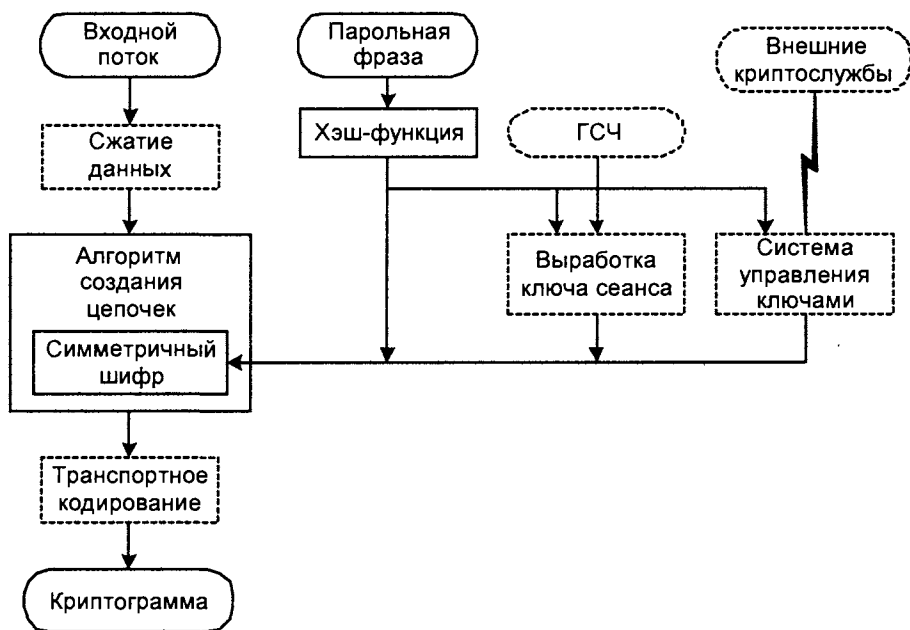


Рис. 20.1. Симметричная криптосистема

Процедура создания ключа



Очень большой простор для всевозможных вариаций предоставляет процедура создания ключа. Существуют четыре основных варианта использования ключа для шифрования.

1. *Мастер-ключ*, получаемый из *парольной фразы*, вводимой пользователем.

2. *Мастер-ключ*, получаемый с помощью *системы управления ключами*.
3. Для шифрования каждого нового сообщения система генерирует случайное или псевдослучайное число, называемое *ключом сеанса*, а сам ключ сеанса шифруется мастер-ключом пользователя и помещается в заголовок сообщения.
4. Для шифрования каждого нового сообщения система получает от *системы управления ключами* каждый раз новый *ключ сеанса* (ключ сеанса в этом варианте не добавляется к сообщению).

Второй вариант отличается от четвертого тем, что мастер-ключ, получаемый от системы управления ключами, может использоваться для шифрования нескольких (обычно очень большого количества) сообщений. А в том случае, когда от системы управления ключами блочный шифр получает ключ сеанса, он используется только однократно — для данного сообщения.

Мастер-ключ



Как видим, очень большую роль в криптосистемах играют системы управления ключами — им посвящен *разд. "Аутентификация пользователей" данной главы*, а относительно простые первый и третий варианты мы рассмотрим в этом разделе. Процесс получения мастер-ключа из некоторой секретной фразы — пароля — является естественно самым ранним и самым простым вариантом генерации ключа. Пароль представляет собой легко запоминаемую пользователем комбинацию символов произвольной длины.

Потенциально, после дополнения нулями или наоборот отсечения лишних символов пароль можно использовать непосредственно как ключ. Однако делать это категорически не рекомендуется. Основной причиной этого требования является очень сильная коррелированность битов пароля между собой. Ведь если секретная фраза является легко запоминаемой, значит, ее информационная насыщенность (энтропия) гораздо меньше объема. Криптостойкость большого количества как ушедших в историю, так и современных шифров (в том числе ГОСТ 28147-89 и AES) очень сильно зависит от того, имеются ли в ключе корреляции между битами или нет. Ведь, например, при использовании в качестве перемешивания операции умножения или сдвига на переменное число бит заранее известные биты ключа ($0011xxxx_2$ для цифр, $010xxxxx_2$ для заглавных латинских букв, $011xxxxx_2$ для строчных латинских букв и т. д.) дадут предсказуемый результат.

Кроме того, отбрасывание лишних символов пароля также недопустимо. Современные исследования приводят следующие цифры информационной насыщенности символов (табл. 20.1).

Таблица 20.1. Приблизительная информационная емкость символов

Характер текста	Энтропия, бит
Непроизносимое сочетание символов и цифр в обоих регистрах	≈ 6
Произносимое, но бессмысленное сочетание символов	≈ 4
Осмысленное слово, связный текст	2–3

Предположим, что используется 128-битный секретный ключ. Тогда при вводе 16 символов (которые и дадут 128 бит) энтропия пароля составляет всего 40 бит — ни о какой стойкой криптографии говорить не приходится. Шифр будет взломан при наличии всего одной пары открытый/зашифрованный блок по словарю.

Универсальным решением как первой, так и второй проблемы является проведение парольной фразы через хэш-функцию. Результат хэширования имеет очень хорошие статистические характеристики — в нем отсутствуют явные корреляции бит, и все биты и их комбинации имеют равномерный закон распределения. На вход хэш-функции подается последовательность символов неограниченной длины, следовательно, ограничения на длину пароля исчезают. До того момента, пока информационная емкость парольной фразы не превышает размер результата хэш-функции, алгоритм хэширования практически не потеряет информацию.

То есть из-за обрезания пароля при непосредственном использовании на длине в 16 символов максимальная информационная емкость ключа не сможет превысить 40 бит (считаем наполнение 2,5 бита на символ для связного текста). А вот при использовании хэш-функции 16-символьный пароль создаст ключ емкостью в 40 бит, 30-символьный — в 75 бит, а 45-символьный — в 110 бит.

Конечно, использовать пароль с энтропией больше длины ключа бессмысленно, но стараться как можно ближе свести эти две величины — это одно из первых условий стойкости всей системы. Для достижения энтропии осмысленной парольной фразы в 128 бит ее длина должна быть не менее 50 символов, для бессмысленного набора символов — 20 символов. Однако следует заметить, что комбинацию из 20 произвольных символов рядовой пользователь запомнить уже не может — тем самым мы вынуждаем его записывать пароль на бумажный или электронный носитель. А в результате вся изощренная система защиты данных подвергается гораздо большей угрозе. Поэтому все-таки в случае необходимости запоминать пароль — лучше использовать длинную, но разумную фразу.

Кстати, набор русского текста при включенной латинской раскладке клавиатуры спасает только в том случае, если ваш противник — иностранец.

Для него и любой иноязычной программы перебора по словарю введенные символы действительно представляют полную бессмыслицу с энтропией около 6 бит на символ, и соответственно, увеличивают время перебора в миллиарды раз. Но от русскоязычных программ и специалистов этот трюк не спасет — для них подобная уловка добавляет всего лишь 1 бит к энтропии сообщения (время перебора возрастает только вдвое). Все в мире относительно, даже информация.

Есть еще один, нетрадиционный, но встречающийся на практике способ защититься от атаки полного перебора пароля. Он состоит в увеличении времени преобразования "пароль >> ключ" до нескольких секунд. Как один из вариантов достижения такого "медленнодействия" операция хэширования повторяется над паролем по нехитрой схеме $tmp[k] = \text{hash}(pass \parallel tmp[k-1])$ порядка миллиона раз. Как следствие, пользователю приходится немного подождать в процессе шифрования и дешифрования. Но это ожидание ничтожно мало по сравнению с тем, сколько придется ожидать злоумышленнику для перебора тех же 2^{40} вариантов пароля по словарю — 30 000 машино-лет.

Ключ сеанса



Технология *сеансовых ключей* позволяет обеспечивать дополнительную стойкость криптосистеме, так как каждый ключ используется только однократно — для шифрования одного конкретного послания. Если в результате криптоатаки злоумышленник получает значение ключа сеанса, раскрытым оказывается только один документ — у абонентов не появляется необходимость экстренно обмениваться новым мастер-ключом по защищенному каналу. Ключ сеанса может быть получен с помощью системы управления ключами, т. е., например, выслан сервером ключей или сгенерирован по какой-либо криптостойкой формуле.

Но есть и самый простой метод, который заключается в генерации ключа сеанса с помощью криптостойкого ГСЧ или ГПСЧ. На созданном локально из случайной величины ключе сеанса производится шифрование документа, затем возникает необходимость сообщить этот ключ принимающей стороне. Ведь в случае создания случайного ключа "на месте" его знает пока только отправитель (в случае использования системы управления ключами получатель может выяснить значение ключа сеанса по другому каналу). Наиболее простой метод — зашифровать ключ сеанса мастер-ключом и поместить в заголовок всей шифрограммы. При этом ключ сеанса имеет хорошие статистические характеристики и очень малый объем. Поэтому даже потенциальное раскрытие ключа сеанса (например, на основе атаки по выбранному открытому тексту) приносит злоумышленнику только еще одну пару блоков "открытый/зашифрованный текст" для атаки на мастер-ключ (а это очень и очень мало даже для самых не требовательных к ресурсам перехваченных сообщений криптоатак).

У технологии сеансовых ключей есть еще одно положительное свойство — рандомизация сообщения. Один и тот же документ или, что важнее, одна и та же краткая команда или телеметрические данные от объекта будут зашифровываться каждый раз все в новый и новый шифротекст. У злоумышленника исчезнет возможность догадываться о характере сообщений, наблюдая за эффектом, который производят эти сообщения на систему-получатель. Конечно, существуют и другие способы рандомизации сообщений, однако ключи сеанса позволяют достичь очень хорошего эффекта как "побочный результат" дополнительной криптостойкости (либо же наоборот). Не следует только забывать о появлении в таких системах опасности воздействия злоумышленника на генераторы случайных величин. За любое улучшение качества приходится платить, и в данном случае — усилением контроля за стойкостью и информационной насыщенностью порождаемых случайных значений.

Алгоритмы создания цепочек



Следующей очень важной частью любой симметричной криптосистемы являются алгоритмы создания цепочек (англ. *chaining modes*). Они определяют закон, по которому производится шифрование блочным шифром (который обладает фиксированной длиной шифруемого блока) текста произвольной длины. Двумя основными техническими вопросами, которые призваны решать алгоритмы создания цепочек, является то, как связываются соседние блоки текста, и то, как шифруется последний неполный блок текста. Однако, как оказалось, кроме этих чисто технических вопросов, с данной темой связаны и некоторые аспекты стойкости ко взлому всей системы — т. е. при проектировании схем создания цепочек пришлось учитывать и мнение криптоаналитиков.

Самым простым методом шифрования текста произвольной длины является разбиение его на последовательные блоки длины, равной длине блока шифрования криптоалгоритма, и проведение над каждым блоком независимых криптопреобразований. Эта схема приведена на рис. 20.2, она получила наименование *ECB (Electronic Code Book)* из-за своего применения в самых простых шифрующих электронных устройствах. У нее есть очень существенный недостаток — при шифровании двух одинаковых блоков на выходе получаются два одинаковых блока шифротекста. Чаще всего такая ситуация имеет место на длинных строках из одинаковых символов — пробелов, точек, подчеркиваний, переводов строк. Очевидно, что данная информация может сообщить злоумышленнику достаточно много о структуре зашифрованного документа. Кроме того, если открытый текст является данными телеметрии или командами дистанционного управления, то при возможности наблюдения за объектом злоумышленник также может частично предугадывать значения шифровок, не взламывая шифр непосредственно.

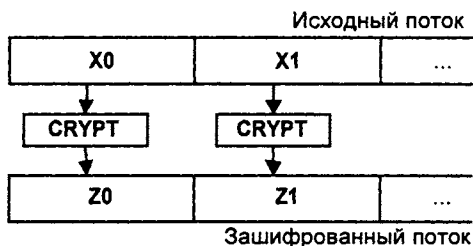


Рис. 20.2. Схема ECB

Для решения этих проблем были предложены чрезвычайно простые (состоящие из одной-двух дополнительных операций) обратимые схемы, связывающие воедино весь шифруемый документ. В 1980 году три подобные схемы были стандартизованы в США, две из них получили в дальнейшем очень широкое распространение. Все три алгоритма приведены на рис. 20.3. В схеме *CBC* (*Cipher Block Chaining*) предыдущий шифроблок накладывается операцией исключающее ИЛИ на открытый текст и только затем производится его шифрование. В схеме *CFB* (*Cipher FeedBack*) блочное шифрование и операция исключающее ИЛИ меняются местами — сначала производится шифрование предыдущего шифротекста, затем результат накладывается на открытый текст исключающим ИЛИ. Именно эти две схемы получили наибольшее распространение. Схема *OFB* (*Output FeedBack*) представляет собой шифрование по схеме гаммирования — начальный вектор последовательно шифруется множество раз, а значения, получающиеся после каждого шифрования, накладываются с помощью XOR на открытый текст.

Все три схемы однозначно обратимы. Все три решают проблему появления в открытом тексте огромных последовательностей одинаковых блоков — период повторения шифротекста в этом случае сравним с 2^N , где N — длина блока криптоалгоритма. При этом, если в качестве начального значения вспомогательной переменной (англ. *Initial Vector* — *IV*) выбирается фиксированное значение, например, блок из N нулей, то одинаковые документы будут преобразовываться в одинаковый шифротекст. Если же начальный вектор *IV* для каждого документа будет генерироваться с помощью ГСЧ, ГПСЧ или, как в некоторых реализациях, присылаться удаленной стороной, то алгоритмы создания цепочек будут выполнять еще и роль рандомизаторов процесса шифрования. В случае генерации начального вектора на передающей стороне необходимо присоединять его к шифруемому документу (в открытом виде).

Несмотря на схожесть, каждая схема создания цепочек имеет свои специфические характеристики. Так, например, схемы CFB и OFB позволяют помещать в выходной поток ровно столько бит, сколько было в исходном сообщении. А вот алгоритмы ECB и CBC требуют расширения открытого сообщения до длины, кратной длине блока, и указания каким-либо образом дешифрующей стороне, сколько же "лишних" бит добавлено.

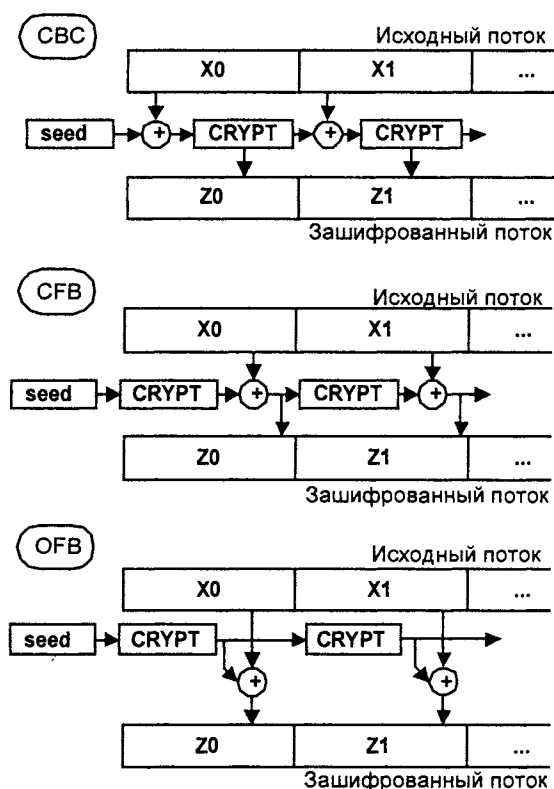


Рис. 20.3. Схемы CBC, CFB, OFB

Когда исследованию был подвергнут эффект, который оказывает на зашифрованный текст изменение одного или нескольких бит (результат сбоя в канале связи или искажения злоумышленником), выяснилась следующая картина (табл. 20.2).

Таблица 20.2. Характер влияния искажений на дешифрование

Алгоритм создания цепочек	Эффект искажения одного блока шифротекста на текст после расшифровки
ECB	Полностью повреждается модифицированный блок
CBC	Полностью повреждается модифицированный блок, инвертируются измененные биты в следующем блоке
CFB	Инвертируются измененные биты в модифицированном блоке, полностью повреждается следующий блок
OFB	Инвертируются измененные биты

Как следствие, схема OFB категорически не должна применяться в каналах, незащищенных от модификации злоумышленником. Иначе, скажем, изменение суммы банковского электронного денежного перевода с 200 000 на 300 000 денежных единиц пройдет совершенно незамеченным — взломщику всего лишь нужно будет точно знать номера байт в сообщении, на которые приходятся цифры денежной суммы. Схемы CFB и CBC имеют примерно одинаковую стойкость к подобным махинациям. Для них все зависит от того, сможет ли дешифрующая сторона эффективно обнаружить полностью поврежденный блок — признак повреждения одного или нескольких бит в другом блоке. При этом в схеме CFB возможна необнаруживаемая фальсификация в последнем блоке, так как полностью повреждаемый следующий блок отсутствует.

Очевидно, что свойства схем создания цепочек размножить специфическим образом ошибки в передаче балансируют между обнаружением модификаций и восстановлением данных. В некоторых случаях, основной целью системы является гарантированное обнаружение любой модификации потока данных (умышленной или случайной). В других случаях наоборот, пользователи рады смириться с потерей нескольких бит, лишь бы поврежденная в результате ошибки передачи часть данных была как можно меньше. Второй цели, как никакая другая, удовлетворяет схема OFB (и вообще любой симметричный шифр гаммирования). А вот для целей гарантированного обнаружения модификаций (без применения специфичных методов аутентификации) недавно была предложена схема *IGE (Infinite Garble Extension)* — схема неограниченного размножения ошибки (рис. 20.4). В рекуррентной формуле алгоритма IGE, задействован не только предыдущий блок шифротекста, но и предыдущий открытый блок. В результате этого модификация единственного бита шифротекста полностью повреждает текущий блок и все последующие. При наличии в передаваемом тексте хотя бы минимальной избыточности схема IGE достаточно надежно защищает от модификации зашифрованного потока без каких-либо дополнительных мер, подобных ЭЦП.

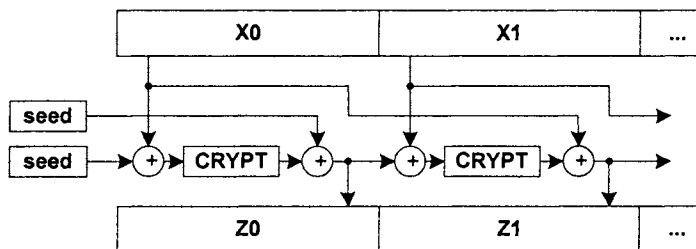


Рис. 20.4. Схема IGE

Асимметричные криптосистемы



Асимметричные криптосистемы в качестве своего ядра используют алгоритмы шифрования на открытом ключе. Основной причиной, требующей усложнения процесса шифрования, является чрезвычайная медленность асимметричных шифров. При одинаковой стойкости шифрование блока данных на открытом ключе в тысячи раз медленнее симметричного шифра. В связи с этим была предложена очень простая типовая схема асимметричного шифрования, изображенная на рис. 20.5.

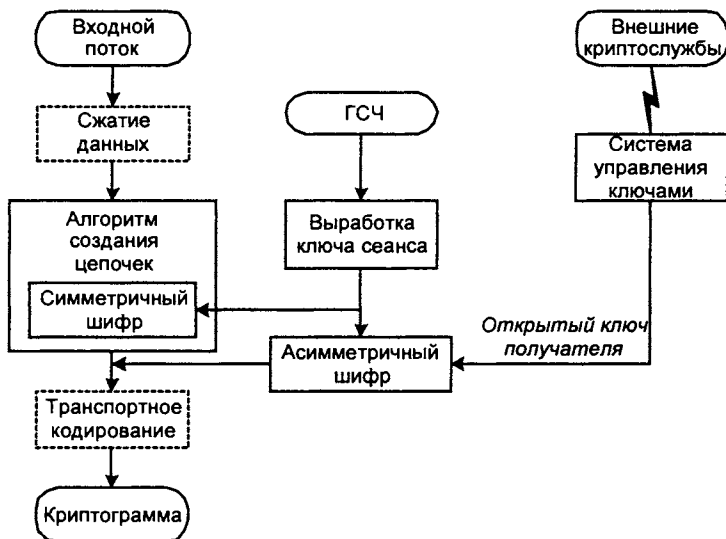


Рис. 20.5. Асимметричная криптосистема

На открытом ключе шифруется только ключ сеанса, непосредственно перед этим сгенерированный на передающей стороне. Весь же текст сообщения шифруется любым блочным шифром с помощью ключа сеанса и какой-либо схемы создания цепочек. Сообщение в целом состоит из зашифрованного ключа сеанса и собственно основной шифрограммы. Как и в любой другой криптосхеме, ключ сеанса после проведения шифрования тщательно уничтожается, а криптограмма отправляется получателю. На приемной стороне с помощью закрытого ключа дешифруется ключ сеанса, а затем уже и все сообщение.

Прочие компоненты криптосистем



К вспомогательным компонентам криптосистем относятся алгоритмы компрессии данных, отметки времени, порядкового номера, транспортного и помехоустойчивого кодирования. Из этого списка к дополнительным мерам

безопасности переписки можно отнести предварительное сжатие данных и вставку отметки времени (англ. *timestamp*) или порядкового номера сообщения.

Сжатие данных

Сжатие данных увеличивает информационную плотность входного потока данных и, как следствие, улучшает его статистические характеристики. В отношении атак по шифротексту и по известному открытому тексту это, несомненно, играет против злоумышленника.

В отношении атаки по выбранному открытому тексту, когда злоумышленник имеет возможность еще и модифицировать поток перед шифрованием, сжатие данных также в принципе играет против него, но здесь есть и "подводные камни". Конечно, сжатие является своего рода сверткой всего предыдущего текста, и тем самым значительно усложняет процесс подбора документа, который в противном случае давал бы желаемые двоичные значения на желаемых позициях. Однако эта свертка отнюдь не необратима, и с большой долей вероятности желаемый текст все же удастся создать. Минусом применения компрессии в случае атаки по выбранному открытому тексту является то, что у злоумышленника появляется возможность генерировать двоичные значения из всего диапазона от 00_{16} до FF_{16} , а не только из печатных символов кода ASCII. А это иногда бывает необходимо на той или иной стадии криптоанализа шифра.

В современной вычислительной технике применяются три класса универсальных алгоритмов сжатия: компрессия *алфавитного уровня*, *фонемного уровня* и *словарного уровня*. Кроме этого, существует множество узкоспециальных методов, например, сжатие аудио- и видео изображений. Несомненно, выбор алгоритма компрессии должен основываться на характере шифруемых сообщений.

Алгоритмы алфавитного уровня основаны на том факте, что некоторые символы из потенциального (обычно — 256-символьного) алфавитного набора в произвольном потоке данных могут встречаться чаще среднего периода повтора, а другие, соответственно, — реже. Следовательно, если для записи распространенных символов использовать короткие последовательности бит, длиной меньше 8, а для записи редких символов — длинные, то суммарный объем потока уменьшится.

Алфавитный алгоритм сжатия Хаффмана (*код Хаффмана*) задает закон выбора последовательностей бит следующим образом. Пусть у нас имеется список всех символов, встречающихся в исходном тексте, причем известно количество появлений каждого символа в нем. Выпишем их вертикально в ряд в виде ячеек будущего графа по правому краю листа (рис. 20.6, а). Выберем два символа с наименьшим количеством повторений в тексте (если три или большее число символов имеют одинаковые значения, выбираем

любые два из них). Проведем от них линии влево к новой вершине графа и запишем в нее значение, равное сумме частот повторения каждого из объединяемых символов (рис. 20.6, б). Отныне не будем принимать во внимание при поиске наименьших частот повторения два объединенных узла (для этого сотрем числа в этих двух вершинах), но будем рассматривать новую вершину как полноценную ячейку с частотой появления, равной сумме частот появления двух соединившихся вершин. Будем повторять операцию объединения вершин до тех пор, пока не придем к одной вершине с числом (рис. 20.6, в и г). Для проверки: очевидно, что в ней будет записана длина кодируемого файла.

У нас получился связный граф без циклов, то есть дерево с корнем в последней созданной вершине. Оно носит название *дерево Хаффмана*. Расставим на двух ребрах графа, исходящих из каждой вершины, биты 0 и 1 произвольно — например, на каждом верхнем ребре 0, а на каждом нижнем — 1. Для определения кода каждой конкретной буквы необходимо просто пройти от корня дерева до нее, выписывая нули и единицы по маршруту следования. Для рисунка 20.6 символ "А" получает код "100", символ "Б" — код "11", символ "К" — код "101", а символ "О" — код "0".

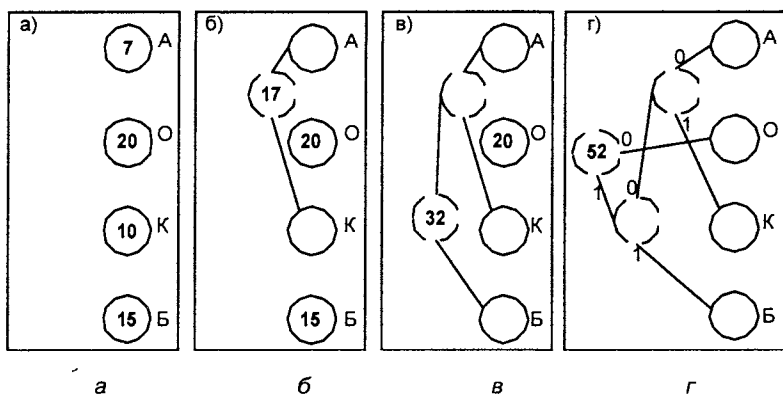


Рис. 20.6. Построение дерева Хаффмана

Код Хаффмана является *префиксным*, т. е. код никакого символа не является началом кода какого-либо другого символа. Из этого следует, что текст, преобразованный по данному алгоритму, несмотря на переменную длину символов, однозначно восстановим получателем, даже если не сообщается длина кода каждого переданного символа. Получателю пересылают только дерево Хаффмана в компактном виде, а затем входная последовательность кодов символов декодируется им самостоятельно без какой-либо дополнительной информации. Например, при приеме "0100010100001" для данного примера сначала отделяется первый символ "Б": "01-00010100001", затем снова начиная с вершины дерева — "А" "01-000-10100001", далее аналогично декодируется вся запись "01-000-1-01-000-01" = "БАОБАБ".

Классический словарный алгоритм *Лемпеля-Зива* (*Abraham Lempel и Jakob Ziv*) — *LZ77*, названный так по году своего опубликования, предельно прост. Он формулируется следующим образом: *"Если в прошедшем ранее выходном потоке уже встречалась подобная последовательность байт, причем запись о ее длине и смещении от текущей позиции короче, чем сама эта последовательность, то в выходной файл записывается ссылка (смещение, длина), а не сама последовательность"*. Так, например, фраза "колокол_около_колокольни" закодируется как "коло(- 4, 3)_(- 5, 4)о_(- 14, 7)ьни".

Распространенный метод сжатия фонемного уровня *RLE* (англ. *Run Length Encoding*) заключается в записи вместо последовательности одинаковых символов одного символа и их количества. Например, последовательность "AAAAAAA" с помощью алгоритма RLE будет закодирована как "(A, 7)".

В тех случаях, когда о сжимаемом потоке нет предварительной информации, чаще всего используют *универсальные методы* словарного уровня — например, семейство *LZH* (по первым буквам фамилий исследователей: *Lempel, Ziv, Huffman*). Алгоритмы этого класса объединяют предварительную обработку текста словарными методами с последующим сжатием всех оставшихся символов и появившихся ссылок алфавитными методами. Таким образом, при наличии в тексте избыточности хотя бы одного типа универсальные методы уже получают выигрыш при сжатии.

Транспортное кодирование

Необходимость *транспортного кодирования* на выходе, которое превращает поток символов диапазона $[00_{16}...FF_{16}]$ в поток большего объема, но состоящий только из печатных символов ASCII, возникла из-за того, что в подавляющем большинстве случаев криптосистемы являются прозрачной и, более того, опциональной прослойкой систем передачи данных. Таким образом, если система проектировалась для передачи текстовых сообщений, например, по электронной почте или IRC-каналам, то потребовавшееся на определенном этапе встраивание криптопрослойки не должно нарушать соглашений относительно разрешенного в тексте набора символов (рис. 20.7). Не должны появляться лишние переводы строк, служебные символы, символы расширенного набора. В системах, определяющих конец текста по набору данных (например, по двум последовательным переводам строки), и не хранящих отдельным полем длину сообщения, не должно случайно получиться "фальшивого" признака конца текста посередине шифровки.

Алгоритм, выполняющий прямое преобразование, называется кодером, выполняющий обратное преобразование — декодером. Сами схемы кодирова-

ния традиционно носят второе название — *кодек* (сокращение по нескольким первым буквам: кодер/декодер).

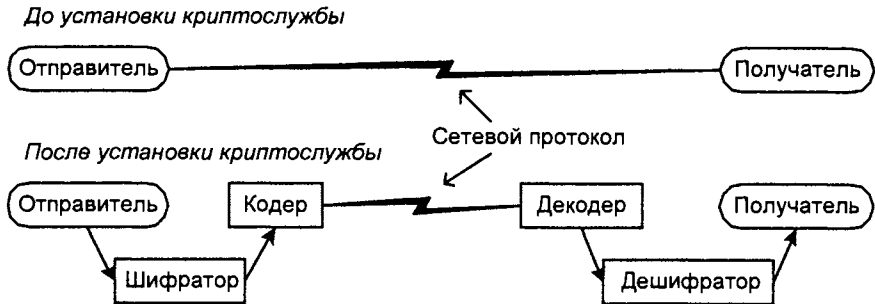


Рис. 20.7. Схема установки транспортного кодака

Среди множества алгоритмов *транспортного кодирования* в криптографии наиболее прижился кодек *Base64*, стандартизованный в [RFC-1341]. Входной поток символов последовательно разбивается на блоки по три байта (24 бита), каждый из которых затем интерпретируется как четыре 6-битных числа. Эти числа используются как индексы в таблице из 64 разрешенных символов, включающей заглавные и строчные буквы латиницы, числа и несколько знаков (полностью таблица приведена в *приложении 12*). Соответствующие ASCII символы записываются в той же последовательности в выходной поток, коэффициент расширения длины получается равным 1,33. Не кратный трем байтам последний блок дополняется нулевыми битами либо до 12, либо до 18 бит, а в качестве оставшихся двух или одного завершающих ASCII символов записываются знаки равенства ("="). По этим финальным символам схему кодирования Base64 часто очень легко отличить. Пример преобразований в данном кодеке приведен на рис. 20.8.

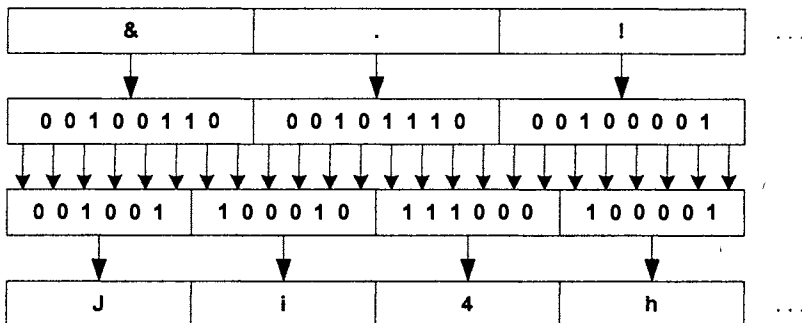


Рис. 20.8. Пример кодирования алгоритмом Base64

Системы управления ключами

Функции и классификация

Управление ключами (УК) является настолько важной и развитой областью криптографии, что требует отдельного и детального рассмотрения. На системы УК возлагается огромный набор различных функций, обеспечение самых разных базовых и вновь приобретенных свойств криптосистем, которые ими укомплектованы. Подобные схемы могут выполнять хранение, пересылку, шифрование (то есть обеспечение конфиденциальности), аутентификацию, "сдачу на хранение" (депонирование) и разделение ключей. Единственным общим свойством систем УК является то, что как результат разнообразных трансформаций они должны снабдить криптосистему ключом (симметричным или асимметричным), на котором и будет произведен основной процесс шифрования документа. Техническая реализация систем управления открытыми ключами (англ. *PKI — Public Key Infrastructure*) описана в *главе 22*.

В зависимости от того, какой тип ключа генерирует в итоге система УК, производится их деление на *системы управления симметричными ключами* и *системы управления асимметричными ключами*. Системы управления симметричными ключами делятся в свою очередь на *системы с наличием начальных мастер-ключей* и *системы с нулевой начальной информацией*. Как отдельный материал рассмотрены системы депонирования ключей и системы разделения секрета. К сожалению, данный раздел не может охватить даже половины различных схем УК и криптографических протоколов на их основе — на сегодняшний день исследователями разработано более сотни различных схем. Все чаще и чаще встречающееся сейчас введение третьего субъекта криптоопераций — *доверенных лиц* с различными функциями и полномочиями — породило целую волну протоколов, обеспечивающих новые свойства криптосистем (апеллируемость, подтверждение даты/времени подписания, депонирование ключей и т. п.). Рассмотрим только несколько наиболее интересные и часто применяемых на практике схемы.

Системы управления симметричными ключами



С предварительной частичной установкой

Все системы управления симметричными ключами вне зависимости от того, сколько участников задействовано в процессе, классифицируются в первую очередь на системы, в которых между субъектами уже установлены защищенные каналы (то есть присутствуют секретные мастер-ключи), и на системы, в которых этого канала нет. В первом случае основной целью систе-

мы УК является либо генерация ключей сеансов, либо обновление ключевой информации, либо, что чаще всего требуется, обмен секретным ключом между двумя абонентами, которые до этого напрямую подобного ключа не имели, хотя цепочка доверенных связей (например, через общего знакомого) уже существовала. Во втором случае, когда два пользователя не обладают никакой общей секретной информацией, им необходимо установить ключ, причем таким образом, чтобы прослушивающий весь обмен сообщениями злоумышленник не смог создать свою "третью" копию ключа.

Случай, когда два абонента уже могут общаться между собой по защищенному каналу, и при этом желают обменяться "свежей" ключевой информацией, на самом деле не содержит практически никаких тонкостей. Необходимо только уделить внимание невозможности перехвата злоумышленником перехваченного ранее пакета с такой же информацией. Для этого в систему вводятся автоинкрементные счетчики и/или штампы даты/времени.

Несколько больший интерес представляют системы с тремя субъектами: двумя абонентами и доверенным сервером ключей. В этом случае считается, что сервер обслуживает большое количество равноправных клиентов, с каждым из них у сервера уже установлен защищенный канал связи, т. е. обмен ключами произошел. Теперь на определенном этапе функционирования системы у двух клиентов сервера, никогда до этого не общавшихся, возникло желание обменяться некоей приватной информацией.

В подобной ситуации классический протокол установления ключа сеанса выглядит примерно следующим образом — рис. 20.9. Вызывающий абонент обозначен А, вызываемый абонент — В, доверенный сервер — S, ключ которым априори обменялись А и S — "A-S", ключ между абонентом В и S — "B-S".

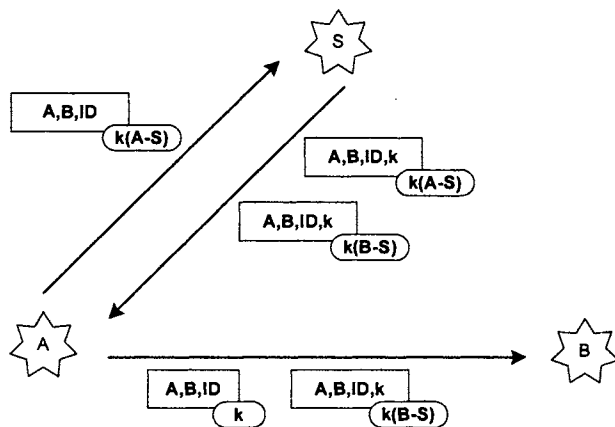


Рис. 20.9. Запрос коммуникационного ключа у доверенного сервера

1. Вызывающий абонент А высылает в адрес сервера сообщение, содержащее имя вызываемого абонента В и некоторое случайное число, идентифицирующее номер соединения.

2. Сервер генерирует с помощью ГСЧ или криптостойкого ГПСЧ ключ сеанса k для данных двух абонентов и возвращает А пакет, содержащий две шифрограммы: в первом, зашифрованном секретным ключом А-S, ключ сеанса, номер сессии и имя вызываемого абонента (для подтверждения), во втором, зашифрованном ключом В-S, — также ключ сеанса k , и имя вызывающего абонента.
3. Первая шифрограмма из ответа сервера предназначена А — он может ее прочесть ключом А-S, но вот прочесть вторую он не может — вторая половинка предназначена для пересылки в неизменной форме абоненту В.

Несомненно вторую шифрограмму из ответа доверенного сервера он мог непосредственно отправить в адрес абонента В, и в результате обе стороны обладали бы одинаковым набором информации. Однако выделение ключа сеанса в отдельный пакет, зашифрованный ключом В-S, дает большую свободу действий вызываемому абоненту. Получив данное разрешение, он может использовать его не непосредственно, а с паузой, например, в течение несколько дней. В этой связи вторая половинка ответа сервера получила название *билет*, и играет практически ту же самую роль, что и настоящий билет. В приведенной схеме (см. рис. 20.9) объем передаваемой информации сокращен до минимума. Наиболее часто подобная (*базовая*) схема дополняется информацией о дате выдачи билета, его сроке действия, полномочиях вызывающего абонента и другой аналогичной информацией. Все эти поля заполняются сервером в момент выдачи билета, заносятся в журнал выдачи билетов и содержатся в билете, конечно же, в зашифрованном на ключе В-S виде.

В случае сбоя в канале передачи данных в момент обмена ключевой информацией или же при наличии злоумышленника, который пытается выдать себя за абонента А, либо абонента В, легитимные пользователи получают вместо синхронного ключа сеанса некую случайную последовательность. Факт различия ключей сеанса у вызывающего и вызываемого пользователей вскроется на одном из первых пакетов, тем более если применяется помехоустойчивое кодирование или передается избыточная информация. Однако, для того чтобы не нагружать более высокие уровни обработки данных ненужными функциями проверки, проверку соответствия ключей, полученных по подобной схеме, возлагают на сам криптопротокол. В этом случае его дополняют двумя дополнительными сообщениями: от А к В и от В к А. Оба сообщения содержат обычно имена абонентов, штамп времени, идентификатор сессии (если он доступен обоим сторонам). Оба сообщения шифруются только что полученным ключом сеанса k . Формируется подобие преамбулы процесса защищенной передачи основного блока данных.

Описанный протокол был использован в качестве основы для алгоритма распространения ключей *Kerberos*. Отличие полного протокола Kerberos, получившего наибольшее распространение на сегодняшний день, от приведенной базовой схемы заключается только в наличии двух уровней работы с

серверами. Первый сервер, к которому обращается вызывающий абонент, носит название *сервер аутентификации* (англ. *Authentication Server — AS*). Билет, полученный от подобного сервера, имеет обычно больший срок действия и называется *билет на получение билетов* (англ. *ticket granting ticket — TGT*). Второй уровень серверов — *серверы выдачи билетов* (англ. *ticket granting server — TGS*) — выполняет уже функции непосредственного разрешения доступа к определенным ресурсам посредством выдачи *билетов на ресурс*. Подобная схема позволяет значительно разгрузить сервера аутентификации, но при этом оставить "всю полноту власти" в едином центре. Общий вид схемы Kerberos приведен на рис. 20.10. Подробнее протокол Kerberos описан в *главе 22*.

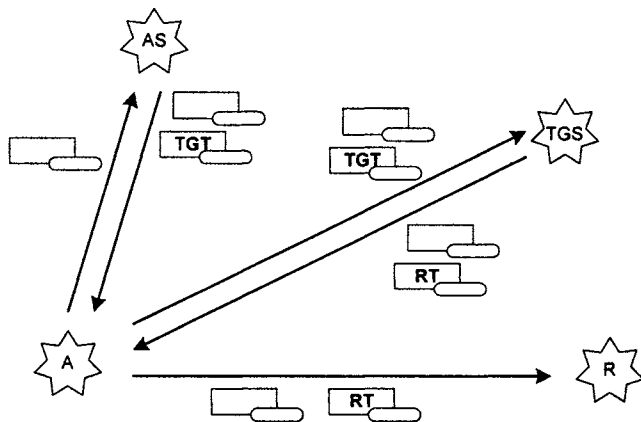


Рис. 20.10. Схема Kerberos

Без предварительной установки

Схемы обмена (выработки) секретного ключа между двумя абонентами, не имевшими до этого защищенного канала, занимают отдельную и очень интересную нишу в современной криптографии. Задача у протоколов этого класса — создать на основе сообщений, передаваемых по открытым каналам связи, у двух абонентов идентичную ключевую информацию, причем сделать это так, чтобы любое третье лицо, законспектировавшее все прошедшие переговоры, этого ключа получить не смогло. На первый взгляд задача кажется совсем неразрешимой, но методы достижения этой цели существуют и при этом весьма надежны. Основой для подобных протоколов являются схемы, очень напоминающие асимметричное шифрование.

Первая схема выработки секретного ключа носит название *схемы Шамира* по фамилии ее создателя. Первоначально все стороны, участвующие в обмене ключей по данной схеме, договариваются о большом простом числе p (не являющимся секретом), обладающим теми же свойствами, что и в схеме

Эль Гамаль. Любые два абонента (А и В), желающие создать секретный ключ сеанса:

1. Создают соответственно два больших случайных числа (a и b), а также их инверсии по модулю p ($a^{-1} \bmod p$ и $b^{-1} \bmod p$) и держат их на своих системах в секрете.
2. Вызывающая сторона генерирует ключ сеанса k ($k < p-2$) и возводит его в степень a по модулю p , после чего отправляет полученное выражение вызываемому абоненту: $M1 = ka \bmod p$.
3. Вызываемая сторона возводит полученное сообщение в степень b и отправляет обратно: $M2 = (M1b \bmod p) = (kab \bmod p)$.
4. Вызывающая сторона дешифрует полученное число инверсией числа a и отправляет обратно: $M3 = (M2 \cdot a^{-1} \bmod p) = (kb \bmod p)$.
5. Наконец, абонент В дешифрует последнее сообщение инверсией числа b и получает желаемый ключ сеанса: $k = (M3 \cdot b^{-1} \bmod p) = (k \bmod p)$.

Для обратимости схемы Шамира требуется, чтобы числа a и b были взаимно просты с $(p-1)$. В целом схема основана на проблеме дискретного логарифма и имеет ту же вычислительную сложность, что и схема Эль Гамаль. Для ее стойкости достаточно выбрать большую разрядность чисел p , k , a , b , качественно генерировать случайные числа a и b и тщательно стирать их из оперативной памяти после использования.

Еще более красивая схема обмена ключами была предложена У. Диффи (*W. Diffie*) и М. Хеллманом (*M. E. Hellman*). Она очень напоминает схему Шамира, также основана на проблеме дискретного логарифма, и, соответственно, также требует предварительной договоренности о большом простом числе p (известном всем остальным желающим). Единственным дополнением является необходимость выбора (также открытого) образующего элемента g .

1. Абоненты А и В генерируют два больших случайных числа, a и b соответственно, и держат их в секрете.
2. Абонент А вычисляет и передает В значение ($g^a \bmod p$), абонент В точно также вычисляет и высылает ($g^b \bmod p$) — эти два действия никак не связаны и могут проводиться одновременно.
3. Вызывающий абонент "довозводит" полученное значение в степень a , получая ($g^{ba} \bmod p$), а вызываемый — в степень b , получая ($g^{ab} \bmod p$).

В силу свойств возведения в степень по модулю простого числа два полученных на последнем шаге значения равны между собой — следовательно, абоненты получили общую секретную информацию ($g^{ab} \bmod p$), не известную никому другому. На ее основе вырабатывается ключ сеанса, например, отсечением лишних бит, либо выполнением операции хэширования $k = f(g^{ab} \bmod p)$.

Характеристики обеих приведенных схем — Шамира и Диффи-Хеллмана — практически идентичны. В алгоритме Диффи-Хеллмана необходимо передавать меньшее количество сообщений на этапе установки ключей. Схема Шамира имеет преимуществом возможность не только генерировать, но и передавать ключ сеанса, т. е. может выполнять функции *транспорта ключей*. Обе схемы уязвимы в случае наличия в сети *злоумышленника-посредника* (англ. *man-in-the-middle*), этим термином называют третье лицо, который в силу обстоятельств может не только прослушивать, но и модифицировать сообщения в канале.

В этом случае он начинает следующую игру: для вызывающего абонента он представляется вызываемым абонентом, для вызываемого наоборот — вызывающим. Для этого он генерирует собственные пары случайных чисел a' и b' , в момент установления ключа при общении с абонентом А он оперирует числом b' , при общении с В — числом a' . В результате вместо желаемого секретного ключа сеанса k в этой трехточечной системе создается два ключа сеанса k'_A и k'_B , и оба эти ключа известны злоумышленнику-посреднику. В последствии каждое проходящее через него в ту или иную сторону сообщение он сначала дешифрует, читает и даже, если ему это нужно, модифицирует, а затем зашифровывает снова на другом ключе сеанса. Конечно, в обязанности злоумышленника входит и уничтожение всех настоящих сообщений — ни одно из них не должно дойти напрямую от А к В или от В к А. В результате оба абонента общаются по секретному, как им кажется, каналу, а в действительности вся их корреспонденция будет прочтена.

Атака злоумышленником-посредником в целом является очень серьезной "головной болью" всех технологий, основанных на асимметричном шифровании (см. главу 12). Это остро поднимает вопрос не только о конфиденциальности процесса передачи данных, но и о четкой аутентификации удаленной стороны на всех этапах диалога, претендующего на секретность.

Системы управления асимметричными ключами



Асимметричная криптография, которая, казалось бы, решила проблему конфиденциальности сообщений без предварительной передачи секретного ключа по защищенному каналу, оказывается, всего лишь перенесла эту проблему в несколько иную область. При поверхностном взгляде на асимметричную систему кажется — "ищи в сети открытый ключ получателя, шифруй им сообщение и — конфиденциальность достигнута". Но вот тут как раз появляется *злоумышленник-посредник* — это он гипотетически мог расположить на множестве серверов в сети свой открытый ключ под именем абонента-получателя и свой почтовый адрес. В дальнейшем при получении любого письма он дешифрует его своим закрытым ключом, читает и пересылает истинному получателю, зашифровав уже на настоящем откры-

том ключе, который он действительно знает. Не спасают при этом и схемы ЭЦП, если злоумышленник подменил открытые ключи как отправителя, так и получателя. Эти соображения приводят к тому, что предварительный защищенный канал все-таки необходим — для передачи открытого ключа и почтового адреса или хотя бы какого-либо подтверждающего блока данных (например, хэш-суммы открытого ключа).

Однако, асимметричные технологии сделали гораздо больший прорыв в схемах распространения ключей, чем симметричные — были изобретены *сертификаты*. Сертификатом называется блок информации, содержащий данные, уникально идентифицирующие абонента, его открытый ключ и транспортный адрес, причем этот блок информации подписан с помощью ЭЦП другого лица. Абонент, о котором идет речь в сертификате, называется *владельцем ключа*, субъект сети, поставивший подпись под сертификатом — *заверяющим лицом* (в законе РФ "Об электронно-цифровой подписи" — *удостоверяющий центр*). Предположим, абонент А никогда не общался с абонентом С и не может проверить подлинность его открытого ключа, но и А и С общались с неким абонентом В — тогда В может выступить заверяющим лицом и подписать сертификат на владельца ключа С. Тогда абонент А, получив сертификат и проверив подпись В, в чьем открытом ключе он уверен, может отныне полагаться и на открытый ключ абонента С.

В чем же заключается тот самый "прорыв" в схеме распространения ключей? Самое замечательное свойство сертификатов в том, что их использование можно объединять в цепочку. Действительно, предположим, двум желающим пообщаться абонентам А и D не удалось найти общего знакомого, но выяснилось, что А знает некоего В, а D знает некоего С, которые знакомы между собой. Значит, В может отправить А сертификат о ключе С, а С может отправить А сертификат о ключе D. В итоге А получает уверенность в том, что открытый ключ D, имеющийся у него на руках, истинен. Таким образом, была построена *цепочка доверия*, которая по своей сути представляет тот самый предварительный защищенный канал между А и D (отправителем и получателем), но канал этот был собран (и причем по очень несложной и надежной схеме) из нескольких уже существовавших защищенных каналов. Возможность подобного построения *защищенного канала "по требованию"* из нескольких коротких, уже существовавших, и есть преимущество открытой криптографии.

В настоящее время развитие описанной схемы по всему миру идет очень интенсивно. Намечились следующие основные тенденции. Во-первых, стали появляться субъекты, чьей единственной функцией является хранение и заверение ключей — *центры сертификации* (англ. *Certification Authority* — *CA*). Во-вторых, в процесс создания цепочек доверия стали активно включаться крупные производители программного обеспечения. Действительно, если пользователь ЭВМ приобретает лицензионное ПО в фирменной запечатанной коробке с голограммой и другими физическими степенями

защиты, то задача подделки открытого ключа, находящегося на этом диске, становится на порядок более сложной. А имея несколько надежных открытых ключей крупных производителей ПО, пользователь уже в состоянии строить множество цепочек доверия к миллионам абонентов. И сами производители ПО получают в качестве дивидендов возможность аутентично присылать обновления программ по сети, подписанными теми же ключами, чьи открытые половинки были размещены на первоначальном компакт-диске.

Депонирование ключей

Депонированием ключей называется возможность прочтения зашифрованных сообщений абонента строго ограниченным кругом третьих лиц без его согласия. В подавляющем числе случаев этой группой являются либо уполномоченные государственные служащие (спецслужбы и суды), либо служба безопасности предприятия, на котором он работает. Существуют две основные схемы депонирования ключей. Первая заключается в передаче секретного ключа (при симметричном шифровании) либо закрытого ключа (в асимметричных схемах) ответственному органу на хранение. Согласно второй — шифрование каждой порции данных производится на сеансовом ключе, но сам ключ сеанса, кроме того что шифруется мастер-ключом абонента, дополнительно шифруется открытым ключом ответственного органа и прилагается к тому же сообщению.

Схемы имеют разные свойства и различные области применения. Очевидно, что первая схема гораздо более жесткая — у абонента полностью исчезает возможность передать сообщение без его просмотра ответственными лицами (кроме как сменить ключ или шифрующую систему целиком). По второй схеме пользователь потенциально может и не выполнить депонирующую фазу криптопротокола — следовательно, применять подобную схему можно только там, где вмешательство абонента в процесс шифрования невозможно, либо там, где ему самому невыгодно отправлять сообщения без депонирования ключа. Вторая схема позволяет одновременно депонировать ключ нескольким сторонам — ведь шифровать ключ сеанса можно несколько раз разными открытыми ключами, каждая депонирующая запись добавляет только около 30 байт к сообщению. Кроме того, в схеме с ключом сеанса, если у абонента есть такое право, очень легко можно реализовать выбор им того круга лиц, которые смогут прочесть сообщение в экстренном случае.

Среди наиболее известных реализаций первой схемы — государственная программа США по установке программно-аппаратного комплекса шифрования *CLIPPER*. Устройство представляет собой микросхему, работающую на криптоалгоритме *SkipJack*. В микросхему зашит уникальный идентификатор и секретный ключ, копия которого разделена на две части и хранится в двух различных ведомствах США. Разработка и внедрение этого проекта вызвало широкий резонанс общественности США.

Классическим примером второй схемы депонирования ключей является технология фирмы Microsoft — *Encrypted File System (EFS)* (файловая система с шифрованием). Первой ОС, в которую была встроена EFS, является Microsoft Windows 2000. Любой пользователь персональной ЭВМ, с установленной на ней EFS, может выбрать набор файлов, которые при хранении на диске будут шифроваться секретным ключом пользователя. Шифрование производится с помощью ключа сеанса, т. е. секретный ключ пользователя играет роль мастер-ключа. Однако, кроме шифрования мастер-ключом пользователя, копии ключа сеанса шифруются дополнительно открытым ключом администратора этой персональной ЭВМ и открытыми ключами тех пользователей, которых администратор определил в группу "восстановления ключей" данному пользователю. Таким образом, к каждому файлу в EFS, кроме привычного зашифрованного ключа сеанса, добавляются несколько записей депонирования (или, в терминах Microsoft, — *восстановления*) ключей.

Разделение ключей



Разделение секрета, или *разделение ключей* (англ. *secret sharing* и *key sharing* соответственно), представляет собой процедуру дробления секретного ключа на несколько частей с помощью специальной методики так, чтобы возможность шифрования или дешифрования сообщений возникла только при условии сбора всех частей воедино. Основное предназначение данной технологии — равноправное участие в криптооперациях нескольких пользователей, не доверяющих друг другу, в качестве единого лица. Например, три совладельца фирмы желают читать всю зашифрованную переписку по какому-либо контракту только в случае личного присутствия всех троих. Другой распространенный пример — применение в схемах депонирования ключей: передача секретного ключа на хранение одному доверенному лицу ставит владельца ключа в полную зависимость от него, а вот разделение между двумя и более независимыми организациями позволяет хоть как-то защититься от возможного произвола.

Для читателей, которые в дальнейшем будут использовать англоязычную литературу, авторы хотели бы дать предостережение во избежание терминологической путаницы. В системах аутентификации, обмена ключами и ряде других (например, в описании протоколов Kerberos и RADIUS) можно встретить термин *shared secret*. В этом случае речь идет о каком-либо общем наборе данных, известном обычно двум сторонам и секретном для всех остальных. Например, в случае когда клиент изначально зарегистрирован на сервере, сервер и клиент обладают общим секретом (паролем клиента).

Естественно, простое дробление битового вектора ключа на несколько частей — совершенно неприемлемое решение. Ведь в этом случае при наличии одной или нескольких частей ключа пространство для поиска сокращается

вдвое или втрое: вместо 2^{128} достаточно будет перебрать 2^{43} вариантов. Основное правило процедур разделения ключа: если из всего уполномоченного коллектива отсутствует хотя бы один участник, оставшаяся группа все равно не должна ни на шаг приблизиться к истинному значению ключа, т. е. при ключе в 128 бит этой группе также необходимо перебрать все 2^{128} вариантов, чтобы проверить правильность ключа.

Самая простая схема разделения N -битного ключа KEY между двумя или более (K) субъектами заключается в следующем:

- создается $K - 1$ случайных или криптостойких псевдослучайных векторов S_i длины N ;
- последний компонент S_K вычисляется как "исключающее ИЛИ" всех предыдущих и ключа:

$$S_K = S_1 \text{ XOR } S_2 \text{ XOR } \dots \text{ XOR } S_{K-1} \text{ XOR } KEY,$$

после чего ключ KEY надежно уничтожается.

Ни один из компонентов S_i от 1-го до $(K - 1)$ -го не несет никакой информации о ключе, поскольку они созданы случайно, а последний компонент не значит ничего без остальных $(K - 1)$ значений. В итоге получить значение KEY стороны могут, только собравшись вместе и выполнив операцию "исключающего ИЛИ" над всеми K компонентами ключа.

Дальнейшие исследования проблемы разделения секрета породили схемы, в которых воссоздать ключ могут любые M из K субъектов схемы. Например, сообщения могут прочесть любые 3 из 5 членов совета директоров (кворум). Подобные схемы получили название *пороговых*, или *кворумных* (англ. *threshold schemes*). Две самые простые и популярные схемы были практически независимо предложены *A. Shamir* и *G.R. Blakley*.

Схема Шамира основана на том, что для определения всех коэффициентов произвольного полинома степени $M - 1$ требуется как минимум M точек, в которых известно его значение. Знание $M - 1$ и меньшего количества точек не дает никакой информации о полиноме, в то время как для любого набора из M пар значений (x_i, y_i) существует эффективный алгоритм вычисления всех его коэффициентов. Для пороговой схемы с K субъектами и кворумом из M участников выбирается полином степени $M - 1$, причем в значениях его коэффициентов каким-либо способом закладывается ключ KEY . Затем каждому из K субъектов выдается значение полинома в определенной точке в качестве компоненты ключа. Вычисления можно производить в вещественных числах (закладывая информацию в двоичное представление числа), однако чаще применяются целочисленные поля и для коэффициентов и для значений многочлена. В обоих случаях информационная емкость и сложность вычислений как коэффициентов, так и значений должна соответствовать разрядности разделяемого ключа.

Схема Блэкли основана на том, что систему линейных уравнений с M неизвестными можно решить только при наличии не менее M уравнений в этой системе — в противном случае всегда будет существовать бесконечное множество ее решений. Схема Блэкли имеет очень простую и красивую геометрическую интерпретацию. Если необходимо, чтобы секретный ключ могли собрать любые 2 из K субъектов, то выбирается случайная точка на плоскости, в координатах которой закладывается разделяемый ключ *KEY*. Субъектам раздаются параметры произвольных прямых, проходящих через эту точку (по одной прямой на субъекта). Очевидно, что ни один из них не в состоянии самостоятельно определить, какая же точка его прямой является искомой, и в то же время любые два субъекта однозначно и быстро детектируют точку и узнают секретный ключ. Если кворум состоит из 3 человек, то точка выбирается уже не на плоскости, а в трехмерном пространстве, и субъектам раздаются параметры уже плоскостей — любые три плоскости однозначно определяют искомую точку. В общем случае кворума из M субъектов построения идут в M -мерном пространстве.

В обеих схемах добавление нового субъекта в уже работающую схему делается совершенно безболезненно — в первом случае вычисляется еще одна точка-значение полинома, во втором — генерируется еще одна гиперплоскость. А вот изменение кворума в обоих случаях требует регенерации всех компонент. Интересным и иногда применяемым свойством пороговых схем является возможность назначения целочисленных весов-категорий различным субъектам. Это реализуется выдачей одному лицу или организации сразу нескольких (2, 3, 4) компонент из K -элементного набора. Например, если начальник службы безопасности предприятия получил вес "2" и, соответственно, два ключевых компонента, то при кворуме из 3 субъектов, переписку могут читать либо 3 любых сотрудника службы, либо ее начальник вместе с любым одним сотрудником.

Аутентификация пользователей

Задачи удаленной аутентификации



Задача надежной *аутентификации пользователей*, т. е. подтверждение соответствия удаленного субъекта с тем, за кого он себя выдает, также является прерогативой криптографии. Традиционно о задаче аутентификации говорят в терминах сервера/клиента — сторона, инициирующая запрос и, соответственно, нуждающаяся в аутентификации, именуется *клиентом*, сторона, принимающая запрос и производящая аутентификацию, обычно называется *сервером*. Очевидно, что принятие сервером решения об аутентичности клиента может быть принято только на основе какой-либо уже имеющейся на сервере информации о клиенте, т. е. установление предварительного защищенного канала считается выполненным и выходит за рамки собственно процесса аутентификации.

Также традиционным для задачи аутентификации является оперирование *парольной фразой* — т. е. некоторым небольшим и легко запоминающимся пользователем набором символов. Таким образом, парольная фраза играет роль секретного ключа. Аутентификация с помощью полноценных (128-битных и больше) симметричных ключей и с помощью асимметричных технологий требует использования специальных носителей — запомнить такой ключ рядовому пользователю уже не под силу. А следовательно, задача подобной аутентификации не отличается от проверки подлинности блока данных (в данном случае — записанного на электронном носителе ключа) и также подпадает под действие других криптопротоколов.

В последнее время с развитием Веб-технологий появился способ аутентификации пользователей на основе пользовательских цифровых сертификатов. Естественно, в этом случае пользователь должен произвести предварительную работу по генерации пары ключей и получению сертификата.

Однопакетная парольная аутентификация



В настоящее время существуют два основных метода аутентификации пользователей:

- ☐ с односторонней передачей парольной информации или ее части;
- ☐ по технологии "запрос-ответ".

Односторонняя передача пароля является самой простой и самой незащищенной схемой аутентификации. Пароль передается от клиента к серверу в начале сообщения (или сеанса) либо в открытом виде, либо в виде хэш-суммы. Сервер, который при любой схеме аутентификации должен хранить базу данных паролей всех клиентов, сверяет полученный пакет с сохраненной информацией и принимает решение о личности пользователя.

Передача и хранение хэш-сумм в отличие от открытой парольной фразы имеет следующие преимущества:

- ☐ передаваемая на сервер информация имеет фиксированную, заранее известную длину;
- ☐ необратимость хэш-функции не позволит злоумышленнику при перехвате пакета или взломе серверной стороны получить парольную фразу в открытом виде.

Второе свойство в плане общей безопасности защищает пользователей, которые для доступа на разные серверы используют одну и ту же парольную фразу (в хэшируемый блок можно добавлять уникальное имя сервера и тогда один и тот же пароль будет давать разные хэш-значения). В плане частной безопасности данного сервера хэш-значение может защитить пользователя от дискредитации, если у злоумышленника не окажется возможности

повлиять на процесс собственно хэширования на клиентской стороне и на передачу данных по линиям связи. Ведь в этом случае он не сможет ввести в диалоговом окне подключения к серверу со своей машины имя атакуемого клиента и парольную фразу (он не смог ее восстановить из хэш-суммы). Таким образом, потенциально в программно-аппаратных комплексах, защищенных от модификации их функционирования, можно хранить на сервере хэш-суммы паролей пользователей даже в открытом виде (но так все-таки не поступают — на всякий случай).

Аутентификация "запрос-ответ"



Действительно защищенной от перехвата при прослушивании является *схема "запрос-ответ"* (англ. *challenge handshake*). Она основана на том, что клиент подтверждает серверу знание парольной информации, не раскрывая ее. Традиционная методика схемы "запрос-ответ" выглядит следующим образом:

1. Сервер генерирует случайное или псевдослучайное число (*challenge*) (оно должно отвечать только одному требованию — уникальности, т. е. неповторяемости с течением времени) и высылает его клиенту.
2. Клиент производит какое-либо однонаправленное криптопреобразование (блочное шифрование или хэширование) над парольной фразой и присланным запросом и высылает результат серверу.
3. Сервер производит независимо от клиента такие же преобразования и сверяет получившийся у него результат с присланным ответом клиента.

Технология "запрос-ответ" надежно защищает абонентов от злоумышленников, имеющих возможность прослушивать канал. Информацию о пароле в открытом виде извлечь они не могут из-за необратимости примененного преобразования, а позднее использовать перехваченный ответ клиента для аутентификации не получится из-за того, что сервер каждый раз генерирует новый вектор запроса. Технические аспекты протоколов аутентификации, использующих эту схему, описаны в *главе 22*.

Характер передаваемой информации в схеме "запрос-ответ" позволяет реализовывать очень простые и, как следствие, доступные по цене программно-аппаратные комплексы аутентификации. Не превышающие по размерам зажигалки или брелока, они подключаются к одному из портов ввода-вывода компьютера и позволяют аутентифицировать своего владельца без раскрытия какой-либо информации о записанном секретном ключе. При этом даже при наличии на ЭВМ троянской программы скомпрометированы будут только те данные, которые передавались в данный сеанс с данной ЭВМ, но не сам ключ аутентификации клиента.

Частным случаем схемы "запрос-ответ" является однопакетная аутентификация пользователя на основе штампа времени. По этой методике сервер не

высылает случайного запроса, а вместо него клиент оперирует значением текущего времени, загрубленного до определенной точности, например, одной или нескольких секунд. Подобная схема требует точной синхронизации часов клиента по отношению к серверу. Кроме того, она несет опасность незаконной аутентификации злоумышленником в течение временного интервала "загрубления" точности — для защиты сервер должен принимать только одну попытку регистрации данного пользователя в течение одного временного интервала.

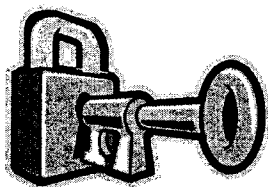
Преимуществом схемы со штампом времени является возможность создавать устройства аутентификации, совершенно не связанные с терминальной ЭВМ клиента. Выполняемые в виде брелоков или часов (и даже интегрируемые в них) они содержат на дисплее сменяющуюся с определенным интервалом кодовую комбинацию из нескольких (4 – 6) цифр/букв. При регистрации на сервере клиент вводит в диалоговом окне текущую запись с индикатора, на основе которой сервер принимает решение о личности удаленного пользователя.

Перспективы криптографии

Закончить рассказ о криптографии можно профессиональной шуткой специалистов по безопасности телекоммуникаций. Есть только три способа защитить данные при пересылке по открытым каналам связи. Какие? — Шифрование, шифрование и еще раз шифрование.

Криптография является очень интересной и бурно развивающейся областью информационных технологий. При умелом и правильном применении она может решить огромное количество проблем, возникающих в сфере информационной безопасности. С каждым днем появляются все новые и новые криптопротоколы с поистине удивительными свойствами и функциями. С экспоненциальным ростом электронной коммерции и виртуального общения криптографические системы станут неотъемлемой, хотя, может быть, и совершенно незаметной, частью нашей жизни.

Криптография — невероятно мощный инструмент, но его критически необходимо использовать правильно. Небольшое нарушение правил ее эксплуатации может "свести на нет" все усилия специалистов по защите вашего предприятия. И самым главным врагом криптографии на сегодняшний день является отнюдь не криптоанализ, а характер и привычки самого человека — конечного пользователя любой криптосистемы. Для того чтобы правильно пользоваться криптографией, нужно в первую очередь знать основы ее функционирования и базовые правила информационной безопасности. В том случае она будет четко, надежно и незаметно служить человеку, как и огромное множество других прикладных наук.



Часть VI

Чем защищать: общетехнические средства

Глава 21. Контроль физического доступа

Глава 22. Протоколы сетевой безопасности

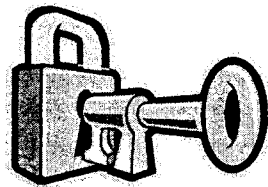
**Глава 23. Автоматизированные средства
безопасности**

**Глава 24. Выбор конкретных
автоматизированных
средств безопасности**

Глава 25. Разработка приложений

**Глава 26. Работа с персоналом
и оборудованием**

Глава 21



Контроль физического доступа

Проблема контроля физического доступа

Иногда в организации безопасности предприятия наблюдаются перегибы в обеспечении защиты информации, по-видимому, в зависимости от того, кто конкретно занимается организацией службы. Они обычно бывают двух видов.

- ❑ Проход на предприятие происходит через несколько пропускных пунктов; помещение, где находятся компьютеры, закрывается на несколько замков, которые открываются только в присутствии директора или начальника охраны; возможно, даже сам компьютер на ночь запирают в сейф. При этом сам компьютер со всеми данными, представляющими критическую важность для организации, регулярно подключается или постоянно подключен к Интернету или просто к работающему модему, любые самые минимальные средства хотя бы программной защиты отсутствуют, а пароль на доступ к компьютеру — "123".
- ❑ Сеть предприятия защищена мощным межсетевым экраном, возможно, специально модифицированным местными умельцами для усиления защиты и пропускающим только протоколы HTTP, SMTP и DNS. На каждой рабочей станции регулярно (иногда даже в ущерб работе пользователей) запускаются антивирусные сканеры. Сеть пронизана агентами системы обнаружения атак, а каждое действие пользователя отражается в различных электронных регистрационных журналах. При этом центральный сервер сети находится в углу зала работы с клиентами, где по окончании рабочего дня уборщицы собираются попить чайку.

Конечно же, это крайние примеры, однако в том или ином виде такие ситуации встречаются на практике. На протяжении всей книги подробно описываются системы, близкие ко второму примеру. Но, естественно, нельзя забывать о вопросах физического доступа к информационным объектам.

Ранее (в *главах 9 и 10*) были приведены примеры атак для случаев, когда злоумышленник имеет доступ к консоли управления системой либо к физическим устройствам ввода/вывода. Поскольку консоль — это обычно место привилеги-

рованного доступа, здесь иногда разрешается выполнение команд, которые нельзя исполнять удаленно, скажем, перезагрузка системы с обходом механизмов защиты либо дополнительное подключение аппаратных устройств.

Кроме того, не следует забывать и о таких тривиальных способах атаки, как похищение носителя с информацией — жесткого диска, магнитных лент в устройствах архивации и др.

Следовательно, основной принцип информационной безопасности — *доступ к информационным объектам должны иметь только уполномоченные сотрудники*, необходимо просто не забывать распространять и на физическое воплощение объектов. В разделе, посвященном физической безопасности, книги "Handbook of Information Security Management" [HISM], Tom Peltier пишет, что для обеспечения безопасности существуют пять ключевых моментов:

- ☐ Двери офиса заперты.
- ☐ Столы и шкафы заперты.
- ☐ Рабочие станции в безопасности.
- ☐ Дискеты в безопасности.
- ☐ Информация предприятия в безопасности.

В последнем пункте имеется в виду другие альтернативные носители информации, такие как распечатки, бумажные документы и т. д.

Механизмы контроля физического доступа

Общие сведения



Механизмы, обеспечивающие контроль физического доступа, схожи с остальными механизмами информационной безопасности, а именно — субъект для доступа к объекту должен быть идентифицирован, аутентифицирован и авторизован. Возможным отличием от электронной работы субъектов с объектами в данном случае является то, что человеческие действия могут быть более распределены во времени и, соответственно, также могут быть проконтролированы людьми в режиме, близком к реальному времени, вплоть до прибытия службы безопасности на место в момент совершения злоумышленных действий.

Современные способы контроля физического доступа разделяются, обычно, на следующие группы:

- ☐ система охраны периметра;
- ☐ система контроля и управления доступом;

- ❑ система видеонаблюдения;
- ❑ система охранной сигнализации (часто совмещается с системой охраны периметра, иногда включает систему пожарной сигнализации);
- ❑ система хранения (сейфы, шкафы и т. п.).

Если позволяет бюджет, выделяемый на обеспечение безопасности объекта, то перечисленные системы возможно интегрировать в единую схему под названием *система управления зданием*, или *интеллектуальное здание*, куда опционально включаются управление лифтами, освещением, водоснабжением, электроэнергией и пр.

Работы по проектированию и внедрению таких систем и, тем более, всего интеллектуального здания, довольно сложны и обычно выполняются сторонней компанией. Вполне логично доверить выполнение таких работ профессионалам, приведем только основные моменты, на которые следует обратить внимание, чтобы проконтролировать качество исполнения работ.

Система охраны периметра



Следует определить, какую территорию следует охранять и какие типы субъектов должны находиться на ней, режим нахождения субъектов.

Для примера рассмотрим здание, обнесенное забором, внутри здания технический этаж, на котором размещена серверная комната. Естественно, что за забором может находиться кто угодно, но в непосредственной близости к забору и у ворот организации посторонние могут находиться только ограниченное время (проходя мимо) — другие случаи должны привлекать внимание службы безопасности. За забор (*охраняемый периметр*) могут пройти только сотрудники организации и лица, которых они сопровождают, причем только в рабочее время, скажем с 8.00 до 18.00. В здание могут пройти только те, кто имеет пропуск, вне зависимости от организационной или должностной принадлежности, причем входить можно — работникам с 8.00 до 18.00, выходить до 22.00, а субъекты могут входить только до 16.00, а выходить строго до 18.00. На технический этаж могут подниматься исключительно работники технической службы и только в рабочее время, а в серверную комнату может входить лишь системный оператор (администратор) и только после информирования службы безопасности.

Теперь можно определять, какие механизмы необходимо использовать для решения сформулированных задач. Желательно, чтобы забор был полностью непроходимым (в зависимости от режимности организации, с исключением возможности подкопа или перелета), а также был снабжен камерами видеонаблюдения для контроля приближения к забору. У внутренней стороны забора может быть также установление датчиков движения либо "наступания" для того, чтобы, с одной стороны, обнаружить субъектов, миновавших забор снаружи, а, с другой стороны, предотвратить приближение

работников организации к забору. На стене забора (здания, подвала) допустима установка датчиков вибрации на предмет попыток пролома стен. На воротах забора достаточно визуального контроля сотрудника безопасности для проверки документов, а также наличие металлоискателя и/или просвечивающего устройства для проверки вносимых предметов. Но вопросы открытого прохода субъектов на территорию — в разд. "Система управления доступом" данной главы.

Система управления доступом



Управление доступом можно условно разделить на две части — *управление первичным проходом* на охраняемую территорию и *управление перемещением* по охраняемой территории. Если внутри одной охраняемой территории находится другая, еще более строго охраняемая, то количество разделений, соответственно, удваивается.

Задача первичного контроля — отсечь тех, к кому не могут быть применены авторизационные правила, т. е. тех, у кого на территории нет никаких прав доступа; обеспечить, чтобы на территорию не были пронесены запрещенные предметы, а также ряд других функций, в том числе, возможно, те, которые не могут быть заранее регламентированы или автоматизированы (например, принятие решения при подозрении, что предъявленный пропуск принадлежит другому лицу). Наиболее известные механизмы в данном случае следующие:

- ❑ турникеты и металлические ворота, обеспечивающие разделение человеческого потока на отдельных персон;
- ❑ шлюзовые кабины, обеспечивающие проход строго по 1 человеку, без возможности сопровождающего принудить авторизованного сотрудника к совместному проходу через контрольные ворота;
- ❑ металлоискатели, желательно с возможностью настройки на габариты проносимых металлических предметов, чтобы отделить, скажем, связку ключей от ножа;
- ❑ просвечивающие устройства — необходимо только определить, будут ли это устройства безопасные для свето- и магниточувствительных материалов, либо наоборот, жестко выводящие такие материалы из строя (в этом случае необходимо предупреждение о предъявлении таких материалов и сдаче их на хранение или анализ);
- ❑ переговорные устройства, если управление входом/выходом осуществляется удаленно (сюда входят домофоны, видеодомофоны, интерфоны и т. п.).

Кроме перечисленного, на входе могут стоять устройства считывания параметров доступа, такие же как на всех контролируемых участках остальной территории.

Задача контроля перемещений — идентифицировать/аутентифицировать субъекта, запрашивающего разрешение на проход, и, применив к нему авторизационные правила, пропустить его или запретить проход.

Идентификация/аутентификация субъекта производится на основе стандартных принципов — о них уже было рассказано в *главе 10*. Напомним их.

- ❑ Что-то, что субъект знает. Обычно это некий ПИН (персональный идентификационный номер) или код, который необходимо набрать на панели замка.
- ❑ Что-то, что субъект имеет. Это может быть и обычный дверной ключ или токен — карточка с магнитной полосой, энергонезависимый идентификатор TouchMemory, смарт-карта или proximity-карта.
- ❑ Что-то, что физически или психологически неотделимо от субъекта. Это обычно биометрические параметры тела субъекта. Иногда из данного раздела выделяют подраздел "Что-то, что субъект делает" — скорость и тип печати на клавиатуре, механика личной подписи, распознавание голоса. Иногда это разделение важно, так как при определенных условиях, например, под принуждением, субъект не может изменить, скажем, отпечатки своих пальцев, но может изменить голос и, в последнем случае, злоумышленник не сможет проникнуть на территорию, даже принуждая субъекта.

Учитывая специфику человеческой личности, можно сказать, что система контроля доступа, ориентированная только на набор кода, недостаточно надежна, так как даже ПИНЫ имеют тенденцию становиться широко известными через некоторое время, не говоря уже об общем коде доступа.

При выборе токена следует дополнительно учитывать следующие параметры:

- ❑ износ (магнитная полоса стирается с карточки при многократном считывании, энергонезависимые элементы имеют ограниченный срок хранения/использования);
- ❑ скорость прохода (прикладывание элемента к считывателю или протаскивание/вставка карточки в считыватель требуют определенного времени);
- ❑ стоимость;
- ❑ возможность нанесения фотографии владельца на токен;
- ❑ прочность на возможный излом/повреждение и т. п.

Среди биометрических средств аутентификации наиболее часто встречающиеся: отпечатки пальцев, сетчатка глаза, рисунок ладони, геометрия руки, форма лица, параметры голоса. Однако перед принятием решения о приобретении того или иного средства необходимо ознакомиться с последней имеющейся статистикой по ложному срабатыванию и ложному отказу для данного вида биометрии. Кроме того, при использовании таких устройств,

следует учитывать культурные традиции и персональные привычки — для некоторых людей, например, невозможно прикасаться к поверхностям, которые касались несколько десятков или сотен человек перед этим.

При проектировании системы контроля перемещения следует учитывать необходимость применения к субъекту единой политики доступа, вне зависимости от того, в какой части территории запрашивается доступ. Это означает, что, во-первых, политика должна быть создана и сохранена в одном месте для обеспечения единообразия применения, а, во-вторых, единая политика должна быть распределена по нескольким устройствам для обеспечения продолжения работы при выходе из строя центрального сервера политики. Дополнительно следует предусмотреть формат регистрации перемещений субъектов в электронном журнале, возможности анализа журнала, а также его интеграцию с другими журналами — например, с журналом доступа в компьютерную сеть. Авторизационная политика должна не только определять, разрешен или запрещен проход субъекта, но и дату, день недели и время суток, а также направление прохода (например, запрет выхода, если не фиксировался вход). Кроме того, в системе необходимо предусмотреть оповещение на экран дежурного сотрудника (или другие варианты) в случае, если проходная дверь остается открытой в течение промежутка времени большего, чем необходимо для прохода.

Возможно, еще следует учесть изменения политики для массовых явлений — игнорирование запретов и открытие дверей в случае пожара или игнорирование разрешений и закрытие дверей в случае обнаружения злоумышленника. Последняя деталь — с одной стороны, следует обязательно иметь мастер-ключи (ключи безусловного прохода) на случай сбоев системы, но, с другой стороны, строго учитывать их хранение и использование. Для помещений, в которые ограничен только вход, а выход разрешен всем, кто зашел, следует иметь на выходе не считыватель, а всего лишь кнопку открывания двери.

Система видеонаблюдения



Первым этапом установки подобной системы является определение ее целей: контроль подходов к периметру, проходов, поведения работников в помещениях. В зависимости от задач *системе видеонаблюдения* потребуются разные типы и камер и другого оборудования. Затем необходимо определить:

- ☐ действие на камеру слежения со стороны открытой среды (дождя, снега, ветра, пыли);
- ☐ освещенность, площадь и открытость территории (возможно, придется устанавливать источники света, демонтировать потенциальные укрытия и т. д.);
- ☐ использовать ли камеру скрытую или открытую;

- ☐ будет ли заметно ее вращение (если камера вращающаяся и не скрыта);
- ☐ нужно ли видеть только общие контуры объектов или также и детали (увеличение изображения);
- ☐ достаточно ли черно-белого изображения или требуется цветное;
- ☐ не станет ли сама камера объектом похищения, если она установлена в доступном месте и вне быстрой досягаемости сотрудников службы безопасности;
- ☐ как будет просматриваться изображение с камер — по очереди, по несколько, все сразу;
- ☐ будет ли производиться запись изображения и его хранение.

До начала установки оборудования надо протестировать отдельные экземпляры камер, мониторов, видеомagneтофонов. Возможно, потребуются различные типы оборудования — скажем, черно-белые/статичные камеры для наблюдения за периметром и цветные/вращающиеся для наблюдения за работниками в офисе. В этом случае необходимо протестировать все типы камер.

Если требуется вести запись изображения, то желательно ответить дополнительно на ряд вопросов:

- ☐ будет ли записываться изображение со всех камер или только с некоторых;
- ☐ будет ли запись вестись постоянно или только в случае срабатывания сигнала тревоги в пределах досягаемости камеры;
- ☐ будет ли вестись запись непрерывно или возможна дискретная запись (кадр в секунду/две/три);
- ☐ необходимо ли накладывать на изображение дату/время;
- ☐ надо ли записывать звук;
- ☐ будет ли запись цифровой или аналоговой.

Существует дополнительный ряд вопросов, на которые представитель службы безопасности не сможет ответить самостоятельно, например, какой тип кронштейнов для крепления камер использовать. В зависимости от степени доверия компании подрядчику такие вопросы можно оставить на ее усмотрение либо пригласить сторонних экспертов.

Система охранной (пожарной) сигнализации



Для выбора типа (а вернее даже — типов) охранной сигнализации следует определить, какие события требуют подачи сигнала тревоги. Возможно это сотрясение или пролом — для заборов и стен, касание или приближение — для оград, нарушение целостности — для окон, наступание/помещение и массы — для открытого пространства, изменение объема — для закрытого пространства, открытие/закрытие двери, движение — для различных участков охраняемой территории. Для пожарных датчиков признаками срабатывания будут задымле-

ние помещения и/или повышение температуры. Дорогим, но достаточно надежным решением будет также детектор пламени, который определяет источник огня по инфракрасной энергии пламени или по его пульсации.

Перед приобретением конкретного вида датчиков необходимо ознакомиться со статистикой качества работы устройства — особенно с процентом ложного срабатывания от помех и несрабатывания при реальном нарушении. Возможно, эти характеристики придется протестировать самим специалистам службы безопасности либо поискать у независимых экспертов.

При проектировании системы охранной сигнализации следует сразу привязывать срабатывание сигнала тревоги к определенному сценарию поведения всей системы безопасности. Например, срабатывание датчика движения в ночное время одновременно вызывает поворот камеры в сторону датчика, переход записи с камеры с дискретного на непрерывный режимы, блокирование дверей прохода и сигнал на пульт охранной смены. Пожарные датчики следует привязывать к системе пожаротушения и громкой связи оповещения персонала.

Система хранения



Система хранения (хранилище материальных ценностей) сильно зависит от типа организации. Очевидно, что для банка, производящего операции с наличными, требуются одни типы сейфов, а для компании по разработке программного обеспечения совсем другие. В любом случае необходимо определиться с физическим объемом хранимых объектов, устойчивостью хранилища к физическому или электронному взлому, пожароустойчивость, переносимость, скрытость монтажа.

Интеграция систем контроля



При рассмотрении всех компонентов системы физической безопасности сразу возникает вопрос, а не интегрировать ли все это оборудование в единый комплекс? Положительный ответ на него не так уж и безупречен, как может показаться с первого взгляда. Территориальное разнесение отдельных датчиков и целых подсистем, которое всегда имеет место в реальных ситуациях, поднимает при интегрировании служб такую важную проблему, как связь. Возможно, для злоумышленника, которым является, например, уже попавший на территорию сотрудник, будет гораздо проще вывести из строя систему связи датчиков с центром сигнализации, чем пытаться как-либо обмануть сам датчик.

Здесь следует сделать небольшое отступление. Выведение из строя линии связи датчика с центром или самого датчика, как цель злоумышленника зависит от того, как сконфигурирована система. Наиболее простой пример — датчик открытия прохода (двери). Как должен вести себя замок в

случае, если датчик (или линия связи) отказал? Очевидный, казалось бы, ответ — блокирование прохода до починки датчика может представлять смертельную угрозу в случае землетрясения или пожара — ведь в этом случае персонал может оказаться заблокированным в здании. Таким образом, решение должно учитывать специфику работы организации, здания и другие параметры. Кроме того, возможно препятствовать вмешательству злоумышленника с помощью дополнительных мер. В нашем случае с датчиком прохода — это может быть антивандальный кожух, видеокамера с контролем использования датчика и т. п.

Естественно, система с единым центром обработки и примитивнейшими устройствами на местах в большинстве случаев обойдется предприятию гораздо дешевле других, но стоит задуматься, насколько оправдана эта экономия. Идеальным и, наверное, самым дорогим вариантом является установка на местах независимых, т. е. полнофункциональных, систем мониторинга и сигнализации, но соединенных информационными каналами с единым центром. В этом случае при потере связи с единым пультом подсистемы начинают работать по последнему полученному из центра алгоритму либо по аварийному плану. При этом крайне желательно, чтобы каждая подобная система имела собственные альтернативные каналы оповещения заинтересованных лиц во вне предприятия (например, органов охраны правопорядка или резервного внешнего центра сигнализации учреждения). В крайнем случае возможно использование громкоговорящей и световой сигнализации.

Кроме того, при объединении датчиков и служб физического контроля следует задуматься, стоит ли совмещать сеть управления системой безопасности и общую компьютерную сеть предприятия или необходимо разнести эти два информационных пространства. Дело в том, что в большинстве организаций администратор компьютерной сети является если не царем и богом, то во всяком случае имеет расширенные права по управлению информационными системами и объектами. Действительно ли руководитель организации хочет, чтобы системный администратор имел также и возможность управления (или потенциального внесения погрешностей в работу) системой физической безопасности?

Прочие аспекты

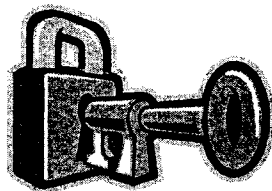


В заключение хочется сказать, что после того, как все типы оборудования в зависимости от требуемых задач будут выбраны, поставщик смонтирует и установит систему и подсистемы, на плечи службы безопасности ляжет тяжелый груз по приему системы в эксплуатацию. Необходимо будет проверить работу каждой единицы оборудования как по отдельности, так и при стрессовой нагрузке (например, работу системы управления доступом при массовом выходе сотрудников из здания), срабатывание запланированных сценариев, как по отдельности, так и вместе (что будет, если поступит од-

новременно сигнал злоумышленного вторжения, блокирующий открытие дверей и сигнал пожарной тревоги, открывающий все двери). Возможно, при таком тестировании будет обнаружено, что какие-либо мелочи типа доводчиков принудительного закрытия дверей были упущены, хотя это уже минус компании-порядчику по установке системы.

И хотелось бы отметить еще пару важных моментов. Во-первых, еще раз упомянуть о том, что следует предусмотреть возможность интеграции системы безопасности с другими системами, пусть даже офф-лайн. Например, будет полезно сравнивать журнал доступа в автоматизированную систему пользователя с его рабочего места с фактом его реального физического присутствия на месте по журналу системы управления доступом. Во-вторых, персоналу службы безопасности для эффективного использования новой системы безопасности следует пройти соответствующее обучение и, возможно, сертификацию. Кроме того, для дежурных смен системы, может быть, придется предусмотреть отдельные требования по физическому и психологическому здоровью, поскольку смотреть несколько часов подряд в мониторы видеокамер — задача не из легких.

Глава 22



Протоколы сетевой безопасности

Для того чтобы успешно внедрять и поддерживать различные механизмы информационной безопасности, оценивать системы перед выбором при приобретении, зная их сильные и слабые места, беседовать на одном языке с коллегами-профессионалами и понимать, о чем пишут в специальной литературе, необходимо иметь представление о том, какие способы защиты уже придуманы и воплощены, возможно, являются отраслевым стандартом в вопросах защиты информации.

Одним из видов таких стандартов являются различные протоколы безопасности (набор общепризнанных и поддерживаемых правил и форматов взаимодействия), о которых будет рассказано в данной главе.

Если возникла необходимость досконально разобраться в конкретном протоколе, в первую очередь следует обратиться к соответствующему документу из огромного сетевого информатория *Request For Comments — RFC — Запрос на обсуждение* (если таковой существует для данного протокола) на сайте <http://ietf.org/>. Однако иногда проще начинать подробное изучение после того, как составлено общее представление о предмете, именно для этого и предназначена данная глава.

Некоторым из специалистов также нет необходимости очень подробно знакомиться с реализацией конкретных механизмов, им будет достаточно данного материала.

Задачи протоколов сетевой безопасности



Тем, кому не требуется знать, как работают протоколы безопасности, тем не менее следует хотя бы различать их по функциональности области приложения и решаемым задачам. Мы уже знаем, что перед тем, как применить какие-либо механизмы безопасности к объекту или субъекту, необходимо его однозначно идентифицировать (то есть узнать его системно-информационное

наименование и убедиться, что оно реально соответствует объекту/субъекту), или, иначе говоря, аутентифицировать.

Поэтому первая задача протоколов безопасности — *аутентификация* удаленного объекта или субъекта (пользователя, системы или процесса). Практически все протоколы несут в себе аутентификацию, как одну из функций, но есть ряд протоколов предназначенных специально для аутентификации (PAP, CHAP и их подвиды; RADIUS, TACACS, Kerberos, S/Key).

Следующая задача протоколов безопасности — *обеспечить защиту информации* при прохождении по каналам связи, т. е. согласовать ключи шифрования, зашифровать данные в точке отправления, расшифровать в точке получения. Говоря о подобных протоколах, следует понимать область их приложения, относительно модели OSI или TCP/IP. Это важно учитывать при выборе конкретной модели безопасности, так как необходимо знать, какая часть информации остается видимой после криптографической обработки (адреса отправителя и получателя, другая служебная информация).

Например, протоколы прикладного уровня (механизмы шифрования в рамках конкретного бизнес-приложения или стандартные протоколы прикладного уровня, такие как SHTTP) оставляют в открытом, нешифрованном виде всю информацию, которая необходима для работы нижних уровней (номера портов, IP-адреса, последовательные номера пакетов и т. п.). Следует помнить, что эта информация остается видимой и, если покидает в таком виде локальную сеть, то дает возможному злоумышленнику некоторые дополнительные сведения о топологии сети (по адресам), о работающих приложениях (по портам) и т. д. Протоколы транспортного уровня (SSL, Secure Shell, SOCKS) скрывают картину работы отдельных приложений, но оставляют информацию для сетевого уровня. Многие протоколы сетевого или подсетевого уровня (IPSec, L2F, PPTP, L2TP) могут скрывать:

- либо только данные, пришедшие от протокола верхнего уровня (транспортного), оставляя видимыми адреса отправителя и получателя и некоторую другую информацию;
- либо полностью инкапсулировать все данные сетевого уровня, выставляя новые заголовки и окончания пакетов; такой режим работы часто еще называют *туннелированием* и именно он участвует в построении *виртуальных частных сетей* (англ. *virtual private network* — VPN).

Некоторые протоколы известны не только своей функциональной нагрузкой, но и задачами, которые они решают. Скажем, SSL — это де-факто сложившийся стандарт защиты интернет-соединений, в том числе при использовании систем оплаты по пластиковым картам. Сервер протокола SOCKS, обеспечивающий защиту данных между отправителем и получателем

лем, выступает как *шлюз-посредник приложений* (англ. *application-level proxy*). Kerberos популярен как система, позволяющая пользователю аутентифицировать себя (например, вводить пароль) лишь однажды, при входе в систему, а далее получать прозрачный доступ (в рамках своих прав) ко всем ресурсам сети — механизм получил специальное название *SSO* — *single sign-on* — "один вход".

Существуют дополнительные протоколы, ориентированные на выполнение специальных задач, такие как X.509 — протокол цифровых сертификатов, указывающий, каким образом субъекты, использующие в открытой сети механизмы асимметричной криптографии, должны распространять свои открытые ключи через центры сертификации (CA — *certification authority*). LDAP (*Lightweight Directory Access Protocol*) — протокол, регулирующий доступ к данным об объектах и субъектах данной зоны управления (домена, службы каталога и т. п.).

При анализе безопасности конкретной системы необходимо выяснить, какие механизмы защиты она использует (идентификацию, аутентификацию, защищенный обмен данными и/или управляющими сообщениями, регулирует доступ и т. д.), выяснить, на основе каких протоколов работают соответствующие механизмы, и далее делать вывод об их сильных или слабых сторонах. С другой стороны, принимая участие в разработке новой системы, следует знать, какие уже разработанные механизмы можно использовать целиком, а какие замечательные красивые идеи (такие как одноразовые пароли S/Key или распределенная аутентификация без пересылки пароля Kerberos) можно встроить в конкретную систему.

Техническое описание протоколов



Профессионал в области информационной безопасности скорее всего уже сталкивался с описанными протоколами. Порой достаточно посмотреть, как раскрывается данный акроним (аббревиатура), чтобы понять, о чем идет речь. Иногда требуется и подробное изучение. И конечно же, возникает желание найти ресурс или издание, где подобный материал в изложении средней детализации объединен и, возможно, дополнен субъективными мыслями. Один из лучших англоязычных ресурсов в данном направлении — сайт корпорации Cisco Systems, Inc. — <http://www.cisco.com/>. Среди русскоязычных печатных изданий достаточно большой спектр протоколов рассмотрен в переводной книге У. Блека [Блек1], хотя, конечно, уровень детализации подобных материалов не может сравниться с электронными спецификациями. В данной главе мы постараемся внести посильную лепту в этот поток информации, обзорно рассмотрев наиболее распространенные протоколы безопасности и выделив те их свойства и параметры, с которыми приходится сталкиваться с наибольшей вероятностью на практике.

Point-to-Point Protocol

Для обеспечения аутентификации в сетях, основанных на телефонных линиях (коммутируемые или выделенные каналы, dial-in, в западном варианте ISDN), используется ряд специальных протоколов. Чаще всего они работают совместно с протоколом *PPP (Point-to-Point Protocol)* [RFC-1661], его рассмотрение выходит за рамки данной книги, так как он не относится собственно к протоколам безопасности. Для обеспечения единой базы укажем структуру (формат) кадра PPP — рис. 22.1.

Байт флага	Байт адреса	Контрольный байт	Два байта протокола	Данные	Контрольная последовательность кадров	Байт флага

Рис. 22.1. Структура кадра PPP

При описании схем аутентификации в PPP используется следующая терминология: субъект, для которого требуется аутентификация, называется *узлом* (англ. *peer*), а объект, производящий аутентификацию, называется *аутентификатором* (англ. *authenticator*).

Password Authentication Protocol

Данный протокол — наиболее простой из протоколов подтверждения удаленным субъектом своего идентификатора для объекта, предоставляющего ресурсы для использования. Аутентификация происходит за две итерации (англ. *two-way handshake*). При использовании *PAP (Password Authentication Protocol)* [RFC-1334] в поле *Протокола* (два байта протокола) кадра PPP указывается соответствующее PAP-значение 0x023, поле данных преобразуется в четыре дополнительных поля (рис. 22.2).

Код	Идентификатор	Длина	Данные

Рис. 22.2. Структура поля *Данные* кадра PPP

При этом поле *Код* указывает на следующие возможные типы PAP-пакета.

Код = 1: Аутентификационный запрос.

Код = 2: Подтверждение аутентификации.

Код = 3: Отказ в аутентификации.

Поле *Идентификатор* обеспечивает соответствие пары запрос/ответ (должен меняться при каждом новом аутентификационном запросе).

Поле *Длина* указывает совокупную длину всех четырех полей.

Поле *Данные* содержит данные пакета — для аутентификационного запроса будет иметь вид, приведенный на рис. 22.3.

Длина идентификатора	Идентификатор	Длина пароля	Пароль

Рис. 22.3. Структура поля *Данные RAR пакета-запроса*

Пакет аутентификационного запроса будет посылаться субъектом, желающим получить доступ неоднократно до наступления одного из следующих событий:

- ☐ получение подтверждения или отказа;
- ☐ истечение счетчика попыток.

При получении запроса объектом производится распознавание полученных результатов (сравнение с имеющимися у объекта значениями). По результатам распознавания субъекту высылается пакет с полем *Данные* следующего формата — рис. 22.4.

Длина сообщения	Сообщение

Рис. 22.4. Структура поля *Данные RAR пакета-ответа*

При этом в полях (см. рис. 22.2) *Код* указывается 2 или 3 (в зависимости от того, подтверждена аутентификация или отвергнута), в поле *Идентификатор* — идентификатор соответствующего запроса. В полях ответа (см. рис. 22.4) указывается: *Длина сообщения* — размер следующего поля, *Сообщение* — возлагается на конкретную реализацию, оно обязано не влиять на работу протокола и рекомендовано формировать его удобным для прочтения.

Дополнительно указано, что, поскольку пакет с подтверждением аутентификации может быть утерян, реализация должна предусматривать возможность обработки повторного запроса на аутентификацию.

Таким образом, схема работы протокола следующая (рис. 22.5):

1. Устанавливается PPP соединение.
2. Субъект посылает аутентификационный запрос с указанием своего идентификатора и пароля.
3. Объект проверяет полученные данные и подтверждает аутентификацию или отказывает в ней.

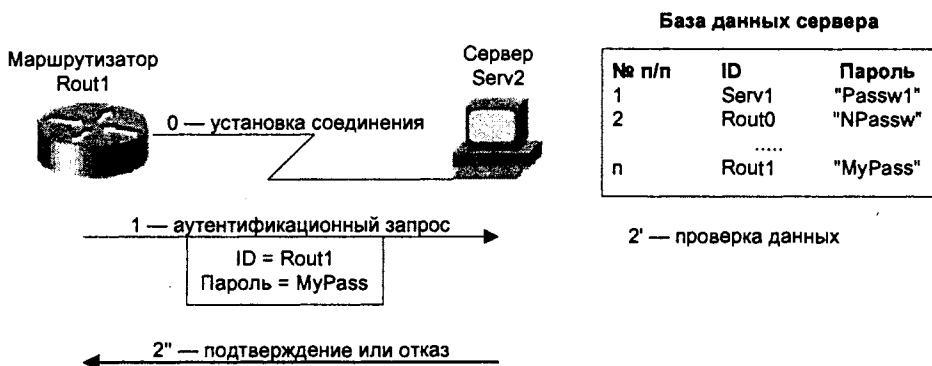


Рис. 22.5. Пример PAP-аутентификации маршрутизатора сервером

Следует обратить особое внимание, что весь обмен данными (в том числе и пересылка пароля) происходят в открытом виде, без применения криптографических средств. При этом частоту и время отправки пакетов контролирует сам субъект.

Для данного протокола существует другая известная реализация — *SPAP* (*Shiva Password Authentication Protocol*).

Challenge Handshake Authentication Protocol

Протокол *Challenge Handshake Authentication Protocol (CHAP)* [RFC-1994] используется для первоначальной аутентификации субъекта после установления соединения, но может, в зависимости от конкретной реализации, быть использован для периодического подтверждения аутентифицированности субъекта во время работы в рамках установленного соединения. Аутентификация происходит за три итерации (англ. *three-way handshake*). В поле *Протокол* PPP кадра указывается значение $0 \times c223$, поле *Данные* преобразуется в четыре поля, аналогично PAP пакету (см. рис. 22.2) со схожими типами, единственное отличие в том, что поле *Код* имеет теперь четыре значения:

Код = 1: Запрос на предоставление данных для аутентификации.

Код = 2: Ответ на запрос с аутентификационными данными.

Код = 3: Подтверждение аутентификации.

Код = 4: Отказ в аутентификации.

Формат поля *Данные* зависит от поля *Код*.

Реализация протокола CHAP требует, чтобы обе стороны имели в распоряжении *заранее согласованный пароль* (англ. *pre-shared secret*), который не высылается по сети, но чаще всего присутствует у объекта в открытом виде (*plaintext*), при этом длина пароля должна быть не меньше 1 байта, и в принципе он должен соответствовать требованиям хэш-алгоритма, выбранного для реализации.

Собственно процедура аутентификации инициируется, в отличие от PAP, не субъектом, а объектом и происходит следующим образом (рис. 22.6):

1. Устанавливается PPP-соединение.
2. Объект посылает субъекту запрос на предоставление данных для произведения аутентификации.
3. Субъект отвечает необходимыми данными, часть из которых взята из содержимого запроса.
4. Объект анализирует полученные данные и отвечает подтверждением или отказом.

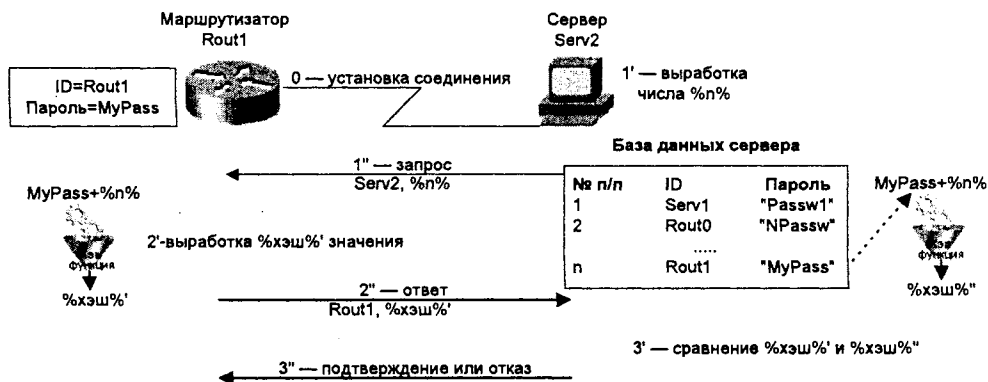


Рис. 22.6. Пример CHAP-аутентификации маршрутизатора сервером

Рассмотрим подробнее структуру пакета для запроса и ответа (поле *Данные* кадра PPP) — рис. 22.7.

Код	Идентификатор	Длина	Длина числа	Число	Имя

Рис. 22.7. Структура CHAP-пакета запроса или ответа

В поле *Код* указываются значения 1 для запроса и 2 для ответа.

Поле *Идентификатор* используется аналогично полю *Идентификатор* протокола PAP для обеспечения однозначного соответствия между запросом и ответом. Оно должно уменьшаться при новых запросах.

Поле *Длина* показывает суммарную длину всех полей пакета.

Поле *Длина числа* определяет размер следующего поля.

В поле *Число* указывается специальное значение, которое должно удовлетворять двум требованиям: быть уникальным и быть непредсказуемым. От качества выбора этого числа зависит, сможет ли злоумышленник использо-

вать перехваченные им запросы-ответы при аутентификации уполномоченного субъекта. Для пакета *Запрос* это значение должно изменяться при каждом новом пакете.

В поле *Имя* указывается наименование системы, пославшей пакет. Так как субъект может аутентифицироваться на нескольких объектах, и объект может аутентифицировать несколько субъектов, это поле можно использовать для поиска пароля в соответствующей базе данных.

После получения *Запроса* субъект производит следующие действия:

1. Выбирает соответствующий объекту пароль.
2. Использует пароль и полученное *Число* как параметры для хэш-функции (обычно MD5) и получает соответствующее хэш-значение.
3. Формирует пакет *Ответа*, где в поле *Число* указывает полученное хэш-значение, а в поле *Имя* — свое наименование.
4. Высылает пакет объекту.

Объект, получив пакет, производит свое вычисление хэш-значения с использованием соответствующего пароля для данного субъекта и *Числа*. В случае совпадения рассчитанного и присланного хэш-значений, происходит подтверждение аутентификации, иначе — отказ. Формат пакета для этого аналогичен формату пакета PAP с *Кодом* = 3 для успешной авторизации и с *Кодом* = 4 для отказа.

Существует реализация данного протокола компанией Microsoft (MS-CHAP — Microsoft Challenge Handshake Authentication Protocol), в которой хранение пароля субъектом и объектом обеспечивается в зашифрованном виде.

Extensible Authentication Protocol

Главная особенность протокола Extensible Authentication Protocol (EAP) [RFC-2284] в том, что он может использовать различные (множественные — англ. multiple) аутентификационные механизмы, выбор механизма аутентификации переносится на усмотрение объекта, который может запросить у субъекта дополнительные параметры для определения такого механизма.

В поле *Протокол* PPP кадра указывается значение 0xc227, поле *Данные* преобразуется в четыре поля, аналогично PAP и CHAP пакету (см. рис. 22.2) со схожими типами, единственное отличие в том, что поле *Код* имеет четыре значения, как и для CHAP.

Код = 1: Запрос на предоставление данных.

Код = 2: Ответ на запрос.

Код = 3: Подтверждение аутентификации.

Код = 4: Отказ в аутентификации.

Поскольку процедура аутентификации может потребовать участия пользователя, необходимо предусматривать соответствующие тайм-ауты при рассылках объектом запросов на аутентификацию.

Процедура аутентификации инициируется объектом (англ. *authenticator*) и происходит следующим образом:

1. Устанавливается PPP соединение.
2. Объект посылает субъекту запрос на предоставление данных для производства аутентификации.
3. Субъект отвечает необходимыми данными, часть из которых взята из содержимого запроса.
4. Объект анализирует полученные данные и отвечает подтверждением или отказом.

Структура EAP пакета для запроса и ответа приведена на рис. 22.8.

Код	Идентификатор	Длина	Тип	Данные типа

Рис. 22.8. Структура EAP-пакета запроса или ответа

В поле *Код* указываются значения 1 для запроса и 2 для ответа

Поле *Идентификатор* используется аналогично соответствующему полю протокола PAP и CHAP для обеспечения соответствия между запросом и ответом. Оно должно уменьшаться при новых запросах. Однако необходимо учесть, что поле *Идентификатор* может содержать повторяющееся значение (и должно совпадать для повторяющихся (англ. *retransmitted*) запросов) для рассмотренных случаев запаздывания с получением данных от пользователя.

Поле *Длина* указывает суммарную длину всех полей пакета.

Поле *Длина числа* указывает размер следующего поля.

Поле *Тип* обозначает тип запроса или ответа (собственно тип аутентификации) и должно указываться в запросе и совпадать с ним в ответе (если субъект поддерживает предложенный тип аутентификации), либо в ответе указывается NAK, если данный тип аутентификации не поддерживается. (Субъект также может указать приемлемый для себя тип аутентификации.)

Поле *Данные Типа* содержит данные, соответствующие указанному типу.

При успешной аутентификации объект высылает субъекту пакет следующей структуры (рис. 22.9).

Поле *Код* равно 3 при подтверждении и 4 — при отказе. Однако при этом объект может запросить дополнительную аутентификацию, учитывая, что отказ мог произойти по причине человеческой ошибки.

Код	Идентификатор	Длина

Рис. 22.9. Структура EAP-пакета подтверждения или отказа аутентификации

Поле *Идентификатор* соответствует идентификатору запроса, на который посылается ответ.

Поле *Длина* равно 4.

Теперь рассмотрим, какие типы аутентификации (значения поля *Тип* в запросе/ответе) поддерживаются протоколом.

Тип = 1: Обозначение субъекта (англ. *identity*).

Тип = 2: Сообщение.

Тип = 3: NAK (только для ответов).

Тип = 4: Ответ MD5.

Тип = 5: Одноразовый пароль.

Тип = 6: Вид карты токена (Generic token card).

Необходимо учесть, что [RFC-1938], описывающий систему одноразовых паролей, заменен на [RFC-2289].

Тип 1 используется для определения субъекта, возможно, в поле *Данные Типа* использования сообщения для приглашения пользователя ко вводу своего идентификатора (не путать с идентификатором пакета).

Тип 2 используется для передачи субъекту сообщения (например, сообщение о скором истечении пароля).

Тип 3 используется только при ответе и указывает, что предложенный метод аутентификации неприемлем для субъекта, в поле *Данные Типа* указывается желаемый тип аутентификации.

Тип 4 используется для запроса/ответа (число или хэш-значение) согласно протоколу CHAP.

Для *Типа* 5 поле *Данные Типа* содержит запрос для одноразового пароля или 6 слов из соответствующего словаря одноразовых паролей.

Тип 6 используется при применении различных типов токенных карт, требующих пользовательского ввода. В запросе указывается текстовая (ASCII) информация, а в ответе — аутентификационные данные (например, то, что пользователь считал с токеной карты).

Система одноразовых паролей S/Key

Система одноразовых паролей предназначена для защиты от случаев, когда злоумышленник "прослушивает" сеть, пытаясь перехватить пароль (или соответствующее ему выражение, например, хэш-значение) для дальнейшего его использования (англ. *replay attack*). В системе S/Key [RFC-1760] парольная фраза пересылается по сети только однажды и после этого больше не используется, что делает описанную атаку бессмысленной. При этом сама парольная фраза никогда не пересылается, т. е. пользователь может использовать свой привычный ему осмысленный пароль многократно. Пароль, собственно вводимый пользователем в интерфейсе программы, назовем *секретом*, чтобы не путать его далее с самим одноразовым паролем.

Данная система основана на клиент/серверном подходе. Клиент генерирует одноразовый пароль по схеме, сервер верифицирует его.

Использование одноразового пароля происходит в три фазы:

- Подготовительная. Сбор данных для ввода.
- Генерационная. Многократное применение хэш-функции к данным.
- Вывод. 64-битовый одноразовый пароль выводится в виде, удобном для восприятия пользователем.

Первоначально клиент и сервер должны быть сконфигурированы для использования единого секрета, т. е. он должен присутствовать и у клиента (например, в памяти пользователя), и у сервера. Далее, для создания уникальности одноразового пароля клиент и сервер должны определить случайное число (*seed*) и число итераций применения хэш-функции. Эти значения сервер в ответ на запрос об аутентификации (пакет инициализации) может выслать клиенту в открытом виде (*plaintext*), они не являются секретными. При этом необходимо учесть, что одноразовые пароли используются сериями, в каждой серии случайное число — единое для каждого одноразового пароля, а число итераций уменьшается на 1 с каждым случаем использования пароля.

Рассмотрим пример такой серии, когда клиент очередной раз пытается получить доступ на сервер (рис. 22.10).

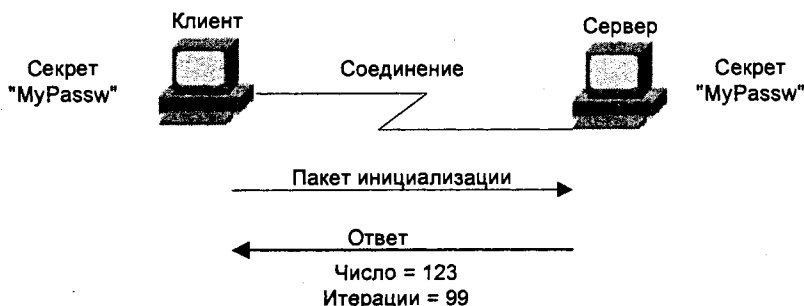


Рис. 22.10. Подготовительная фаза

Клиент складывает полученное число (123) со своим секретом ("MyPassw") и применяет к имеющемуся значению хэш-функцию столько раз, сколько указано в числе итераций (в нашем примере — 99). Полученный 64-битный результат и будет представлять собой одноразовый пароль. Поскольку его ввод требует участия пользователя, данный пароль представляется в виде 6 блоков по 11 бит и заменяется на 6 коротких английских слов (от 1 до 4 букв) из фиксированного словаря в 2048 слов.

Блок из словаря, приведенного в [RFC-1760]:

"A", "ABE", "ACE", "ACT", "AD", "ADA", "ADD", "AGO", "AID", "AIM",
 "AIR", "ALL", "ALP", "AM", "AMY", "AN", "ANA", "AND", "ANN", "ANT",
 "ANY", "APE", "APS", "APT", "ARC", "ARE", "ARK", "ARM", "ART", "AS",
 "ASH", "ASK", "AT", "ATE", "AUG", "AUK", "AVE", "AWE", "AWK", "AWL",
 "AWN", "AX", "AYE", "BAD", "BAG", "BAH", "BAM", "BAN", "BAR", "BAT",
 "BAY", "BE", "BED", "BEE", "BEG", "BEN", "BET", "BEY", "BIB", "BID",
 "BIG", "BIN", "BIT", "BOB", "BOG", "BON", "BOO", "BOP", "BOW", "BOY",
 . . .

Пусть, например, паролем окажется "MAST SONG THAT WOOD LIFE HUNT".

А вот вариант такого же словаря, который может быть использован для русскоязычных пользователей, которым трудно запоминать английские слова. Данная версия предлагается как проект, при конкретной реализации естественно можно воспользоваться любой словарной комбинацией. Необходимо только, чтобы на обеих сторонах, у клиента и у сервера, был один и тот же вариант словаря.

"А", "АД", "АЗ", "АЙ", "АР", "АС", "АХ", "АГА", "АКР",
 "АКТ", "АРА", "АТУ", "АБАК", "АВИА", "АВТО", "АГАТ", "АДАМ", "АЖУР",
 "АЗИЯ", "АЗОТ", "АИСТ", "АЙВА", "АЛЕФ", "АЛЛА", "АЛЛО", "АЛОЭ", "АЛЫЙ",
 "АЛЬТ", "АМИН", "АМУР", "АНДЫ", "АНИС", "АННА", "АНОД", "АНТИ", "АРАБ",
 "АРАК", "АРБА", "АРГО", "АРКА", "АРФА", "АРЫК", "АТОМ", "АХТИ", "АЭРО",
 "БА", "БЫ", "БАЙ", "БАК", "БАЛ", "ВАН", "ВАР", "ВАС", "БАТ",
 "БАХ", "БАЦ", "БАШ", "БЕГ", "БЕЗ", "ВЕЙ", "ВЕН", "БЕС", "БИО",
 . . .

В русскоязычной версии пароль будет выглядеть как "МЫЛО ПЫЛЬ СРЕЗ ТЕСТ МЯСО КРЫМ". Он вводится пользователем и высылается серверу.

Сервер хранит у себя *последний успешный пароль* серии, т. е. пароль, с которым клиент последний раз получал доступ к серверу. Учитывая, что число итераций уменьшается на 1 с каждым использованием пароля, в нашем примере, при последнем успешном подключении к ключу + числу хэш-

функция применялась 100 раз и пароль был "NICE MAIN HAND BEST FAST RATE" (рис. 22.11).

Теперь серверу необходимо всего лишь применить к полученному от клиента паролю хэш-функцию 1 раз и сравнить полученное значение с хранящимся у него последним успешным паролем, если они совпадают — аутентификация прошла удачно. Предполагая, что используемая хэш-функция необратима, злоумышленник, даже перехватив текущий одноразовый пароль (например, со 100 итерациями), не может предугадать следующий пароль серии (с 99 итерациями).

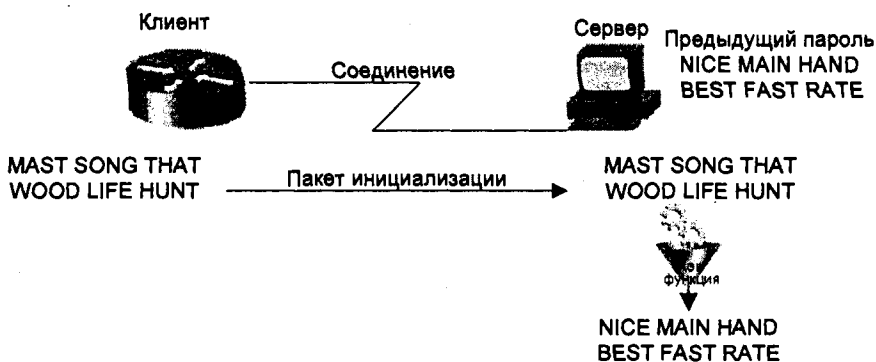


Рис. 22.11. Верификация

По истечении количества итераций клиенту и серверу необходимо обновить случайное число, количество итераций и, возможно, секрет.

Terminal Access Controller Access Control System

TACACS представляет собой семейство или даже поколение протоколов, наиболее известные из которых это *TACACS* [RFC-1492], *XTACACS* и *TACACS+*. TACACS — это UDP-ориентированный протокол, разработанный BBN для MILNET, совмещающий процессы аутентификации и авторизации. XTACACS — это модификация протокола компанией Cisco Systems, Inc., разделяющая процессы аутентификации, авторизации и учета (англ. *accounting*). TACACS+ — это расширение XTACACS с дополнительными возможностями контроля и учета, использующее в качестве транспорта TCP. За данным протоколом зарезервированы TCP и UDP порты 49.

TACACS — клиент/серверный протокол, где в качестве клиента обычно выступает *сервер сетевого доступа* (англ. *Network Access Server* — NAS), с которым соединяется субъект-пользователь (например, по коммутируемому каналу), а в качестве сервера — объект (авторизационный сервер). TACACS+ шифрует обмен информацией между клиентом и сервером на основе обще-

го секрета, который также используется для подтверждения аутентичности самой транзакции.

Процесс *аутентификации* TACACS может использовать различные имеющиеся механизмы, такие как PPP PAP, CHAP, EAP, а также Kerberos или токенные карты. Вообще говоря, аутентификация не обязательна в данном протоколе (процесс авторизации), она может не присутствовать вовсе или присутствовать только при обращении к конкретным службам (сервисам). Аутентификационный пакет бывает трех типов:

- ❑ тип START, формируемый и высылаемый клиентом;
- ❑ тип CONTINUE, формируемый и высылаемый клиентом;
- ❑ тип REPLY, формируемый и высылаемый сервером.

Аутентификация при первом соединении или после перестартовки сессии всегда начинается пакетом START, который содержит указание на используемый механизм аутентификации (например, PAP) и может содержать аутентификационные данные (например, имя пользователя и пароль). Сервер отвечает пакетом REPLY, который указывает, что аутентификация закончена или требует продолжения (в этом случае указывается, какая дополнительная информация требуется), а также может содержать указание перестартовки сессии. Если клиент получил запрос на дополнительную информацию, он высылает пакет CONTINUE с требуемыми данными. Процесс повторяется до завершения авторизационного процесса.

Напомним, что процесс *авторизации* — это определение того, что разрешено делать данному пользователю. Обычно для этого необходимо знать, что это за пользователь (то есть произвести его аутентификацию). Если аутентификация не производилась, к пользователю применяются правила предоставления доступа по умолчанию. Для авторизационной сессии используется два типа пакета:

- ❑ тип REQUEST, формируемый и высылаемый клиентом;
- ❑ тип RESPONSE, формируемый и высылаемый сервером.

В пакете REQUEST присутствует фиксированный набор полей, которые подтверждают аутентичность пользователя или процесса, обратившегося с запросом, и набор параметров, которые указывают, какие сервисы и возможности запрашиваются.

В пакете RESPONSE не просто указываются, разрешено или запрещено использование запрошенных пунктов, но и дополнительные параметры, такие, например, как временной промежуток, в течение которого разрешено использовать данный сервис.

Учет — это процесс фиксирования того, что пользователь делает или сделал. Обычно он следует за аутентификацией и авторизацией и служит целям

учета использования сервисов (например, биллинг) или для аудита безопасности. Для учета используется три типа записей:

- ❑ тип Start records, указывающий, что использование сервиса началось;
- ❑ тип Stop records, указывающий, что использование сервиса прекращено;
- ❑ тип Update records, промежуточный, указывающий, что сервис еще используется.

Эти записи содержат всю информацию, используемую при авторизации, а также дополнительную информацию, такую как время начала и завершения авторизации, использование, потребление ресурсов и т. п.

Общая схема работы по протоколу TACACS+ приведена на схеме 22.12.

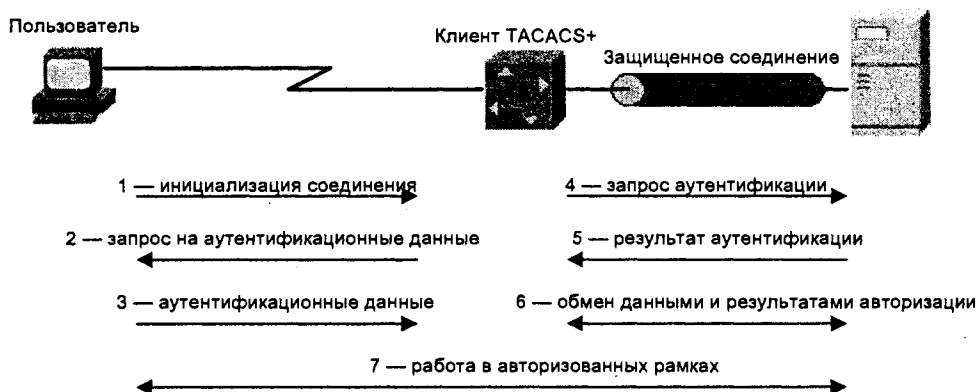


Рис. 22.12. Общая схема работы

Рассмотрим подробнее структуру пакета и работу клиента и сервера по RFC. В нем отдельно рассматриваются UDP- и TCP- представления протокола.

Для UDP представлены *простая* и *расширенная* формы, где простая является подмножеством расширенной. На рис. 22.13 показаны границы байтов, слов и двойных слов.

Поля с 1 по 5 и 14 (последнее) составляют простую форму.

Версия — указывает номер версии протокола.

Тип — указывает тип запроса и бывает следующим:

- ❑ LOGIN;
- ❑ RESPONSE (только при ответе сервера клиенту);
- ❑ CHANGE;
- ❑ FOLLOW;

- ☐ CONNECT;
- ☐ SUPERUSER;
- ☐ LOGOUT;
- ☐ RELOAD;
- ☐ SLIPON;
- ☐ SLIPOFF;
- ☐ SLIPADDR;

причем значения ряда полей не определены в данном RFC.

Версия	Тип	Соответствие	
Длина имени пользователя	Длина пароля	Ответ	Причина
Результат 1			
Адрес назначения			
Порт назначения		Строка	
Результат 2			
Результат 3		Данные	

Рис. 22.13. Структура запроса для UDP

Поле *Соответствие* позволяет определять соответствие между запросом и ответом.

Значения полей *Длина имени пользователя* и *Длина пароля* указывают размеры соответствующих значений.

Поле *Ответ* имеет два значения, устанавливаемые сервером: 1 — принято и 2 — отказано. Другие значения зарезервированы для будущего использования.

В поле *Причина* указываются следующие значения, связанные со значениями других полей:

- 0 — пустое значение;
- 1 — окончание срока действия;
- 2 — пароль;
- 3 — запрещено;
- 4 — выход (нормальное окончание сессии пользователем);

- 5 — таймаут;
- 6 — потеря несущей;
- 7 — слишком много неверных паролей.

Поле *Результат* = 1 используется сервером для ответа на запросы LOGIN или CONNECT.

Поля *Адрес назначения*, *Порт назначения* и *Строка* указывают запрашиваемые клиентом адрес, порт и номер строки.

Поля *Результат* = 2 и *Результат* = 3 используются сервером, как и поле *Результат* = 1.

Поле *Данные* содержит имя пользователя и пароль без разделителей.

Для TCP-запроса (рис. 22.14) указано, что он не совместим с историческими версиями протокола TACACS. Запрос состоит из четырех строк ASCII текста, разделяемых парой символов "CR, LF".

Версия	Тип	Параметры
Имя пользователя		
Пароль		
Строка		

Рис. 22.14. Структура запроса для TCP

Назначение поля *Версия* аналогично варианту для UDP.

Поле *Тип* определяет тип запроса и может быть:

- ☐ AUTH;
- ☐ LOGIN;
- ☐ CONNECT;
- ☐ SUPERUSER;
- ☐ LOGOUT;
- ☐ SLIPON;
- ☐ SLIPOFF.

Поля *Имя пользователя* и *Пароль* содержат соответствующие значения в текстовом виде.

Поле *Строка* указывает номер строки (англ. *line*), запрашиваемой клиентом.

Для ответа на запрос в TCP варианте используется другой формат, изображенный на рис. 22.15.

Номер	Текст

Рис. 22.15. Структура ответа для TCP

Номер — это трехзначное число, определяющее ответ.

Текст — читабельное разъяснение результата.

Первая из трех цифр *Номера* определяет ответ в целом следующим образом:

- 1 — предварительное одобрение;
- 2 — полное одобрение;
- 3 — промежуточное одобрение;
- 4 — временный отказ;
- 5 — постоянный отказ.

Ряд номеров закреплён, чтобы обеспечить единообразие ответов различных реализаций серверов:

- 201 — принято: (результат для LOGIN или CONNECT);
- 202 — принято, пароль истекает: (результат для LOGIN или CONNECT);
- 401 — нет ответа, попробуйте еще;
- 501 — неверный формат;
- 502 — в доступе отказано.

Remote Authentication Dial-In User Service

Последний из действующих в настоящее время RFC в целом по протоколу RADIUS носит номер [RFC-2865]. Кроме того, существуют дополнительные RFC по отдельным аспектам RADIUS (учету, атрибутам, расширениям и т. п.).

Из важных общих замечаний к протоколу необходимо отметить следующие.

- ❑ Опыт показал, что вследствие отсутствия контроля за перегрузкой, протокол может снижать производительность и терять данные в масштабных системах.
- ❑ В качестве транспорта протокол использует UDP (в документе указаны четыре причины использования UDP вместо TCP:
 - возможность использования вторичного сервера при сбое первичного;
 - отличие в требованиях к синхронизации;
 - отсутствие контроля состояния;
 - обеспечение простоты при использовании сервера.

Указано, что ранние версии RADIUS использовали порт 1645, который конфликтовал с сервисом *datametrics*, современные версии должны использовать порт 1812.

- ❑ Как и TACACS, это клиент/серверный протокол, где в качестве клиента обычно выступает *сервер сетевого доступа* (NAS — Network Access Server), с которым соединяется субъект-пользователь, а в качестве сервера — объект (авторизационный сервер). RADIUS аутентифицирует транзакции на основе общего секрета между ним и NAS, не передаваемого по сети, и шифрует им же пароли пользователей при пересылке между клиентом и сервером.
- ❑ Один RADIUS сервер может выступать в качестве *посредника* (англ. *proxy*) между клиентом и другим RADIUS сервером, т. е. направлять второму серверу запросы клиента и возвращать клиенту ответы второго сервера.

Когда пользователь пытается получить доступ к объекту через клиента (в нашем случае NAS), который использует для аутентификации RADIUS сервер, то пользователь должен предоставить данному клиенту необходимую аутентификационную информацию. Клиент формирует *Запрос* на доступ (англ. *Access-Request*), содержащий имя пользователя, его пароль, идентификатор клиента и идентификатор порта, к которому пользователь пытается получить доступ. При этом пароль скрывается путем шифрования алгоритмом MD5 по схеме CFB (подробно этот механизм описан в *главе 20*) с использованием в качестве начального вектора (Initial Vector — IV) уникального номера *Запроса*, а в качестве ключа — заранее установленного между NAS и RADIUS *Секрета*.

Данный *Запрос* направляется RADIUS серверу, причем повторяется несколько раз, если ответ от сервера не приходит в заданное время. *Запросы* могут отправляться другим серверам, если первичный сервер дал сбой.

При получении *Запроса* сервером он идентифицирует клиента и не обрабатывает *Запросы* от клиентов, с которыми у него нет общего *Секрета*. Если клиент одобрен, происходит анализ имени пользователя, указанного в *Запросе*. Сервер просматривает базу данных пользователей на предмет информации о данном пользователе, такой как пароль, идентификатор клиента и порта, с которыми ему разрешено работать. При этом сервер может обратиться к другим серверам за необходимой информацией (в этом случае он сам выступает в роли RADIUS-клиента). Если же в *Запросе* присутствуют атрибуты, указывающие, что RADIUS-сервер должен выступить в качестве посредника (проху), то вся информация из *Запроса* без изменения безусловно копируется в соответствующий пакет.

Если условия, необходимые для анализа не выполняются, сервер формирует и отправляет ответ *Отказ в доступе* (англ. *Access-Reject*), при необходимости сопровождая его соответствующей текстовой информацией, которая будет отображена пользователю.

При соблюдении условий сервер может сразу аутентифицировать клиента пакетом *Подтверждение доступа* (англ. *Access-Accept*) либо потребовать от него дополнительной информации, формируя в этом случае *Требование к доступу* (англ. *Access-Challenge*), которое может содержать текстовую информацию для пользователя, необходимую для ответа на данный вопрос. Необходимо учесть, что RADIUS поддерживает аутентификацию пользователей по протоколам PAP и CHAP. Клиент, получив *Требование*, ожидает от пользователя ответ, затем формирует новый *Запрос* (с новым идентификатором запроса), а атрибут пароля заменяет шифрованным *Ответом пользователя* и включает атрибут *Статуса* (если таковой присутствовал в *Требовании*).

Сервер отвечает на данный запрос либо пакетом *Подтверждение доступа* либо пакетом *Отказ в доступе* либо новым пакетом *Требование к доступу*. Если производится подтверждение доступа, то в пакет вносятся дополнительные атрибуты, необходимые для предоставления сервиса (например, IP-адрес, маска подсети и прочие возможные параметры).

Рассмотрим структуру пакета RADIUS (рис. 22.16).

Код	Идентификатор	Длина
Аутентификатор		
Атрибуты		

Рис. 22.16. Общая структура пакета RADIUS

Поле *Код* (1 октет) предназначено для указания типа пакета. В настоящее время определены следующие типы:

- 1 *Запрос на доступ (Access-Request).*
- 2 *Подтверждение доступа (Access-Accept).*
- 3 *Отказ в доступе (Access-Reject).*
- 4 *Запрос по учету (Accounting-Request).*
- 5 *Ответ по учету (Accounting-Response).*
- 11 *Требование к доступу (Access-Challenge).*
- 12 *Статус Сервера (Status-Server).*
- 13 *Статус Клиента (Status-Client).*
- 255 Зарезервировано.

Использование типов 4 и 5 рассматривается в другом документе (RADIUS Accounting — [RFC-2866]). Типы 12 и 13 — экспериментальные.

Поле *Идентификатор* (1 октет) предназначено для обеспечения соответствия запрос/ответ.

Поле *Длина* (2 октета) указывает длину всего пакета, включая поля *Код*, *Идентификатор*, *Длина*, *Аутентификатор* и *Атрибуты*, и может указывать длину от 20 до 4096 байт. Если реально пакет меньше указанной длины, он не будет обработан. Все данные, превышающие указанную длину, не будут обработаны.

Поле *Аутентификатор* (16 октетов) имеет различные назначения. Для запросов это — Аутентификатор запроса (Request Authenticator) — случайное, уникальное, непредсказуемое число, обеспечивающее защиту от повторного использования перехваченных пакетов. Это значение в дальнейшем используется в качестве начального вектора схемы шифрования CFB и указывается в атрибутах. Для пакетов Подтверждения или отказа в доступе и Требования к доступу это — Аутентификатор ответа (Response Authenticator), он представляет собой следующее значение MD5 (Код + Идентификатор + Длина + Аутентификатор запроса + Атрибуты + Общий секрет) — знак "+" означает конкатенацию ("склеивание") соответствующих блоков данных.

Поле *Атрибуты* переменной длины представляет собой набор произвольного количества пар "Тип/Значение", которые содержат дополнительные данные, требующиеся для различных сервисов и прочих целей. Рассмотрим его формат — рис. 22.17.

Тип	Длина	Значение

Рис. 22.17. Формат поля Атрибуты

Поле *Тип* (1 октет) используется для указания типа атрибута. Значения этого поля не должны использовать следующие зарезервированные номера: 192–223 (экспериментальные), 224–240 (специальные), 241–255 (резервные). Из оставшихся в документе закреплены номера с 1 по 63, не перечисляя все, укажем некоторые:

Имя пользователя — 1.

Пароль пользователя (в открытом виде) — 2.

Пароль пользователя (схема аутентификации CHAP) — 2.

IP адрес NAS — 4.

Тип сервиса — 6.

Текстовое сообщение — краткий ответ пользователю — 18.

Двоичные параметры сессии в нерегламентированном формате (для частного использования) — 26.

Максимальная продолжительность сессии в секундах — 27.

Максимальное возможное бездействие в пределах сессии в секундах — 28.

Телефонный номер, набранный пользователем (в случае если он известен) — 30.

Телефонный номер пользователя (в случае если он определен) — 31.

Статус посредника (Proxy-State) — служебная информация, добавляемая RADIUS-сервером в тех случаях, когда он играет роль клиента другого RADIUS-сервера — 33.

Двоичные данные СНАР-запроса в тех случаях, когда сервер по каким-либо причинам не может разместить их в поле *Аутентификатор* — 60.

Поле *Длина* (1 октет) указывает длину записи об атрибуте, включая *Тип*, *Длину* и *Значение*.

Поле *Значение*, переменной длины, содержит значение атрибута и может быть одним из пяти типов: text (текст), string (двоичные данные переменной длины), address (32-битное целое — адрес), integer (32-битное целое) и time (32-битное целое — время в формате UNIX).

Вкратце рассмотрим документ [RFC-2866], регламентирующий *учет* (англ. *accounting*) в рамках протокола RADIUS.

Если клиент использует возможность учета, то при начале использования сервиса он формирует пакет *Начало учета* (Accounting start), указывающий тип сервиса и имя пользователя, использующего сервис, и отправляет пакет серверу учета, который подтверждает получение пакета. По окончании работы с сервисом клиент формирует и отправляет серверу пакет *Конец учета* (Accounting stop), в котором указывает тип сервиса и, возможно, статистику (время использования, объем полученных и отправленных данных и т. п.), сервер подтверждает получение.

Если сервер не присылает подтверждения, клиент совершает повторные попытки установленное число раз, а затем направляет данные альтернативному серверу. Важно отметить, что сервис учета является опциональным и никак не влияет на характер самой сессии. Если, например, NAS так и не получит от группы RADIUS-серверов подтверждения пакета *Начало учета*, то он все равно допустит пользователя к ресурсу. Более того, сначала происходит само подключение (то есть сразу по приходу ответа *Подтверждение доступа*), а уже затем, что называется, "с меньшим приоритетом" производится отправка пакетов учета.

Для обеспечения учета в пакетах используются поле *Тип* с номерами с 40 по 51.

Kerberos

Общая схема работы

Для протокола Kerberos, также как и для RADIUS существует несколько RFC, отражающие различные аспекты использования протокола (RFC-1964, RFC-2623, RFC-2712, RFC-2942). Однако существует немало весьма подробных разъяснений функционирования протокола, в том числе и на русском языке.

Протокол предназначен для аутентификации субъекта объектом (и наоборот), например сервера клиентом, в случае когда среда передачи данных открыта, а объект изначально ничего не знает о субъекте и не имеет с ним общего секрета, но оба (и субъект, и объект) предварительно идентифицированы третьей стороной — доверенным сервером и имеют с ним общие секреты (никогда не передаваемые по сети). Требование наличия такого секрета и определяет схему защиты протокола — симметричными криптографическими алгоритмами (DES). Согласно принятой терминологии субъект и объект называются *принципалами* (англ. *principals*), а доверенный сервер называют *центром распределения ключей* — ЦПК (англ. *Key Distribution Center* — KDC).

Поскольку ЦПК обеспечивает серьезную работу по аутентификации в распределенной сети, в том числе хранит ключи аутентифицируемых субъектов и объектов, к нему выдвигаются повышенные требования по безопасности (по аналогии с межсетевым экраном):

- ❑ он должен быть размещен в помещении с контролируемым физическим доступом;
- ❑ на сервере не должны быть установлены и запущены посторонние программы и процессы, не относящиеся к его прямой функциональности;
- ❑ он не должен использоваться для хранения посторонних данных и т. п.

Имеется несколько замечаний к работе протокола, сформулированных следующим образом.

- ❑ Протокол не защищает от атак класса "Отказ в обслуживании".
- ❑ Принципалы должны сохранять в тайне свои секреты — протокол не обеспечивает защиты в случае утери секрета.
- ❑ Протокол не защищает от атак типа "Разгадывание пароля": выбор надежного пароля, который служит секретом или на котором строится секрет, оставляется на усмотрение принципалов.
- ❑ Системные часы всех участников аутентификационного процесса должны быть синхронизированы, обеспечение безопасности синхронизации отнесено к функциям дополнительного протокола синхронизации.

- Новые принципалы не должны наследовать идентификаторы старых, удаленных из системы (сети), иначе они могут унаследовать и их права доступа.

Рассмотрим общую схему аутентификации в описанных условиях. Субъект (клиент) хочет получить доступ к каким-либо ресурсам на объекте (сервере ресурсов), который ничего о клиенте не знает. Чтобы предоставить доступ к ресурсам, сервер должен первоначально аутентифицировать клиента. Поскольку общие данные о клиенте и о сервере есть у доверенного сервера (доверенный сервер имеет с ними общие секреты, т. е. для простоты — ключи симметричного шифрования), то схема определяется следующим образом:

- доверенный сервер генерирует некий сессионный ключ;
- шифрует сессионный ключ ключом клиента и отправляет клиенту;
- шифрует сессионный ключ ключом сервера и отправляет серверу.

Теперь клиент и сервер обладают единым сессионным ключом, который, с одной стороны, позволяет им доверять друг другу (так как оба они доверяют серверу, предоставившему сессионный ключ), с другой стороны, позволяет построить защищенное соединение для обмена данными (рис. 22.18).

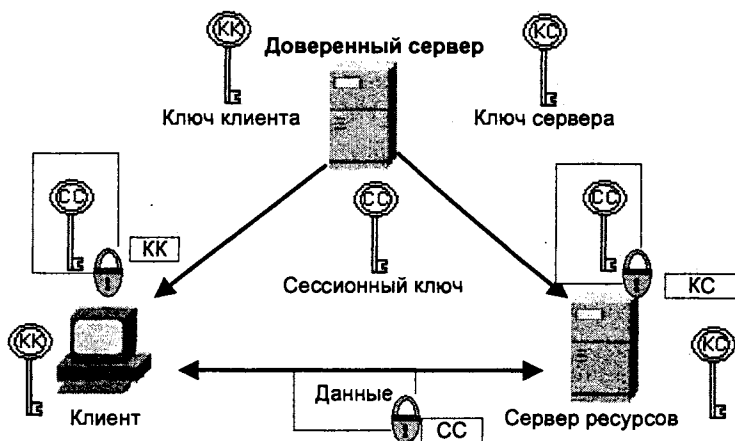


Рис. 22.18. Возможная схема работы

В принятой в протоколе Kerberos терминологии сессионный ключ с сопутствующими данными (наименование принципала, временной штамп и т. п.), зашифрованный ключом принципала, называется *билетом* (англ. *ticket*), или *мандатом*.

При реализации протокола указанная возможная схема была несколько усложнена для обеспечения скорости и надежности. Действительно, если кли-

ент уже получил сессионный ключ, а сервер ресурсов из-за особенностей сети еще нет, то клиент не сможет работать с сервером. С другой стороны, если у доверенного сервера достаточно много клиентов, которые продолжают запрашивать доступ к различным серверам ресурсов, то ему очень часто придется искать в своей базе ключи этих клиентов, а клиенту, соответственно, придется все время держать наготове свой постоянный ключ для связи с доверенным сервером, что небезопасно.

В рабочей схеме Kerberos доверенный сервер, во-первых, формирует дополнительный билет для доступа клиента к самому доверенному серверу (с другим сессионным ключом). Теперь клиенту достаточно помнить только этот новый сессионный ключ, а свой основной ключ он может убрать до следующей регистрации на доверенном сервере. Во-вторых, все билеты (как к серверам ресурсов, так и к самому доверенному серверу) отсылаются только клиенту. Теперь уже забота клиента предоставить вместе с запросом на предоставление ресурса и билет для данного сервера. А уж сервер, используя свой постоянный ключ, извлечет из билета сессионный ключ, который будет использовать для работы с клиентом. Если клиент хочет, чтобы сервер также аутентифицировал себя, он требует от сервера, чтобы тот извлек из клиентского запроса (зашифрованного сессионным ключом) определенные данные (временную метку) и вернул ее клиенту. Таким образом, клиент сможет убедиться, что сервер успешно дешифровал свой билет и, значит, является тем, за кого себя выдает (во всяком случае, по мнению доверенного сервера) (рис. 22.19).

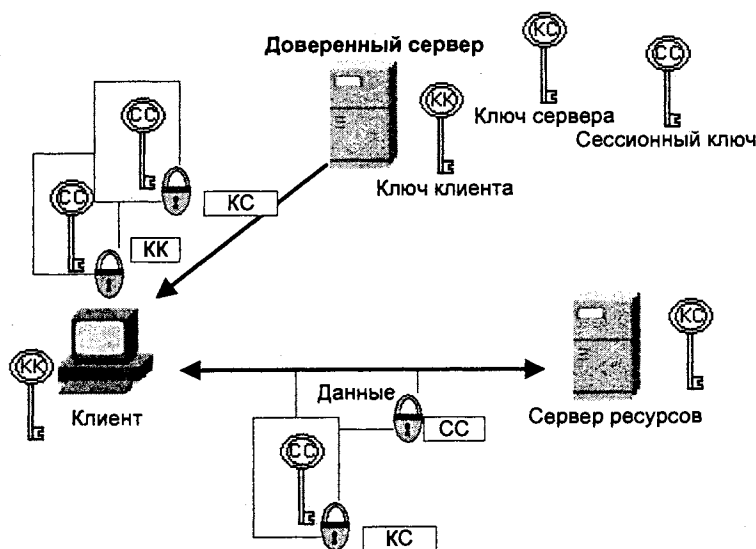


Рис. 22.19. Принципиальная схема Kerberos

Теперь и доверенный сервер не должен искать постоянный ключ клиента в базе, клиент сам при новом запросе предоставит ему билет, в котором находится сессионный ключ для доверенного сервера.

В принятой в протоколе Kerberos терминологии билет для работы с доверенным сервером называется *билетом на получение билетов* (англ. *ticket-granting-tickets* — *TGT*), а билет для сервера ресурсов — *билетом на получение сервиса* (англ. *ticket-granting-service* — *TGS*). Все указанные билеты и связанные с ними ключи действительны только до срока истечения билета или до выхода клиента из сети. Они сохраняются только в оперативной памяти, в т. н. *кэш-памяти билетов* (англ. *credentials cache*) и удаляются по истечении срока или окончании работы клиента в сети.

Таким образом, работа клиента складывается из следующих шагов, каждый из которых составляет отдельный подпротокол протокола Kerberos.

Регистрация в сети: подпротокол "Обмен данными со службой аутентификации"

1. При первой регистрации клиента (пользователя) требуется ввод пароля, который преобразуется в ключ шифрования (стандартный метод преобразования — DES-CBC-MD5, но могут быть использованы и другие алгоритмы). Этот ключ, который считается постоянным или долговременным (до следующей смены пароля пользователем), сохраняется в кэш-памяти.
2. Клиент формирует запрос на аутентификацию (*KRB_AS_REQ* — Kerberos Authentication Service Request), в который включаются идентификатор пользователя (для поиска его ключа), имя службы выдачи билетов, а также предварительные аутентификационные данные, которые могут предотвратить обращение злоумышленника за билетом от имени клиента (например, зашифрованная ключом клиента временная метка).
3. Клиент отправляет ЦПК запрос *KRB_AS_REQ*.
4. Доверенный сервер, получив запрос, ищет в своей базе соответствующий клиенту ключ и проверяет предварительные аутентификационные данные.
5. После успешной проверки ЦПК формирует сессионный ключ и готовит ответ на запрос клиента (*KRB_AS_REP* — Kerberos Authentication Service Reply), куда включает копию сессионного ключа, зашифрованного ключом клиента, и билет *TGT*, зашифрованный своим постоянным ключом, содержащий еще одну копию сессионного ключа для себя и авторизационные данные клиента.
6. Доверенный сервер отправляет клиенту ответ *KRB_AS_REP*.

7. Клиент, получив ответ, дешифрует сессионный ключ, сохраняет его и TGT-билет в кэш-памяти, откуда удаляет свой постоянный ключ. Теперь общаться с ЦПК он будет, используя сессионный ключ.

Получение права на доступ к серверу ресурсов: подпротокол "Обмен билетами на получение сервиса"

1. Клиент решает поработать с сервером ресурсов. Для этого клиент формирует запрос KRB_TGS_REQ — Kerberos Ticket-Granting-Service Request, куда включаются имя пользователя и аутентификационные данные, зашифрованные сессионным ключом для ЦПК, билет TGT, наименование сервера ресурсов.
2. Клиент отправляет запрос KRB_TGS_REQ доверенному серверу.
3. ЦПК, получив запрос, извлекает с помощью своего постоянного ключа сессионный ключ из TGT, а с помощью этого сессионного ключа проверяет аутентификационные данные клиента.
4. ЦПК генерирует второй сессионный ключ для работы клиента и сервера ресурсов.
5. ЦПК формирует ответ KRB_TGS_REP — Kerberos Ticket-Granting-Service Reply, куда помещает новый сессионный ключ, зашифрованный сессионным ключом клиента, и TGS, который состоит из того же нового сессионного ключа, а также данные о клиенте, необходимые для авторизации его на сервере ресурсов, — все это зашифровано постоянным ключом сервера ресурсов.
6. Доверенный сервер отправляет ответ KRB_TGS_REP клиенту.
7. Клиент, получив ответ, с помощью своего сессионного ключа извлекает новый сессионный ключ для работы с сервером ресурсов и сохраняет его в кэш-памяти вместе с TGS для сервера ресурсов. Теперь у клиента есть сессионный ключ и TGT для доверенного сервера и сессионный ключ и TGS для сервера ресурсов.

Использование сервера ресурсов: подпротокол "Обмен данными между клиентом и сервером"

1. Клиент формирует запрос KRB_AP_REQ — Kerberos Application Request, в который он включает свои аутентификационные данные, зашифрованные сессионным ключом для работы с сервером ресурсов, TGS для данного сервера, а также (опционально) требование произвести взаимную аутентификацию.
2. Клиент отправляет запрос KRB_AP_REQ серверу ресурсов.
3. Сервер ресурсов с помощью своего постоянного ключа извлекает авторизационные данные для определения прав клиента и сессионный ключ

для клиента из TGS, с помощью этого сеансового ключа дешифрует аутентификационные данные клиента и проверяет требование взаимной аутентификации. Если такое требование присутствует, он включает в ответ KRB_AP_REP (Kerberos Application Reply) временную метку, зашифрованную сессионным ключом для клиента.

4. Сервер отправляет ответ KRB_AP_REP клиенту.
5. Клиент, получив ответ (и проверив метку времени), убеждается в готовности сервера к работе (и его подлинности) и может начать работу с сервером. При этом данные могут шифроваться сессионным ключом, либо клиент и сервер могут договориться о другом ключе.

Характеристики протокола Kerberos

Одно из преимуществ протокола заключается в том, что клиент может быть аутентифицирован и получить доступ к серверу ресурсов вне своей *области аутентификации* — домена (англ. *realm*) — ЦРК2, при условии, что оба доверенных сервера ЦРК1 и ЦРК2, обслуживающих свои области, установили между собой доверительные отношения (определили общий ключ шифрования при пересылке данных между областями). В этом случае ЦРК2 выступает для клиента в качестве такого же сервера ресурсов, только удаленного и предоставляющего сервис трансляции запросов клиента подведомственным ему серверам ресурсов.

Более того, возможна ситуация, когда клиенту необходим ресурс из области, с которой у его ЦРК1 нет доверительных отношений, но ЦРК обеих областей имеют доверенные отношения с третьим ЦРК3. В этом случае клиент также может быть аутентифицирован на сервере вне своей области аутентификации (рис. 22.20).

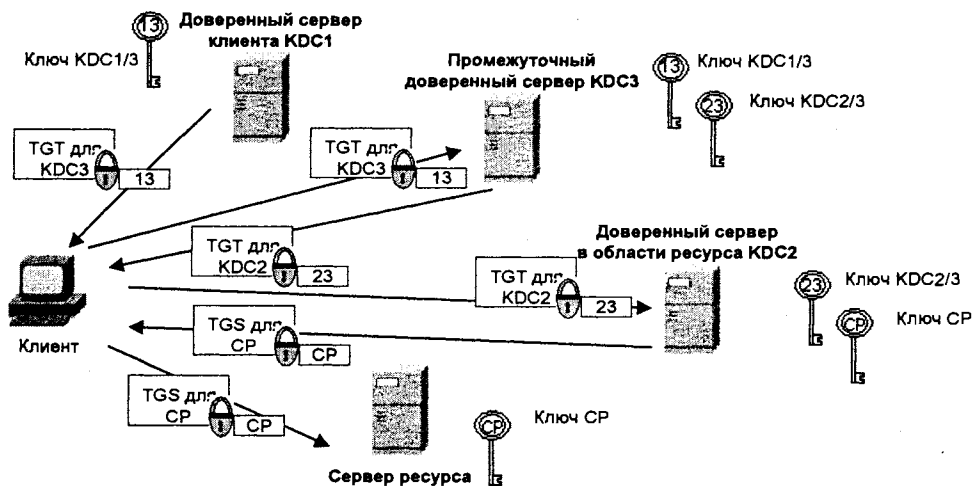


Рис. 22.20. Схема работы с промежуточным доверенным сервером

Сам по себе протокол Kerberos служит только для аутентификации (то есть подтверждения, что клиент является тем, за кого себя выдает), но не для авторизации (определении того, в рамках каких прав клиент может работать). Однако он предусматривает в билете поле данных, необходимых для применения авторизационных правил системой, для которой клиент будет аутентифицирован.

Общий формат билета следующий.

1. Поле *Tkt-VNo* указывает номер версии протокола (текущая версия — 5).
2. Поле *Realm* указывает имя области, где создан билет, т. е. доверенного ЦРК.
3. Поле *SName* указывает имя сервера, для которого предназначен билет.
4. Первые три поля не зашифрованы, остальные данные зашифрованы.
5. Поле *Flags* — флаги билета (рассматриваются далее в *данном разделе*).
6. Поле *Key* — сессионный ключ шифрования.
7. Поле *CRealm* — наименование области, где зарегистрирован клиент и где происходит первичная аутентификация.
8. Поле *CName* — идентификатор самого клиента.
9. Поле *Transited* — наименование областей, которые принимали участие в аутентификации клиента (не в порядке последовательности аутентификации).
10. Поле *AuthTime* — указывается время первоначальной аутентификации клиента из самого первого билета текущей сессии работы клиента (пользователя).
11. Поле *StartTime* — время вступления билета в силу; если не указано, берется равным предыдущему полю.
12. Поле *EndTime* — время окончания срока действия билета вместе с предыдущим полем составляют срок жизни билета. Некоторые службы устанавливают свои ограничения по времени и могут отвергнуть билет, который фактически не достиг срока истечения.
13. Поле *ReNew-Till* — используется совместно с флагом RENEWABLE из поля *Flags*, определяет максимальное время, до которого билет может быть обновлен.
14. Поле *CAddr* — указываются адреса, с которых клиент может использовать данный билет. Если поле пустое — билет можно использовать с любого адреса.
15. Поле *Authorization-Data* — данные о клиенте, которые могут быть использованы системой, получившей билет для применения к клиенту авторизационных правил.

Прежде чем изучить содержимое поля флагов, рассмотрим два важных момента протокола Kerberos — обновление билетов и делегирование полномочий.

Ограничение срока действия билета используется протоколом как одно из средств защиты от компрометации сессионного ключа. В течение срока действия билета клиент может применять (предъявлять) его неограниченное число раз. По истечении срока действия клиенту необходимо получить новый билет, основная задача которого — использовать новый сгенерированный ЦПК сессионный ключ. При этом клиенту придется снова задействовать свой долговременный ключ, т. е. пользователю снова придется набирать свой пароль.

Однако возможен другой способ — регенерация ключей без переформирования всего билета целиком. В этом случае в билете указываются два срока действия — первый, это срок до регенерации текущего ключа, второй — срок действия всего билета с учетом производимых регенераций ключей. Теперь клиенту необходимо позаботиться о том, чтобы билет был представлен ЦПК до срока истечения возможного обновления, и в этом случае билет будет обновлен (будет сгенерирован новый сессионный ключ и указано новое время истечения обновления).

Делегирование полномочий предусмотрено для случаев, когда клиент обращается к серверу ресурсов, который в свою очередь должен будет обратиться к другому серверу без непосредственного участия клиента. Для этого клиент должен передать свои права на аутентификацию первому серверу. Для этого существуют два способа: если клиент знает наименование второго сервера, он может запросить у ЦПК билет на доступ к этому серверу и передать данный билет первому серверу — т. н. *билет посредника* (англ. *proxy ticket*); если клиент не знает наименования второго сервера, то он передает первому серверу свой TGT, который тот использует для запроса нового билета для доступа ко второму серверу — т. н. *переадресованный билет* (англ. *forwarded ticket*).

Рассмотрим подробнее поле *Flags*, учитывая новые возможности обновления и делегирования. Это 32-битовое поле, в котором присутствие флага указывается включением соответствующего бита поля из 0 в 1. Биты имеют следующие значения.

0 RESERVED — зарезервировано для будущего использования.

1 FORWARDABLE — флаг указывает службе формирования билетов, что на основании данного билета TGT можно выпустить новый с другим сетевым адресом, для передачи его серверу.

2 FORWARDED — флаг указывает, что данный билет переадресован или выпущен на основе переадресованного билета.

3 PROXIABLE — флаг указывает, что ЦПК может на основе данного билета генерировать билеты с другим сетевым адресом.

4 PROXY — указывает, что сетевой адрес данного билета отличается от TGT, на основе которого он был выпущен.

Следующие два флага используются для случаев, когда ряду служб необходимо использовать в своих процедурах билеты не для аутентификации, а для других целей. Это могут быть *пакетные процедуры* (англ. *batch procedures*) с использованием билетов различных клиентов. Оставлять билеты активными в этом случае опасно. Эти флаги позволяют использовать билеты в таких процедурах, но оставлять их при этом *неактивными* (англ. *dormant*).

5 MAY-POSTDATE — удостоверяет, что на основе данного билета можно выпустить билет, который активизируется позже.

6 POSTDATED — указывает, что билет помечен как активизируемый позже.

7 INVALID — подтверждает, что сервер ресурсов не должен рассматривать данный билет, он должен быть проверен ЦПК перед использованием.

8 RENEWABLE — указывает, что билет может быть обновлен.

9 INITIAL — показывает, что это билет TGT.

10 PRE-AUTHENT — указывает, что клиент был предварительно аутентифицирован службой ЦПК перед выпуском билета.

11 HW-AUTHENT — подтверждает, что для предварительной аутентификации клиенту потребовалось специальное аппаратное обеспечение.

12—31 — оставлены для будущего использования.

Secure European System for Applications in a Multivendor Environment

Европейскую Систему Безопасности Приложений в Смешанных Программных Средах (SESAME) в общем-то трудно назвать протоколом — это скорее модель или архитектура безопасности. Однако это название иногда упоминается рядом с Kerberos, мы решили рассмотреть его и будем называть схемой. Так как SESAME получил широкое распространение в основном только в Западной Европе, коснемся его только в общем виде.

Схема SESAME была предназначена для обеспечения безопасности при работе в незащищенной сети, например, таких известных своими уязвимостями приложений, как Telnet, RLogin, RSh, RCP и RPC, и ставила своими целями обеспечение следующих задач.

- ☐ Аутентификация (односторонняя или взаимная) с помощью Kerberos или механизмов шифрования с открытым ключом.
- ☐ Обеспечение защиты данных при передаче.
- ☐ Создание *ролевой модели доступа* (англ. *Role-based access control — RBAC*).

- ☐ Делегирование прав.
- ☐ Служба аудита.
- ☐ Мультидоменная поддержка.

Рассмотрим основные составляющие схемы (рис. 22.21).

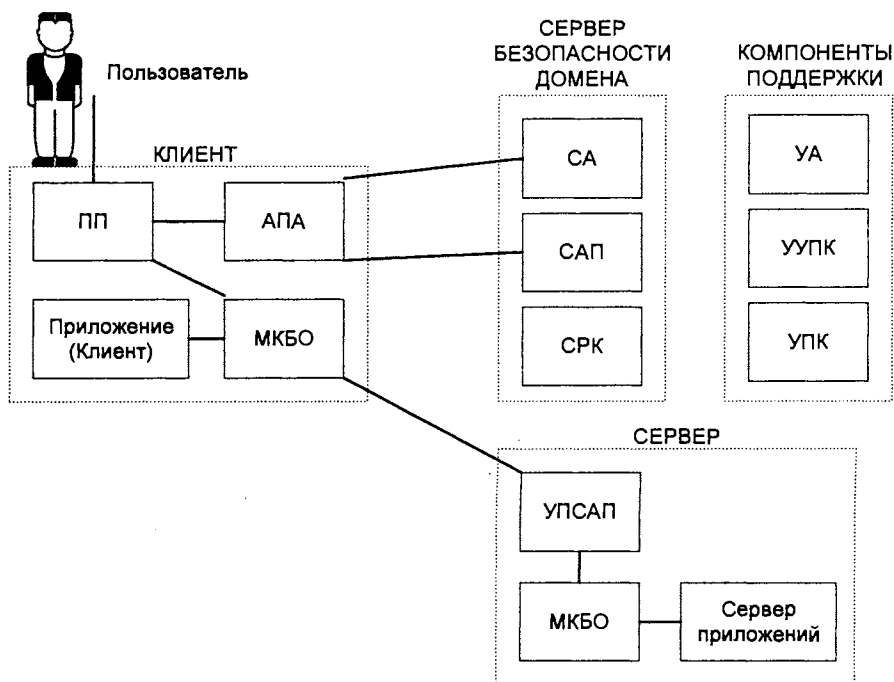


Рис. 22.21. Компоненты SESAME

ПП — *посредник (покровитель) пользователя* (англ. *User Sponsor*) — предоставляет пользователю интерфейс к системе SESAME, предлагая пользователю функциональность для входа в систему (login).

АПА — *атрибут привилегий аутентификации* (англ. *Authentication Privilege Attribute*) — используется ПП при связи с сервером безопасности домена.

СА — *сервер аутентификации* (англ. *Authentication Server*) — аналогичен схеме протокола Kerberos, обеспечивает единый вход в сеть (single sign-on).

САП — *сервер атрибутов привилегий* (англ. *Privilege Attribute Server*) — механизм управления доступом.

СРК — *сервер распределения ключей* (англ. *Key Distribution Server*) — аналогичен СРК протокола Kerberos, обеспечивает управление криптографическими ключами, но в отличие от Kerberos поддерживает аутентификацию на публичных ключах с помощью протокола X.509.

МКБО — менеджер контекста безопасного объединения (англ. *Secure Association Context Manager*) — обеспечивает защиту данных при обмене (аутентификацию и целостность данных, неотрекаемость или апеллируемость)

УПСАП — устройство проверки САП (сертификата атрибутов привелегий) (англ. *Privilege Attribute Certificate Validation Facility*) — производит проверку прав клиента на доступ к приложению на основе САП.

УПК — устройство поддержки криптографии (англ. *Cryptographic Support Facility*) — обеспечивает реализацию конкретных криптографических алгоритмов/протоколов.

УУПК — устройство управления публичными ключами (англ. *Public Key Management Facility*) — обеспечивает работу СРК при работе с публичными ключами.

УА — устройство аудита (англ. *Audit Facility*) — обеспечивает ведение детальных регистрационных журналов.

Дополнительные используемые термины схемы: *инициатор* (англ. *initiator*) — пользователь или приложение от имени пользователя или само по себе, обращающееся за ресурсом к *приложению назначения* (англ. *target*).

Рассмотрим основные характеристики схемы SESAME.

- ❑ Обеспечение безопасности в открытой сети. Схема всегда обеспечивает целостность передаваемых данных и опционально — конфиденциальность (кроме тех случаев, когда конфиденциальность критически необходима — например, при пересылке ключей).
- ❑ Аутентификация и единый вход в сеть. Схема позволяет аутентифицировать инициатора и назначение, а также обеспечивает возможность пользователю регистрироваться в сети только один раз, даже если ему нужен доступ к различным назначениям.
- ❑ В схеме используется *активная* (англ. *push*), а не *пассивная* (англ. *pull*) аутентификационная технология. Две эти технологии отличаются характером передачи данных: в активной схеме инициатор сам отправляет свои аутентификационные данные по назначению, в пассивной — объект назначения запрашивает данные у инициатора.
- ❑ Используется он-лайн, а не офф-лайн сервера безопасности. Наиболее яркий пример офф-лайн сервера — это *Уполномоченный по Сертификатам (УС)* (англ. *Certification Authority — CA*). Подобный сервер выпускает сертификаты, которые далее могут быть сохранены, копированы и использованы без участия самого УС. Это накладывает определенные неудобства, когда в централизованной системе вся работа с аутентификационными данными должна быть учтена. Поэтому в схеме SESAME используется интерактивный (он-лайн) запрос/получение аутентификационных данных.

- ❑ Могут быть использованы различные атрибуты привилегий. Поскольку схема использования аутентификационных данных жестко не закреплена, а лишь придерживается соответствующих стандартов и разработана специально для оборудования от различных поставщиков (англ. *multi-vendor structure*), для доступа к разнообразным назначениям могут быть использованы различные атрибуты привилегий.
- ❑ В схеме различаются несколько видов *идентифицируемости* (англ. *identity*): для аутентификации, для доступа и для аудита.
- ❑ Может использоваться *политика ролей при управлении доступом* (англ. *role-based access control*), т. е. администратор определяет атрибуты привилегий не отдельно для каждого пользователя, а для предопределенной группы пользователей, объединенных общей функциональностью для выполнения конкретных задач. В этом случае задачи по администрированию существенно упрощаются при перемещении пользователя в рамках организации, увольнении и т. п.
- ❑ Использует пути доступа и делегирование. Пользователь (инициатор), обращаясь за конкретным ресурсом, может не знать, какое конкретно назначение им обладает. В этом случае аутентификационные данные инициатора передаются требующемуся назначению.

Secure-HTTP

Secure-HTTP (используются сокращения *S-HTTP* и *SHTTP*) текущая версия 1.2 — протокол, разработанный для обеспечения безопасности сообщений при использовании протокола HTTP и облегченной интеграции с приложениями, ориентированными на HTTP. В качестве ссылки на документацию протокола указывают рабочий документ IETF — **draft-ietf-wts-shhttp-02**. Сохраняя все характеристики HTTP, протокол позволяет производить аутентификацию, шифрование, электронно-цифровую подпись сообщений в любой комбинации. При этом протокол поддерживает как криптографическую схему с открытыми ключами, так и симметричную схему шифрования. Протокол поддерживает гибкое определение алгоритмов шифрования с помощью возможности, называемой *переговоры о параметрах* (англ. *option negotiation*). В ней определяются три составляющих протокола.

- ❑ *Транзакционный модуль* — т. е. будет ли запрос и/или ответ зашифрован и/или подписан.
- ❑ *Криптографические алгоритмы* — что будет использовано для шифрования (в документе указаны DES и RC2), электронно-цифровой подписи (указаны RSA и DSA).
- ❑ *Выбор сертификата* — какой из цифровых сертификатов использовать.

Следует учесть, что как протокол прикладного уровня он защищает html-документы, но оставляет открытой информацию нижележащих уровней. Это,

по-видимому, одна из причин, по которой на практике S-HTTP используется не очень широко, гораздо реже, чем протоколы безопасности транспортного уровня. Поэтому слегка коснемся особенностей его реализации.

Формирование сообщения требует наличия трех составляющих.

1. Собственно сообщение формата HTTP или другие данные.
2. Криптографические предпочтения получателя и ключевая информация.
3. Криптографические предпочтения отправителя и ключевая информация.

Для создания сообщения отправитель должен произвести интеграцию криптографических предпочтений своих и получателя и, сформировав список криптографических опций, применить его с использованием соответствующей ключевой информации.

Восстановление сообщения (в HTTP формат) требует наличия четырех составляющих.

1. Собственно S-HTTP сообщение.
2. Определенные получателем ранее криптографические предпочтения и ключевая информация.
3. Текущие криптографические предпочтения получателя и ключевая информация.
4. Определенные отправителем ранее криптографические предпочтения и ключевая информация.

Для восстановления сообщения получатель, считывая заголовки, определяет, какие криптографические преобразования были произведены, и производит обратные или восстановительные преобразования на основе ключевой информации.

Отметим, что для обеспечения криптографической защиты могут быть использованы механизмы общего распределенного секрета (вводом пароля вручную или с использованием технологии Kerberos) и обмен открытыми ключами. Возможно использование сессионного ключа с помощью предварительных защищенных транзакций или иным способом

Формат сообщения включает строку запроса, строку статуса, заголовок протокола, содержание и опции формата инкапсуляции. Интересен принцип формирования строки статуса, предложенный в документе. Строка статуса определяется ответом сервера и имеет вид "Secure-HTTP/1.2 200 OK" независимо от успешной или неудачной обработки запроса. Как указано, это сделано для предотвращения возможного анализа злоумышленником успешных и неуспешных запросов. Структуру остальных строк можно изучить в документе **draft-ietf-wts-shhttp-02.txt**.

Secure Socket Layer

Общие сведения

Secure Socket Layer (SSL) в настоящее время является, пожалуй, одним из самых популярных протоколов безопасности транспортного уровня, используемых в Интернете. Работая на транспортном уровне стека TCP/IP, он решает следующие задачи.

- ❑ Обеспечивает конфиденциальность данных, т. е. уверенность, что они не были раскрыты в ходе транспортировки между клиентом и сервером, путем шифрования данных всех вышестоящих уровней (представления и прикладного), т. е. оставляет открытой только служебную информацию уровня TCP и ниже.
- ❑ Обеспечивает аутентификацию сервера, т. е. уверенность пользователя (клиента), что он получает доступ именно к тому серверу, к которому необходимо.
- ❑ Опционально может обеспечивать аутентификацию клиента, т. е. обеспечивать уверенность сервера, что он работает с авторизованным клиентом.
- ❑ Обеспечивает целостность передаваемой информации, т. е. уверенность, что информация не была изменена в ходе транспортировки между клиентом и сервером.
- ❑ Опционально может сжимать данные для обеспечения скорости передачи.

Протокол включает в себя два подпротокола — *протокол записи* (SSL record protocol), определяющий формат передачи данных, и *протокол установки связи* (SSL handshake protocol), определяющий механизм установки соединения.

Подпротокол SSL-записи

Основная функция протокола записи — работа с данными, поступающими от уровня приложения. На первом этапе производится фрагментирование данных в блоки для дальнейшей обработки (блок 16 384 байта для версии SSL 3.0). Затем производится упаковка (при отправлении) или распаковка (при получении) данных, если эта опция была задана. Далее производится шифрование данных тем алгоритмом, который был определен для клиента и сервера, а также устанавливается *аутентификационный код сообщения* (англ. message authentication code — MAC) для обеспечения контроля целостности сообщения.

Кроме того, этот подпротокол берет на себя функции генерирования сообщений об ошибках и закрытии сессии. Рассмотрим возможные варианты служебных сообщений, которые представлены в табл. 22.1.

Таблица 22.1. Системные сообщения подпротокола SSL-записи

Сообщение	Значение сообщения
<i>Close_notify</i>	Это нормальное, не связанное с ошибкой, сообщение о закрытии сессии
<i>Unexpected_message</i>	Получено несоответствующее сообщение, которое приводит к закрытию сессии
<i>Bad_record_mac</i>	Получен неверный аутентификационный код сообщения, приводящий к закрытию сессии
<i>Decompression_failure</i>	Функция распаковки получила неверное значение, приводит к закрытию сессии
<i>Handshake_failure</i>	Получатель не способен поддержать требуемый отправителем набор параметров безопасности, приводит к закрытию сессии
<i>No_certificate</i>	Нет доступа к сертификату
<i>Bad_certificate</i>	Сертификат поврежден
<i>Unsup-ported_certificate</i>	Данный тип сертификата не поддерживается
<i>Certificate_revoked</i>	Сертификат отозван выпустившим его субъектом
<i>Certificate_expired</i>	Дата действия сертификата истекла или неверна
<i>Certificate_unknown</i>	При обработке данных сертификата возникли прочие неразрешимые проблемы
<i>Illegal_parameter</i>	Поле данных имеет неверное значение, которое приводит к закрытию сессии

Подпротокол установки связи

Протокол установки связи обеспечивает настройку множества криптографических параметров. В момент, когда клиент пытается установить соединение с сервером, обе стороны должны:

- ☐ согласовать версию протокола;
- ☐ согласовать алгоритм шифрования (выбирается наиболее сильный согласно табл. 22.2 из списка поддерживаемых обеими сторонами);
- ☐ произвести аутентификацию (с обеих сторон или выборочную);
- ☐ с помощью согласованного алгоритма асимметричного шифрования обмениваться общим секретом, на основе которого будет производиться симметричное шифрование.

Клиент посылает сообщение *Client_hello*, в котором указывает версию протокола, случайное число, номер сессии, список поддерживаемых алгоритмов шифрования и список методов компрессии. Сервер отвечает сообщением *Server_hello*, в котором содержатся те же поля и в них указан выбор сервера из представленных клиентом возможностей. Если сервер должен аутентифицировать себя, он высылает клиенту свой сертификат. В случае если у него нет соответствующего сертификата, он высылает сообщение *Server_key_exchange*, в котором в зависимости от выбранного алгоритма асимметричного шифрования устанавливается способ передачи открытого ключа сервера.

Если требуется аутентификация клиента, сервер может послать сообщение *Certificate_request*, запрашивая сертификат клиента. После этого сервер отправляет сообщение *Server_hello_done*, показывая, что он завершил свою работу и ожидает сообщений от клиента.

Клиент высылает сообщение *Client_certificate* со своим сертификатом, либо сообщение *No_certificate*, если у него нет сертификата. Затем клиент высылает сообщение *Client_key_exchange*, в котором он формирует основу для генерации распределенного секрета для симметричного шифрования. Этот предварительный секрет он шифрует открытым ключом сервера из его сертификата.

Если клиент использовал сертификат для своей аутентификации, то он генерирует сообщение *Certificate_verify* для подтверждения своего сертификата. После этого клиент формирует сообщение *Change_cipher_spec*, которое означает, что клиент произвел выбор из списка рассматриваемых параметров шифрования и перенес их в статус текущих. Затем клиент отправляет сообщение *Finished*, означающее, что этап согласования закончен, при этом это сообщение уже зашифровано в рамках новых параметров шифрования.

Сервер также производит установку текущих параметров шифрования и отвечает сообщениями *Change_cipher_spec* и *Finished* (рис. 22.22).

Алгоритмы шифрования и аутентификации, используемые в реализации SSL, представлены в следующем списке:

- ❑ DES — Data Encryption Standard и Triple-DES;
- ❑ RC2 и RC4 — Rivest Cipher 2 и Rivest Cipher 4 соответственно;
- ❑ DSA — Digital Signature Algorithm;
- ❑ RSA и RSA key exchange — Rivest-Shamir-Adleman;
- ❑ MD5 — Message Digest 5 и SHA-1 Secure Hash Algorithm 1;

Кроме того, при отдельных реализациях используются алгоритмы Skipjack и KEA — Key Exchange Algorithm.

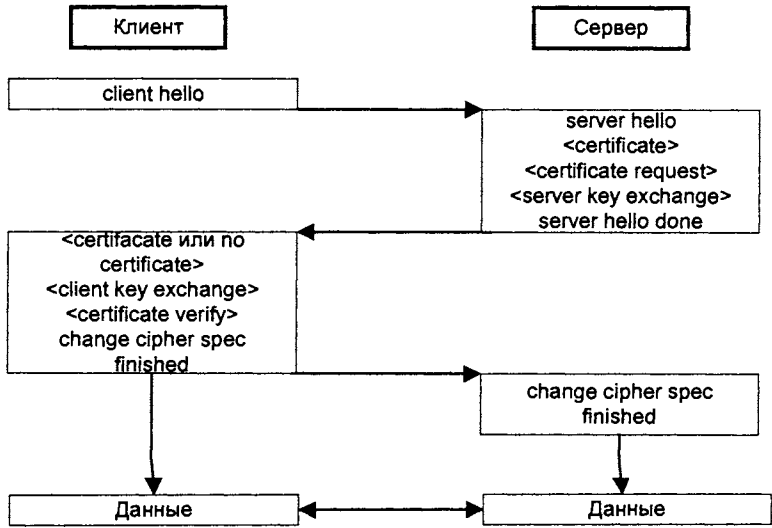


Рис. 22.22. Работа протокола установки связи, опциональные параметры

Различные варианты реализации протокола по убывающей от наиболее к наименее стойкой в криптографическом смысле конфигурации показаны в табл. 22.2. Данные представлены по состоянию до снятия ограничения на вывоз криптографии с длиной ключа более 40 бит из США.

Таблица 22.2. Параметры шифрования протокола SSL

Конфигурация		Поддержка версий
Шифрование	Аутентификация	
TripleDES (168 бит)	SHA-1 (160 бит)	SSL 2.0 и SSL 3.0
RC-4 (128 бит)	MD5 (128 бит)	SSL 2.0 и SSL 3.0
RC-2 (128 бит)	MD5 (128 бит)	SSL 2.0
DES (56 бит)	SHA-1 (160 бит)	SSL 2.0 и SSL 3.0 (но версия 2 иногда использует MD5 для аутентификации сообщений)
RC-4 (40 бит)	MD5 (128 бит)	SSL 2.0 и SSL 3.0
RC-2 (40 бит)	MD5 (128 бит)	SSL 2.0 и SSL 3.0
Без шифрования	MD5 (128 бит)	SSL 3.0

В заключение рассмотрим процессы проверки цифрового сертификата сервера клиентом и аутентификации клиента сервером.

Когда клиент получает цифровой сертификат сервера, он проверяет следующие параметры.

- ❑ Срок действия сертификата: попадает ли текущая дата в этот период, если нет — процесс обработки прекращается, если да — переходит дальше.
- ❑ Находится ли субъект, выпустивший сертификат (CA — Certification Authority), в списке доверенных CA клиента. Каждый клиент поддерживает список доверенных CA. Если субъект отсутствует в списке — сервер не аутентифицируется.
- ❑ Используя открытый ключ CA из списка доверенных CA, клиент проверяет корректность электронно-цифровой подписи на сертификате сервера. Если подпись некорректна, то предполагается, что сертификат был изменен и не принимается.
- ❑ Проверяется соответствие *имени сервера* (англ. *domain name*), указанного в сертификате, реальному имени сервера. Если они совпадают — сервер аутентифицирован.

Если сервер сконфигурирован таким образом, что он требует аутентификации клиента, процесс происходит следующим образом.

- ❑ Сервер и клиент совместно генерируют некое случайное значение, затем клиент устанавливает свою электронно-цифровую подпись на это значение.
- ❑ Сервер проверяет, соответствует ли открытый ключ клиента из сертификата клиента этой электронно-цифровой подписи.
- ❑ Сервер проверяет срок действия сертификата, попадает ли текущая дата в этот период.
- ❑ Сервер проверяет, находится ли CA, выпустивший сертификат, в списке доверенных CA сервера. Каждый сервер поддерживает список доверенных CA.
- ❑ Используя открытый ключ CA из списка доверенных CA, сервер проверяет корректность электронно-цифровой подписи на сертификате клиента.
- ❑ Сервер проверяет, находится ли сертификат клиента в списке сертификатов клиентов, которые могут быть аутентифицированы данным сервером.

Если аутентификация прошла успешно, сервер производит авторизацию — т. е. проверяет, имеет ли клиент доступ к запрашиваемым ресурсам и в зависимости от правил доступа разрешает или запрещает доступ клиента к ресурсам.

Говоря о SSL, необходимо упомянуть и протокол *TLS (Transport Layer Security — Безопасность Транспортного Уровня)*, который является в отличие от SSL официально опубликованным интернет-стандартом [RFC-2246], и популярность которого, по-видимому, будет расти. В указанном RFC говорится, что между протоколами нет кардинальных различий, и при этом TLS

обеспечивает обратную совместимость с SSL. Одним из основных преимуществ TLS следует назвать возможность его встраивания в работу приложений независимыми разработчиками и расширения криптографических возможностей.

Secure Shell

(Стандарты **draft-ietf-secsh-architecture-05**, **draft-ietf-secsh-userauth-07**, **draft-ietf-secsh-transport-07**, **draft-ietf-secsh-connect-07**.)

Secure Shell (*SSH*) — это протокол, обеспечивающий безопасную работу различных сетевых сервисов (таких как `login`) на незащищенной сети. Он состоит из трех подпротоколов.

- ❑ *Протокол транспортного уровня (SSH-TRANS)* обеспечивает аутентификацию сервера, конфиденциальность и целостность данных. Опционально может производить компрессию данных. Обычно работает поверх TCP/IP, но может быть использован и с любым надежным потоком данных (8-битовый двоичный транспорт с коррекцией ошибок).
- ❑ *Протокол аутентификации пользователя (SSH-USERAUTH)* аутентифицирует клиента перед сервером. Функционирует поверх протокола транспортного уровня.
- ❑ *Протокол соединения (SSH-CONN)* совмещает (мультиплексирует) различные логические каналы в единый зашифрованный канал. Функционирует поверх протокола аутентификации пользователя.

Рассмотрим общую архитектуру протокола и коснемся трех его составляющих. Одним из наиболее важных элементов схемы является требование аутентификации, которая производится на основе алгоритмов шифрования с открытым ключом. Это означает, что каждый участник взаимодействия (хост) должен иметь соответствующий ключ. Модель протокола допускает как вариант, когда один хост имеет множество ключей, так и вариант, когда множество хостов пользуются одним ключом. Для каждого из используемых алгоритмов хост должен иметь отдельный ключ (в качестве стандартного указан алгоритм *DSS* — *Digital Signature Standard*).

Ключ сервера используется для обеспечения уверенности клиента, что он работает с соответствующим сервером. Для этого клиент должен иметь некоторые предварительные знания о ключе сервера. Протокол допускает существование двух различных моделей обеспечения клиента такой информацией.

- ❑ *Децентрализованная.* Клиент имеет базу данных открытых ключей хостов, ассоциированных с именем хоста. Преимущество модели в том, что она не требует инфраструктуры централизованного администрирования, недостаток — в том, что поддержание локальной базы данных требует дополнительных затрат.
- ❑ *Централизованная.* Присутствует доверенная сторона CA (*certification authority*), которая производит подтверждение ассоциации хост/открытый

ключ для каждого ключа, полученного клиентом. Таким образом, клиенту достаточно хранить только ключ самого СА. С другой стороны, до начала взаимодействия, каждый сервер должен быть сертифицирован СА. Теоретически допускается в качестве опции протокола возможность использования первой сессии с хостом без подтвержденной ассоциации хост/ключ, однако указывается, что такая возможность уязвима с точки зрения безопасности для атаки класса *злоумышленник-посередине* (англ. *man-in-the-middle*).

Допускаются альтернативные методы подтверждения корректности соответствия ключа хосту, например, вычисление 16-ричного значения хэш-функции открытого ключа (указан алгоритм *SHA-1 Secure Hash Algorithm*) и далее передача его для верификации внешними каналами связи (например, по телефону).

Для конкретной реализации протокола необходимо учитывать ряд требований.

- ❑ Алгоритмы шифрования, контроля целостности и сжатия должны быть отдельными для каждого из направлений, при этом должен быть определен предпочтительный алгоритм (первый из указанного в списке для каждой категории).
- ❑ Для аутентификации хоста должны использоваться алгоритмы с открытым ключом и методы защищенного обмена ключами.
- ❑ Сервер должен требовать аутентификации для каждого пользователя. Для некоторых или всех пользователей сервер может требовать множественную аутентификацию. При этом требуемые сервером алгоритмы аутентификации могут зависеть от месторасположения, откуда пользователь пытается получить доступ.
- ❑ Операции, которые разрешены пользователю к выполнению, определяются протоколом соединения. Должны быть ясно определены допущения в работе, например, такие как запрет серверу инициировать сессии или исполнять команды на клиентской машине, а также установка соединения до его запроса клиентом. Кроме того, необходимо определить, кто и какие ТСП-порты может запрашивать для соединения. Также необходимо учитывать, что соединение может проходить через межсетевой экран и пересекаться с его собственной политикой.
- ❑ Алгоритмы шифрования, поддержания целостности и открытых ключей должны быть известны и проверены, длины ключей обеспечивать устойчивость к криптоанализу. Алгоритмы должны согласовываться, так чтобы в случае взлома одного можно было перейти на другой без модификации основного протокола.

Для работы с документами, описывающими три составляющих подпротокола, необходимо иметь представление о типах данных, используемых в протоколе:

- ❑ байт (byte) представляет некое 8-битовое значение (октет). Набор данных фиксированной длины может быть представлен, как *byte(n)*, где *n* — количество байтов в массиве;

- ❑ булевское (boolean) представляет один байт, интерпретируемый как 0 — "ложь" и 1 или любое ненулевое значение — "истина";
- ❑ uint32 — 32-битное беззнаковое целое, хранимое как четыре байта в порядке убывания значимости;
- ❑ строка (string) двоичная строка произвольной длины. При хранении строки сначала записывается длина строки *len* в формате uint32, затем следуют 0 или более байт символов — byte (*len*);
- ❑ Mprint представляет целое увеличенной точности в двойном формате, наиболее значимый бит слева. В наиболее значимом бите первого байта записан 0 для положительного числа, 1 — для отрицательного.

Сетевые адреса отображаются в виде строк, DNS-имена не используются, так как протокол DNS не обеспечивает необходимой безопасности, если в адресе встречается двоеточие (:), он интерпретируется, как IPv6.

Для использования в протоколе обозначений различных алгоритмов шифрования, хэширования, сжатия и пр., применяются идентификаторы алгоритмов в виде строк размером до 64 знаков с учетом регистров. Эти идентификаторы или имена могут быть в двух форматах.

- ❑ Имена, не содержащие символы @ и запятую, — это зарезервированные обозначения института *IANA* (*Internet Assigned Numbers Authority*), такие как (без учета кавычек): "3des-cbc", "sha-1", "hmac-sha1". Сюда входят наименования для алгоритмов шифрования, алгоритмов контроля целостности (message authentication control — MAC), алгоритмов с открытым ключом, методов обмена ключами и имен сервисов/протоколов.
- ❑ Возможно определение дополнительных алгоритмов, используя имена для них в формате **name@domain**, где *domain* — это полностью обозначенное доменное имя (*Full Qualified Domain Name* — *FQDN*).

SSH использует нумерацию сообщений от 1 до 255, распределенные следующим образом.

- 1—19 Характеристики протокола транспортного уровня (disconnect, ignore, debug и т. п.).
- 20—29 Согласование алгоритмов.
- 30—49 Специфики способов обмена ключами.
- 50—59 Характеристики протокола аутентификации пользователя.
- 60—79 Специфики методов аутентификации пользователя.
- 80—89 Характеристики протокола соединения.
- 90—127 Сообщения, связанные с работой канала.
- 128—191 Зарезервированные для клиентских протоколов.
- 192—255 Локальные расширения (для особенностей реализации).

Рассмотрим некоторые особенности подпротоколов.

Подпротокол транспортного уровня

Напомним, что протокол обеспечивает аутентификацию только сервера, но не клиента, использует зарегистрированный TCP порт 22. Протокол применяет следующие методы (табл. 22.3).

Таблица 22.3. Параметры протокола SSH

Вид метода	Наименование	Описание	Требование поддержки
Сжатие	none	Без сжатия	Обязательно
	zlib	GNU ZLIB (LZ77)	Опционально
Шифрование	3des-cbc	Трехключевой TripleDES в режиме CBC (168 бит)	Обязательно
	blowfish-cbc	Blowfish в режиме CBC (128 бит)	Рекомендовано
	twofish-cbc	Twofish в режиме CBC (256 бит)	Рекомендовано
	arcfour	Arcfour потоковый шифр	Опционально
	idea-cbc	IDEA в режиме CBC	Опционально
	cast128-cbc	CAST-128 в режиме CBC	Опционально
	none	Без шифрования	Не рекомендовано
Контроль целостности	hmac-sha1	HMAC-SHA1 длиной 160 битов	Обязательно
	hmac-sha-96	Первые 96 бит от HMAC-SHA1	Рекомендовано
	hmac-md5	HMAC-MD5 длиной 128 бит	Опционально
	hmac-md5-96	Первые 96 бит от HMAC-MD5	Опционально
	none	Без контроля	Не рекомендовано
Обмен ключами	diffie-hellman-group1-sha1	Метод Диффи-Хеллмана, комбинированный с электронно-цифровой подписью	Обязательно
Открытые ключи	ssh-dss	Обычный DSS (Digital Signature Standard)	Обязательно
	x509v3	Сертификаты X.509	Рекомендовано
	spki	Сертификаты SPKI	Опционально
	pgp	Сертификаты OpenPGP	Опционально

Подпротокол аутентификации

Цель протокола — произвести аутентификацию клиента. Предполагается, что он будет работать поверх протокола транспортного уровня, который уже произвел аутентификацию сервера, установил шифрованное соединение и определил идентификатор сессии.

Для усложнения возможных атак перебора ключей аутентификации, сервер может временно блокировать свою работу после нескольких неудачных попыток аутентификации. Если при работе транспортного уровня была выбрана nereкомендованная опция работы без шифрования, тогда аутентификационные методы, основанные на передаче секретных данных, должны быть заблокированы. Кроме того, если отсутствует контроль целостности, необходимо отключить возможность изменения аутентификационных данных, таких как пароль, так как это может привести к отказу в работе сервиса (атака класса *deny-of-service*), если атакующий изменит шифрованный поток данных.

Сервер управляет аутентификацией, указывая клиенту, какие виды аутентификации необходимы в какой период времени. Клиенту предоставляется возможность выбора из списка методов аутентификации сервера. Таким образом, с одной стороны, сервер контролирует аутентификационные методы, с другой — дает клиенту возможность выбрать поддерживаемый им метод.

Метод *none* (без аутентификации) зарезервирован, но не должен предлагаться сервером в качестве допустимых методов аутентификации. Клиент может предложить этот метод серверу, но сервер должен отказать в нем, если только он не предоставляет возможности работы без аутентификации. В основном цель такого запроса клиента — это получение списка методов аутентификации от сервера. Если после отправления списка методов сервером клиент не отвечает, сервер должен иметь возможность разрыва соединения по таймауту (рекомендуемый период — 10 минут).

Подпротокол соединения

Предполагается, что протокол функционирует поверх транспорта, обеспечивающего безопасность путем шифрования, контроля целостности и аутентификации. Протокол используется для выполнения команд на удаленной машине, а также может быть использован сервером для запуска команд на исполнение на клиентской станции (это может быть запрещено в отдельных реализациях для снижения уровня риска атаки). Протокол подробно описывает форматы различных сообщений о выполнении команд, таких как открытие и закрытие канала, передача данных, продвижение (*forwarding*) и т. п.

Протокол SOCKS, версия 5

Протокол SOCKS5 [RFC-1928] предназначен для прозрачного и безопасного прохождения сложных протоколов прикладного уровня через межсетевой экран типа *посредник приложений* (англ. *application proxy*). Одним из основных требований к безопасности в этом смысле является строгая аутентификация. SOCKS5 завоевал широкую популярность в Интернете из-за простоты своего функционирования в сочетании с широкой универсальностью выполняемых функций. Протокол разработан для обеспечения функционирования клиент/серверных приложений поверх TCP или UDP (в отличие от версии 4) и работает в виде некоей прослойки между прикладным и транспортным уровнями, не предоставляя никаких сервисов сетевого уровня. Протокол различает работу TCP- и UDP-клиентов.

Когда TCP-клиент хочет установить соединение с сервером, который доступен только через межсетевой экран, он должен установить соединение с управляющим портом SOCKS-сервера (номер порта, зарегистрированный IANA, — 1080). В рамках управляющего соединения сервер обрабатывает запрос (рис. 22.23) и производит следующие действия:

- ☐ согласовывает метод аутентификации клиента;
- ☐ аутентифицирует клиента выбранным методом;
- ☐ в случае необходимости производит трансляцию имени удаленного сервера в сетевой адрес (например, DNS-имени в IP-адрес).

Затем SOCKS-сервер устанавливает два соединения вне управляющего соединения для передачи данных.

- ☐ Соединение между собой и удаленным сервером, в рамках которого он от своего имени будет пересылать все запросы клиента и принимать ответы на них; порт SOCKS-сервера в сторону удаленного ресурса выбирается произвольно динамически, и в случае необходимости сообщается клиенту (в рамках управляющего соединения).
- ☐ Соединение между собой и клиентом, в рамках которого клиентом будут передаваться запросы к удаленному серверу и возвращаться SOCKS-сервером ответы на них от удаленного ресурса; порт SOCKS-сервера в сторону клиента выбирается произвольно динамически и всегда сообщается клиенту (в рамках управляющего соединения).

Как и в любой *прокси*-технологии (англ. *proxy* — *посредник*), клиент никогда не производит соединения непосредственно с удаленным сервером. Все данные как в прямом, так и в обратном направлении проходят через SOCKS-сервер.

Еще один важный момент: поскольку аутентичность клиента является одной из наиболее важных функциональных нагрузок протокола SOCKS5, то в нем заложена жесткая привязка к управляющему соединению. Как только случайно или преднамеренно разрывается соединение клиента на управ-

ляющий порт, сразу же SOCKS-сервер перестает транслировать данные как в прямом, так и в обратном направлении во всех "рабочих" соединениях данного клиента. Это касается как TCP, так и UDP-соединений.

VER	NMETHODS	METHODS
1 октет	1 октет	от 1 до 255 октетов

Рис. 22.23. Формат аутентификационного запроса клиента SOCKS-серверу

Поле *VER* указывает номер версии протокола (для текущей версии = 05).

Поле *NMETHODS* указывает количество октетов в поле *METHODS*.

Поле *METHODS* может принимать следующие значения:

00 — не требуется аутентификации;

01 — аутентификация GSS-API [RFC-1961];

03 — аутентификация username/password [RFC-1929];

02–7F — методы аутентификации IANA (Internet Assigned Numbers Authority);

08–FE — собственные реализации методов аутентификации;

FF — нет приемлемых методов аутентификации — возвращается сервером, если клиент не может представить методов аутентификации, допустимых для сервера.

Для обеспечения совместимости реализация протокола обязательно должна поддерживать аутентификацию GSS-API и желательно поддерживать аутентификацию username/password.

После аутентификации, клиент направляет серверу запрос на соединение (рис. 22.24).

VER	CMD	RSV	ATYPE	DST.ADDR	DST.PORT
1 октет	1 октет	00	1 октет	Зависит от ATYP	2 октета

Рис. 22.24. Формат запроса на соединение клиента SOCKS-серверу

Поле *VER* указывает номер версии протокола (для текущей версии = 05).

Поле *CMD* — вид запроса:

Connect — соединение (*CMD* = 01);

Bind — связь (*CMD* = 02);

UDP-Associate — UDP-ассоциация (*CMD* = 03).

Поле *RSV* зарезервировано (= 00).

Поле *ATYP* — тип адреса в поле *DST.ADDR* из следующих значений:

IP-адрес версии 4 (*ATYP* = 01);

доменное имя (*ATYP* = 03);

IP-адрес версии 6 (*ATYP* = 04).

Поле *DST.ADDR* — адрес назначения.

Поле *DST.PORT* — порт назначения.

SOCKS-сервер после обработки клиентского запроса возвращает клиенту следующий ответ (рис. 22.25).

VER	REP	RSV	ATYP	BND.ADDR	BND.PORT
1 октет	1 октет	00	1 октет	Зависит от ATYP	2 октета

Рис. 22.25. Ответ SOCKS-сервера

Поле *VER* указывает номер версии протокола (для текущей версии = 05).

Поле *REP* — код ответа:

00 — успешно;

01 — неклассифицированная ошибка SOCKS-сервера;

02 — соединение не разрешено правилами;

03 — сеть недоступна;

04 — хост недоступен;

05 — в соединении отказано;

06 — TTL (time-to-live) время до удаленного сервера истекло;

07 — команда не поддерживается;

08 — тип адреса не поддерживается;

09 — FF — не назначены.

Поле *RSV* зарезервировано (= 00).

Поле *ATYP* — тип адреса в поле *BND.ADDR* составлен из следующих значений:

IP-адрес версии 4 (*ATYP* = 01);

доменное имя (*ATYP* = 03);

IP-адрес версии 6 (*ATYP* = 04).

Поля *BND.ADDR* и *BND.PORT* представляют собой адрес и порт SOCKS-сервера, на котором он открыл какое-либо соединение (к клиенту или к

серверу), но имеют разную смысловую нагрузку в зависимости от типа запрашиваемого клиентом соединения.

Если в запросе указан тип *Connect*, то происходит *активное подключение* (клиент инициирует сессию к удаленному ресурсу). В этом случае поля *BND.ADDR* и *BND.PORT* содержат адрес и порт, на которых SOCKS-сервер открыл соединение в сторону клиента. В рамках этого соединения SOCKS-сервер будет транслировать все данные, пришедшие от удаленного сервера. При запросе типа *Connect* адрес и порт соединения, открытого SOCKS-сервером в сторону удаленного ресурса, остаются для клиента неизвестными.

Если в запросе указан тип *Bind*, то это означает, что подключение должно было произойти *по пассивной схеме*. Если бы не было межсетевого экрана, то клиент должен был открыть на ожидание сессии произвольный порт, сообщить его номер удаленному серверу, а уже тот — инициировать сеанс на указанный номер порта клиента. Классическим примером протокола, требующего по умолчанию для передачи данных именно такой тип соединения, является FTP. Однако межсетевой экран, если такой существует между клиентом и сервером, не должен позволять удаленному хосту инициировать соединения на произвольный (как это кажется экрану, ведь номер был выделен операционной системой динамически) порт клиента (а для МСЭ соединение покажется именно инициированным извне на "случайный" порт).

Существует два решения этой проблемы. Можно заставить межсетевой экран "читать" команды протоколов прикладного уровня (например, того же FTP), выделять в них сообщения от клиента с указанием номера открытого им порта и временно заносить этот порт в список разрешенных к соединению со стороны удаленного сервера. Данная технология носит наименование *application recognition (AR)* — *распознавание приложений*, у нее есть и несомненные преимущества и недостатки.

Второй метод основан на использовании прокси-серверов прикладного уровня, каковым и является SOCKS-сервер. Рассмотрим, как происходит соединение.

1. Клиент отправляет SOCKS-серверу запрос типа *Bind*.
2. SOCKS-сервер открывает в сторону удаленного ресурса динамически произвольный порт в режиме ожидания соединения и сообщает клиенту в рамках управляющей сессии в первом своем ответе адрес и порт открытого соединения (поля *BND.ADDR* и *BND.PORT*).
3. Клиент формирует запрос прикладного уровня, в котором требовалось указать порт пассивного соединения, но подставляет туда не свой адрес, а присланные от SOCKS-сервера параметры.
4. Как только в адрес открытого SOCKS-сервером в режиме ожидания порта приходит первый пакет от удаленного ресурса (запрос на инициирование соединения), SOCKS-сервер высылает в рамках управляющей

сессии второй ответ клиенту, указывая в полях *BND.ADDR* и *BND.PORT* адрес и порт, открытые им для этого потока уже в сторону клиента.

5. После установления всех соединений SOCKS-сервер начинает прозрачно транслировать данные в обоих направлениях подобно тому, как если бы соединение устанавливалось по типу *Connect*.

Для запроса *UDP-Associate*, используемого для возможности пересылки UDP-датаграмм через межсетевой экран, в полях *BND.ADDR* и *BND.PORT* ответа SOCKS-сервер указывает свои адрес и порт, куда клиент обязан посылать UDP-запросы, которые должны пересылаться удаленному ресурсу. Напомним, что ассоциация сразу же прекращается с разрывом управляющего TCP-соединения.

Клиент должен направлять свои датаграммы (UDP-relay) SOCKS-серверу, производя *инкапсуляцию* (сокрытие значимых данных) датаграммы, если это требуется выбранным методом аутентификации. Каждая UDP-датаграмма должна быть снабжена следующим дополнительным заголовком (рис. 22.26).

RSV	FRAG	ATYPE	DST.ADDR	DST.PORT	DATA
2 октета	1 октет	1 октет	Зависит от ATYPE	2 октета	Данные

Рис. 22.26. UDP-запрос

Поле *RSV* зарезервировано (= 0000).

Поле *FRAG* — номер текущего фрагмента, если данные фрагментированы.

Поле *ATYPE* — тип адреса в поле *DST.ADDR* состоит из следующих значений:

- ☐ IP-адрес версии 4 (*ATYPE* = 01);
- ☐ доменное имя (*ATYPE* = 03);
- ☐ IP-адрес версии 6 (*ATYPE* = 04).

Поле *DST.ADDR* — требуемый адрес назначения.

Поле *DST.PORT* — требуемый порт назначения.

Поле *DATA* — данные пользователя.

Несколько слов о том, зачем было необходимо включать в ответ SOCKS-сервера поле *BND.ADDR*, и использовать именно его, а не первоначальный адрес управляющей сессии, при открытии клиентом вторичных соединений. Дело в том, что при разработке протокол SOCKS5 позиционировался как схема, позволяющая создавать целый кластер прокси-серверов, работающих под управлением мастер-сервера. В этом случае мастер-сервер поддерживает только управляющие сессии со всеми подключенными к кластеру клиентами, а всю нагрузку на собственно трансляцию потоков данных (то есть вторичные соединения) несут распределенно несколько "рабочих" серверов.

При обработке каждого очередного запроса от клиента мастер-сервер по какому-либо правилу выбирает один "рабочий" сервер и обменивается с ним всей необходимой информацией о запросе. Затем в ответе клиенту в поля *BND.ADDR* и *BND.PORT* подставляются уже координаты "рабочего" сервера трансляции.

В заключение необходимо добавить, что существуют отдельные документы, описывающие способы аутентификации SOCKS. Для метода GSS-API — это [RFC-1961], для метода username/password — [RFC-1929].

Протокол IP-Security

Обзор компонентов протокола

Описанию протокола IP-Security (*IPSec*) или, если быть точнее, набора протоколов (англ. *protocol suite*) можно посвятить отдельную книгу. Достаточно перечислить документы, описывающие его функционирование RFC с 2401 по 2412 (рис. 22.27).

- ☐ Архитектура безопасности для IP.
- ☐ Аутентификационный заголовок (*Authentication Header — AH*) для IP.
- ☐ Использование HMAC-MD5-96 в ESP и AH.
- ☐ Использование HMAC-SHA-1-96 в ESP и AH.
- ☐ Шифрующий алгоритм DES-CBC для ESP.
- ☐ Безопасное сокрытие существенных данных (*Encapsulating Security Payload — ESP*) для IP.
- ☐ Безопасность IP-домена интерпретаций для ISAKMP.
- ☐ ISAKMP (*Internet Security Association and Key Management Protocol*) — Протокол ассоциаций (наборов параметров) безопасности и управления ключами в Интернете.
- ☐ Обмен ключами в Интернете (*Internet Key Exchange — IKE*).
- ☐ Нулевой (NULL) алгоритм шифрования и его использование с IPSec.
- ☐ Общее описание (roadmap document) безопасности IP.
- ☐ Протокол определения ключей Oakley.

Основополагающими понятиями IPSec являются:

- ☐ аутентификационный заголовок (AH);
- ☐ безопасное сокрытие данных (ESP);
- ☐ варианты работы: туннельный и транспортный;
- ☐ ассоциации безопасности;
- ☐ управление ключами.

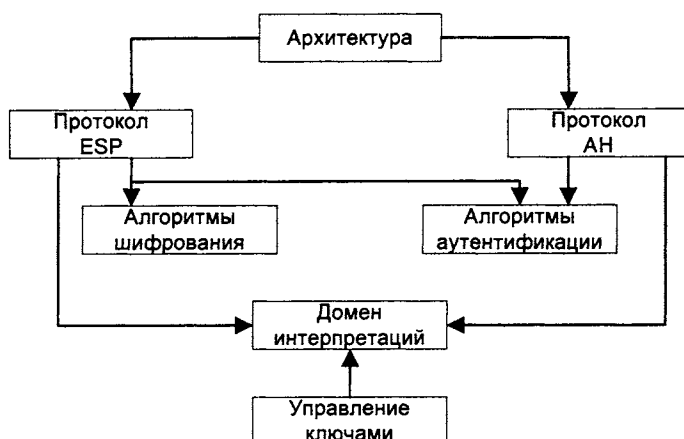


Рис. 22.27. Взаимосвязь отдельных направлений и частей IPsec

Аутентификационный заголовок

Аутентификационный заголовок (англ. *authentication header*) АН (рис. 22.28) предназначен для обеспечения аутентификации отправителя, контроля целостности данных и опционально для предотвращения *повторной отправки* (англ. *replay*) пакета — при условии, что принимающая сторона настроена производить проверку последовательного номера пакета. Дополнительно необходимо заметить, что поля *IP*-пакета, которые изменяются в пути следования, не подлежат контролю целостности.

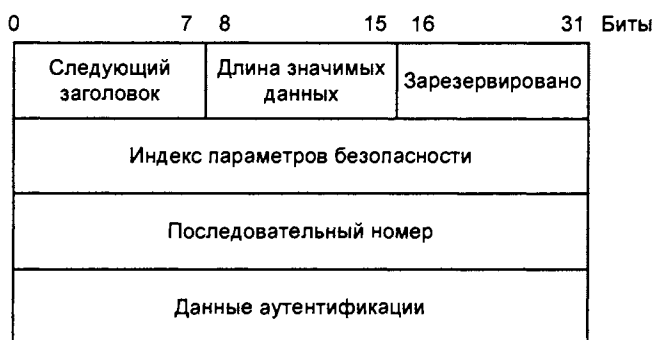


Рис. 22.28. Формат АН

- ❑ Поле *Следующий заголовок* (*Next header*) идентифицирует тип следующих значимых данных за аутентификационным заголовком. Значения поля берутся из предопределенного множества установленных IANA (Internet Assigned Numbers Authority) номеров.

- ❑ Поле *Длина значимых данных (Payload length)* определяет длину самого АН в 32-битовых словах минус 2 слова, поскольку для заголовков, расширяемых для IPv6, установлено требование сокращения длины заголовка на 64 бита.
- ❑ *Зарезервированное* поле должно быть нулевым.
- ❑ Поле *Индекс параметров безопасности (Security Parameters Index — SPI)* — это значение, которое в совокупности с адресом назначения и самим протоколом (в данном случае АН) однозначно определяет Ассоциацию безопасности (Security Association — SA) для данной датаграммы в виде 32-битного номера. Номера с 1 по 255 зарезервированы IANA; SPI, равный нулю, означает, что SA не установлена. *Ассоциация безопасности* — это набор параметров (версия алгоритмов шифрования и аутентификации, схема обмена ключами и т. п.), определяющих, каким образом будет обеспечиваться защита данных.
- ❑ Поле *Последовательный номер (Sequence number)* — это монотонно возрастающий от 0 (при установлении SA) номер пакета. Он используется для возможности контроля получателем ситуации повторной пересылки пакетов.
- ❑ Поле *Данные аутентификации (Authentication data)* содержат *Значение контроля целостности (Integrity check value — ICV)*, рассчитанное по всем данным, которые не изменяются в пути следования пакета или предсказуемы на момент достижения им получателя. Значение ICV рассчитывается в зависимости от алгоритма, определенного в SA, например Код аутентификации сообщения (Message Authentication Code — MAC) с ключом симметричного или асимметричного алгоритма или хэш-функции.

Безопасное сокрытие существенных данных

Отличительной функцией протокола *Encapsulating Security Payload (ESP)* — *Безопасное Соккрытие Данных* — является обеспечение конфиденциальности путем шифрования части данных, представляющих информацию протоколов верхнего уровня (рис. 22.29). Дополнительной функцией может быть названо сокрытие реального объема сетевого трафика. ESP обеспечивает аутентификацию и контроль целостности, как и АН, и, как следствие, может быть использован как отдельно, так и вместе с АН (разделяя или совмещая функции). Кроме того, отдельно выделяется понятие *уплотненный режим* (англ. *nested fashion*) путем использования туннельного варианта, речь о котором пойдет в разд. "Туннельный режим работы" данной главы. Различают варианты использования протокола *хост-хост*, *шлюз-шлюз* и *хост-шлюз*. Соответственно механизмы защиты применяются сразу при выходе данных с хоста, их породившего, либо при покидании границ защищенной сети (в варианте шлюза).

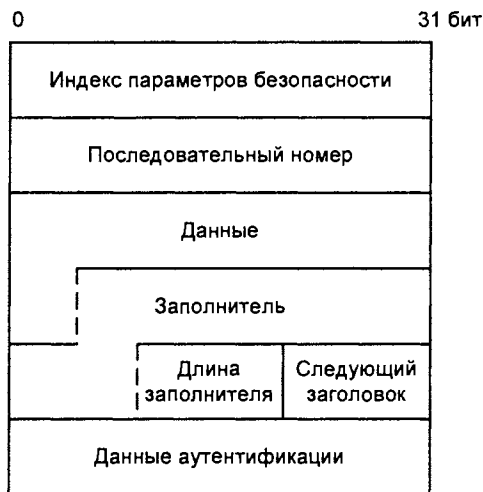


Рис. 22.29. Формат ESP

- ❑ Поле *Индекс параметров безопасности* (*Security Parameters Index — SPI*) значение, которое совместно с адресом назначения и самим протоколом определяет Ассоциацию безопасности (*Security Association — SA*) для данной датаграммы в виде 32-битного числа. (0 означает, что SA не установлена.)
- ❑ Поле *Последовательный номер* (*Sequence number*) — возрастающий от 0 (при установлении SA) номер пакета. Используется для возможности контроля получателем ситуаций повторной пересылки пакетов.

Эти два поля носят специальное название *Заголовок ESP (ESP header)*.

- ❑ Поле *Существенные данные* (*Payload data*) — поле переменной длины, содержащее данные, тип которых указан в поле *Следующий заголовок*. Дополнительно здесь может содержаться информация, необходимая для функционирования алгоритмов шифрования сокрытия данных. Способы размещения такой информации описываются в отдельных документах.
- ❑ Поле *Заполнитель* (*Padding*) — поле переменной длины, служащее следующим основным целям. Во-первых, его задача дополнить данные до длины блока, требуемого для применения криптографического алгоритма. Во-вторых, скрыть истинный размер существенных данных.
- ❑ Поле *Длина заполнителя* (*Pad length*) — указывает размер заполнителя в байтах от 0 до 255, с тем чтобы его можно было отделить от существенных данных.
- ❑ Поле *Следующий заголовок* (*Next header*) — 8-битовое поле, указывающее тип данных в поле *Данные*. Возможные значения поля установлены IANA (Internet Assigned Numbers Authority).

Последние три поля: *Заполнитель*, *Длина заполнителя* и *Следующий заголовок* носят специальное название *Хвост ESP (ESP trailer)*.

- ❑ Поле *Данные аутентификации (Authentication data)* — поле переменной длины, содержит *Значение контроля целостности (Integrity check value — ICV)*, рассчитанные по содержимому ESP пакета за минусом самих данных аутентификации.

Каждый из протоколов АН и ESP поддерживает два модуля работы — транспортный и туннельный.

Транспортный режим работы

В этом варианте механизмы безопасности применяются только для протоколов, начиная с транспортного (TCP) уровня и выше, оставляя данные самого сетевого уровня (заголовок IP) без дополнительной защиты. Места размещения дополнительной информации, вставляемой протоколами в пакет, представлены на рис. 22.30.

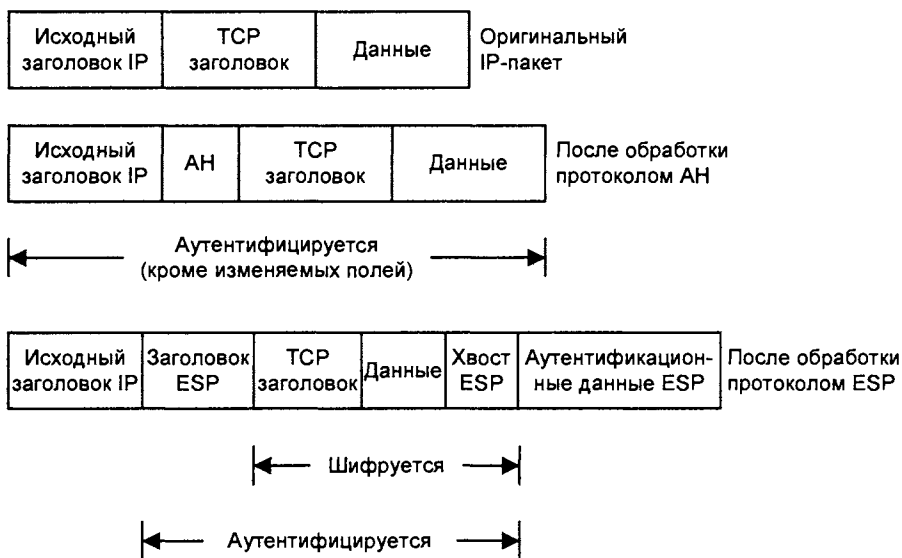


Рис. 22.30. Транспортный режим

Туннельный режим работы

Этот режим интересен тем, что обеспечивает защиту также и данных сетевого уровня путем добавления нового IP-заголовка. После определения ассоциаций безопасности (например, между двумя шлюзами) истинные адреса хостов отправления и назначения (и другие служебные поля) полностью защищаются от модификаций для АН или вообще скрываются для ESP, а в

новый заголовок выставляются адреса и другие данные для шлюзов (отправления/получения). На рис. 22.31 видны преимущества и недостатки обоих протоколов. ESP обеспечивает сокрытие данных, но не полную аутентификацию всего пакета. АН полностью аутентифицирует, но не скрывает данные. В этом причина того, что для обеспечения высокого уровня безопасности, применение протоколов совмещается.

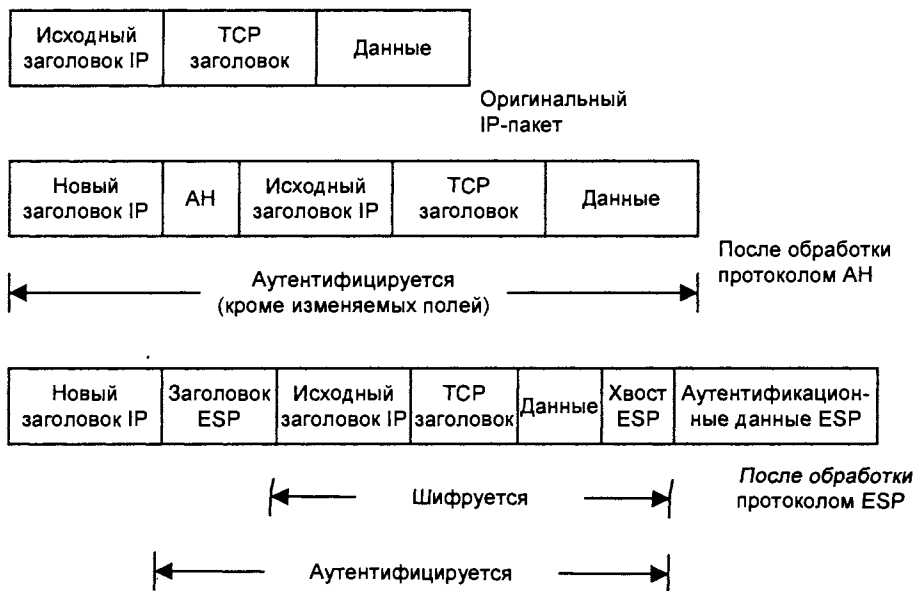


Рис. 22.31. Туннельный режим

Ассоциации безопасности

Концепция *Ассоциаций безопасности* (англ. *Security Association — SA*) является основополагающей для IPSec. SA определяет взаимоотношения между двумя или более участниками безопасной связи и описывает, какие сервисы будут использованы для обеспечения безопасности: алгоритмы шифрования, алгоритмы аутентификации и общие сессионные ключи.

SA всегда *однонаправленна* (англ. *unidirectional*), поэтому для одного двунаправленного соединения между двумя участниками требуется две SA (по одной на каждое направление). Также для АН и ESP должны быть использованы различные SA, если применяются оба протокола, по две SA должны присутствовать для каждого направления.

Набор всех SA, установленных на узле для связи с другим узлом (узлами), хранится в специальной *Базе данных ассоциаций безопасности* (*Security Association Database — SAD*). Каждый узел поддерживает две SAD: одну для

входящего и одну для исходящего трафика. В зависимости от реализации может потребоваться несколько пар SAD для мультиинтерфейсных узлов — по одной SAD на каждый интерфейс.

Когда у узла существует несколько соединений с другим узлом (узлами), необходимо определить, которую из SA применять к пакетам какого соединения. Для этой цели служит *База данных политик безопасности (Security Policy Database — SPD)*. Запись SPD состоит из полей, связывающих SA с идентификатором пакетов соединения — *селектором* (англ. *selector*). Селектор состоит из следующих параметров.

- ☐ IP-адрес назначения (индивидуальный, групповой, широковещательный), диапазон адресов + маска или групповой символ (wildcard).
- ☐ IP-адрес источника (индивидуальный, групповой, широковещательный), диапазон адресов + маска или групповой символ (wildcard).
- ☐ Имя в двух вариантах:
 - Идентификатор пользователя в формате DNS (fully qualified user name) string или X.500 distinguished name;
 - Наименование системы в формате DNS (fully qualified domain name), X.500 distinguished name или X.500 general name;
- ☐ Уровень чувствительности данных по меткам IPSO/CIPSO (такие как "не определено", "коммерческая собственность", "секретно" и т. п.) — этот параметр опционален.
- ☐ Протокол транспортного уровня.
- ☐ Порт назначения.
- ☐ Порт источника.

Протокол обмена ключами — IKE

Поскольку основным механизмом обеспечения безопасности данных протокола являются криптографические методы, участники защищенного соединения должны наладить обмен соответствующими криптографическими ключами. Обеспечить настройку процесса такого обмена можно вручную и автоматически. Первый способ допустим для небольшого количества достаточно статичных систем, а в общем случае это производится автоматически.

Для автоматического обмена ключами по умолчанию используется *Протокол управления ключами в Интернете (Internet Key Management Protocol — IKMP)*, иначе называемый *Обмен ключами в Интернете (Internet Key Exchange — IKE)*. Дополнительно или альтернативно могут быть применены другие протоколы, такие как Kerberos или SKIP.

IKE совмещает в себе три основных направления (отдельных протокола).

- ☐ *ISAKMP (Internet Security Association and Key Management Protocol)* — Протокол ассоциаций безопасности и управления ключами в Интернете.

Это общее описание (framework) для обеспечения аутентификации и обмена ключей без указания конкретных прикладных алгоритмов.

- ❑ *Oakley (Oakley key determination protocol)* — Протокол определения ключей Окли. Он описывает последовательности обмена ключами — моды (mode) и описывает предоставляемые ими функции.
- ❑ *SKEMI (Secure Key Exchange Mechanism for Internet)* — Механизм безопасного обмена ключами в Интернете. Он описывает многофункциональные технологии, предоставляющие анонимность, неотрекаемость (апеллируемость) и быстрое обновление ключей.

В ходе установления Ассоциации безопасности, IKE согласовывает следующие атрибуты: алгоритм шифрования, алгоритм хэширования, метод аутентификации и данные о группе преобразования алгоритма Диффи-Хеллмана.

Аутентификация может быть произведена с помощью следующих методов:

- ❑ предопределенного секрета (секретная информация, известная обеим сторонам соединения до начала установки соединения);
- ❑ криптографии с открытым ключом (обе стороны генерируют случайное число и шифруют его вместе со своим идентификатором открытым ключом противоположной стороны);
- ❑ электронной цифровой подписью (стороной используется закрытый ключ, открытый ключ которого есть у противоположной стороны).

Изучая IKE, придется столкнуться с понятиями *фаз* (англ. *phase*) и *режимов* (англ. *mode*). Кратко опишем их. IKE содержит две фазы согласования ключей. В первой фазе происходит создание защищенного канала, во второй — согласование и обмен ключами, установление SA. Первая фаза использует один из двух режимов: *Основной* (англ. *Main Mode*) или *Агрессивный* (англ. *Aggressive Mode*). Различие между ними в уровне защищенности и скорости работы. Основной режим, более медленный, защищает всю информацию, передаваемую между узлами. Агрессивный режим для ускорения работы оставляет ряд параметров открытыми и уязвимыми для прослушивания, его рекомендуется использовать только в случае, когда критическим вопросом является скорость работы. Во второй фазе используется *Быстрый режим* (англ. *Quick Mode*), названный так потому, что не производит аутентификации узлов, считая, что это было сделано в первой фазе. Эта фаза обеспечивает обмен ключами, с помощью которых происходит шифрование данных.

Из существенных понятий остался упомянутый, но не определенный *Домен интерпретаций* (*Domain of Interpretation — DOI*). DOI — это понятие в рамках протокола ISAKMP, которое используется для группировки протоколов, использующих данный протокол для согласования с SA. Эти протоколы, в рамках своего DOI, используют единое пространство имен для протоколов и механизмов безопасности, единые идентификаторы протоколов обмена ключами и ряд других параметров.

Simple Key Management for Internet Protocol

Другим вариантом протокола обеспечения безопасности на сетевом уровне является протокол Simple Key Management for Internet Protocol (*SKIP*) — *Простой протокол управления ключами в Интернете* [Aziz]. Этот протокол развивается в основном усилиями компании Sun Microsystems. Отличительной его особенностью является исключительное использование в качестве криптографического алгоритма метода Диффи-Хеллмана. С криптографической точки зрения алгоритм подробно описан в *главе 20*, рассмотрим только основные его положения (рис. 22.32).

Для каждого субъекта I его парой ключей будут являться некоторое число i (закрытый ключ) и $g^i \bmod(p)$ (открытый ключ). Для каждого субъекта J его парой ключей будут являться j (закрытый ключ) и $g^j \bmod(p)$ (открытый ключ), где p и g — определенные, известные всем желающим, параметры. Тогда их совместным распределяемым аутентифицирующим секретом будет $K_{ij} = g^{ij} \bmod p$. Подчеркивается, что это *долговременный секрет*, так как в общем случае его не требуется менять для каждой коммуникационной сессии.

Поскольку K_{ij} — долговременный ключ, использование его для защиты самих данных небезопасно, так как дает возможность накопления материала для криптоанализа. Поэтому для шифрования самих данных используется отдельный *сессионный ключ* K_p , а долговременный K_{ij} используется только для шифрования сессионного ключа. Учитывая, что долговременный ключ при расчетах можно кэшировать, а само шифрование симметричное, т. е. скорость может быть достаточно высока, сессионный ключ можно генерировать вообще для каждого пакета. Аутентификация в данном случае обеспечивается предположением, что если пакет успешно дешифрован, значит, он был зашифрован именно тем, кто знает секретный ключ отправителя, т. е. самим отправителем.

Открыт	Зашифрован K_{ij}	Зашифрован K_p
Заголовок IP	Сессионный ключ K_p	Исходный IP-пакет

Рис. 22.32. Шифрованный SKIP-пакет

В качестве дополнительной меры обеспечения защиты используется формирование ключа для шифрования не в чистом виде K_{ij} , а производная $K_{ijn} = h(K_{ij}, n)$, где $h()$ — хэш-функция, а n — постоянно увеличивающийся счетчик. Этот счетчик может изменяться на единицу в зависимости от конкретной задачи раз в день, час, минуту и т. д. Таким образом, обеспечивается дополнительная защита от возможного повторения посылки пакета, скопированного злоумышленником в предыдущем такте n .

Отдельное внимание в документации по SKIP посвящено проблемам широковещательного (англ. *multicast*) трафика. Использование в таком случае единого ключа шифрования данных из общего *Центра Распределения Ключей (ЦРК) (Key Distribution Center — KDC)* порождает две основные проблемы:

- ❑ затруднение с оперативной заменой ключа в сетях с большим количеством узлов и высокими скоростями (последнее — по причине того, что период смены ключей зависит не только от времени использования, но и от объемов шифруемого трафика, т. е. в сетях с более интенсивным защищенным трафиком ключи следует менять чаще);
- ❑ поскольку один и тот же ключ используется для защиты большого объема данных от разных узлов, это дает больше материала для криптоанализа и, таким образом, представляет угрозу для безопасности.

Решение этих проблем уже заложено в принципе защиты протокола SKIP. ЦРК достаточно своевременно распределять не сам ключ шифрования, а ключ шифрования ключей. На рис. 22.32 его функцию выполняет ключ K_{ij} , однако в групповом трафике он имеет специальное название *Ключ Группового Обмена (Group Interchange Key — GIK)*. Теперь каждый узел может сгенерировать свой собственный сессионный ключ и отправлять групповые сообщения, шифруя им данные. Узлу необходимо только обеспечить наличие в пакете данного сессионного ключа, зашифрованного GIK.

Layer 2 Forwarding, Point-to-Point Tunneling Protocol, Layer 2 Tunneling Protocol

Описание этих трех протоколов второго (канального) уровня традиционно объединяется из-за их функционального сходства. Layer 2 Forwarding (L2F) — это продукт компании Cisco Systems, Inc. [RFC-2341], Point-to-Point Tunneling Protocol (PPTP) — компании Microsoft [RFC-2637], а Layer 2 Tunneling Protocol (L2TP) — это совместная разработка Cisco Systems, Inc. и Microsoft в рамках рабочей группы IETF [RFC-2661]. Совместная разработка была предпринята как раз по причине сходной функциональности для развития преимуществ первых двух протоколов.

Основной отличительной чертой этих трех протоколов является их способность работать по телефонным (dial-up) линиям поверх протоколов PPP (Point-to-Point Protocol), SLIP (Serial Line Internet Protocol) и предоставлять возможность независимой работы протоколам третьего (сетевого) уровня — IP, IPX, AppleTalk. Поскольку речь идет все же о телефонных линиях, сети, основанные на использовании этих протоколов, называют *VPDN (Virtual Private Dial-up Network — Виртуальная частная сеть с коммутируемым доступом)*, в отличие от интернет-ориентированных частных сетей *VPN (Virtual Private Network)*.

Описывая L2F, разработчики из компании Cisco Systems, Inc. выделяют следующие три протокола, или группы протоколов, необходимых для создания туннелей в защищенной сети:

- ❑ *пассажирский протокол* (англ. *passenger protocol*) — это тот протокол, который будет инкапсулирован (PPP, SLIP и т. п.);
- ❑ *инкапсулирующий протокол* (англ. *encapsulating protocol*) — это тот протокол, который отвечает за создание, поддержание и снятие туннеля (в нашем случае — L2F);
- ❑ *несущий протокол* (англ. *carrier protocol*) — используется для транспортировки инкапсулированного протокола (чаще всего это IP), транспортный протокол для L2F — это UDP, порт 1701.

Типовой сценарий построения туннеля следующий:

1. Удаленный пользователь устанавливает соединение по телефонной линии (например, PPP) к *серверу сетевого доступа* (*Network Access Server — NAS*) провайдера, предоставляющего услуги несущего протокола (например, интернет-провайдер с IP).
2. NAS производит идентификацию/аутентификацию пользователя на основе данных самого провайдера. Для уменьшения размера базы данных по пользователям у провайдера можно, если это допускается корпоративной политикой организации пользователя, производить идентификацию на основе не индивидуального имени пользователя, а общекорпоративного, доменного имени.
3. NAS инициирует создание туннеля (L2F) к шлюзу доступа (gateway) сети, к которой хочет получить доступ удаленный пользователь.
4. Шлюз анализирует аутентификационные данные пользователя на основе собственной базы безопасности и допускает или не допускает создание туннеля (в нашем примере — допускает).
5. NAS регистрирует создание туннеля и, при необходимости, объем передаваемого трафика.
6. Шлюз производит необходимый обмен информацией с удаленным пользователем в рамках PPP (например, присваивает IP-адрес).
7. Устанавливается туннель между удаленным пользователем и шлюзом.

Следует понимать, что здесь "туннель" используется для обозначения того, что протокол PPP функционирует прозрачно от удаленного пользователя до шлюза, как если бы пользователь устанавливал соединение напрямую со шлюзом, минуя NAS. При этом данные не обязательно защищены криптографически на всем пути от пользователя до шлюза, туннель только предоставляет возможность применения всех механизмов аутентификации, авторизации и учета, поддерживаемых PPP.

Говоря о PPTP, используют понятия *Концентратор доступа PPTP* (*PPTP Access Concentrator — PAC*) и *Сетевой сервер PPTP* (*PPTP Network Server — PNS*), которые образуют оконечные точки туннеля. При этом функционально разделяют обязанности NAS:

- ☐ по обеспечению физического интерфейса к телефонной линии;
- ☐ логическое окончание (termination) сессии Протокола управления связью (LCP — Link Control Protocol) и Протокола управления сетью (NCP — Network Control Protocol) в PPP;
- ☐ аутентификации средствами PPP;
- ☐ агрегирование и связывание каналов в Мультиканальном протоколе PPP;
- ☐ мультипротокольную маршрутизацию;
- ☐ переключение (англ. *bridging*) между физическими интерфейсами.

Таким образом, PAC присоединяется к телефонной линии, а PNS выполняет серверную часть.

В PPTP-соединении существуют две основных составляющих: *Управляющее соединение* (англ. *Control Connection*) — обычное TCP-соединение по порту 1723; и IP-туннель между PAC и PNS, который инкапсулирует PPP-пакеты по протоколу *GRE* (*General Routing Encapsulation — Общая маршрутная инкапсуляция*). Управляющее соединение отвечает за установление, управление и прекращение туннелируемой сессии, по нему PNS информируется о входящем вызове, а PAC уведомляется о необходимости сделать исходящий вызов. Туннель обеспечивает перенос PPP-пакетов, принадлежащих данной сессии, мультиплексирование различных сессий в одном туннеле, а также включает методы управления коллизиями и контроля ошибок.

Коснемся немного протокола GRE по последней доступной спецификации [RFC-2784]. Он предназначен для инкапсуляции некоего протокола сетевого уровня по другому протоколу сетевого уровня.

В общем виде инкапсулированный пакет выглядит следующим образом:

Заголовок несущего протокола | GRE-заголовок | Инкапсулированный пакет, где GRE-заголовок имеет следующий формат (рис. 22.33).

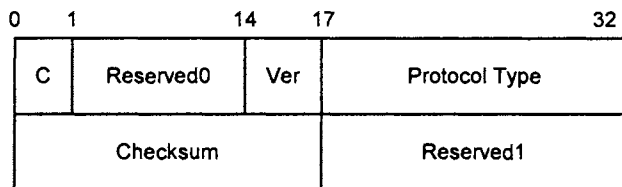


Рис. 22.33. Заголовок GRE

Поля *Checksum* и *Reserved1* опциональны и несут смысловую нагрузку, только если бит *C* выставлен равным 1.

Поле *Reserved0* — биты с 1 по 5 оставлены для обеспечения совместимости с GRE из RFC-1701, если совместимость не требуется, они должны быть равны 0, биты 6-12 оставлены для будущего использования.

Поле *Ver* — номер версии должен быть равен 0.

Поле *Protocol Type* указывает тип инкапсулированного протокола по RFC-1700.

Поле *Checksum* содержит контрольную сумму для всего заголовка.

Поле *Reserved1* оставлено для будущего использования.

Поскольку L2TP является совместным творчеством компаний Cisco Systems, Inc. и Microsoft, предназначенным для объединения всего лучшего, что было в двух исходных протоколах, будем надеяться, что он является наиболее перспективным и остановимся на нем подробнее.

Из PPTP взято понятие двух оконечных точек туннеля — *Концентратора Доступа* и *Сетевого Сервера*, терминология взята применительно к L2TP — *LAC (L2TP Access Concentrator)* и *LNS (L2TP Network Server)*. Из L2F взято использование UDP с номером порта 1701 в качестве транспорта по IP.

L2TP (рис. 22.34) использует два типа сообщений:

1. Управляющие сообщения для установления, поддержания и снятия туннеля и вызовов.
2. Сообщения данных для инкапсуляции PPP пакетов в туннель.

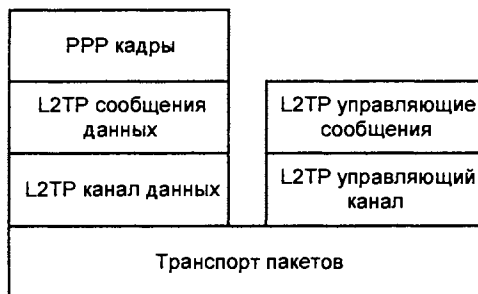


Рис. 22.34. Структура протокола L2TP

В качестве транспорта пакетов используется UDP для IP, но также возможно использование ATM, Frame Relay и т. п.

Канал данных описывается в общем случае как канал ненадежной доставки (потерянные сообщения данных не передаются заново). Управляющий канал описывается как надежный за счет последовательной нумерации пакетов и повторной передачи пропущенных. Опционально такую нумерацию возможно поддерживать и для канала данных.

Заголовок L2TP пакета имеет следующий формат (рис. 22.35).

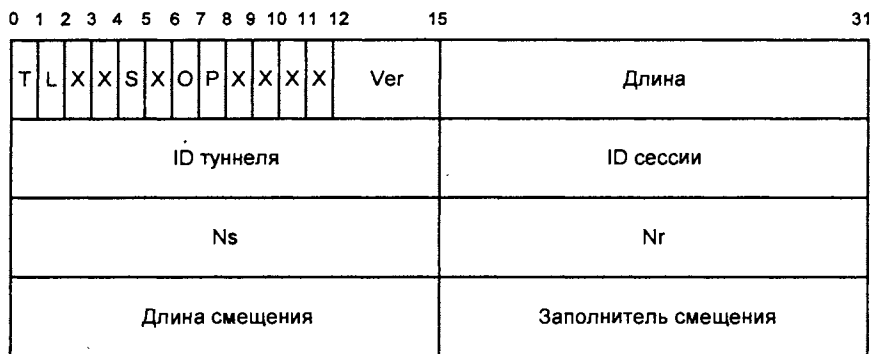


Рис. 22.35. Заголовок L2TP-пакета

Поле *T (Type)* — тип, указывает тип сообщения — 0 для данных и 1 для управления.

Поле *L (Length)* — длина, указывает на наличие поля *Длины*: если $L = 1$, *Длина* присутствует. Значение 1 обязательно для управляющих сообщений.

Поля *X* оставлены для будущего использования. Должны быть равны 0 для исходящих сообщений и проигнорированы для входящих.

Поле *S (Sequence)* — *Последовательность*, равное 1, указывает, что присутствуют поля *Ns* и *Nr*. Значение 1 обязательно для управляющих сообщений.

Поле *O (Offset)* — *Смещение*, равное 1, указывает, что присутствует поле *Длина смещения*. Значение 0 обязательно для управляющих сообщений.

Поле *P (Priority)* — *Приоритет*, равное 1, указывает, что эти данные имеют преимущество в обработке в очередях и передаче. Используется, например, в сообщениях, предназначенных для поддержки актуальности связи (*LCP keeplive*). Значение 0 обязательно для управляющих сообщений.

Поле *Ver* — *Версия*, должно быть равно 2 для текущей версии протокола [RFC-2661], 1 определяет протокол L2F [RFC-2341].

Поле *Длина* указывает длину всего сообщения в октетах.

Поле *ID туннеля* — это идентификатор для Управляющего соединения, он имеет только локальное значение и может быть различным для одного туннеля на обоих концах.

Поле *ID сессии* — это идентификатор конкретной сессии в данном туннеле, он имеет только локальное значение и может быть различным для одного туннеля на обоих концах.

Поле *Ns* — это последовательный номер сообщения данных или управляющего сообщения от 0 до 2^{16} .

Поле *Nr* — это следующий ожидаемый номер для управляющего сообщения (для сообщения данных он игнорируется), равен $(Ns + 1)$.

Поле *Длина смещения* указывает количество октетов после заголовка до начала собственно данных.

Поле *Заполнитель смещения* — это собственно октеты от конца заголовка до начала данных.

В описании протокола L2TP вводится новое для этого семейства понятие, которому посвящена значительная часть [RFC-2661]. Это — пара *Атрибут/Значение* (*Attribute/Value Pair*) — соединение некоего параметра с его значением. AVP — единый метод кодировки типов и содержимого сообщений в рамках протокола. Сами управляющие сообщения представлены следующим набором:

Сообщения управления:

0 Зарезервировано

1 SCCRQ (Start-Control-Connection-Request) — Запрос начала Управляющего соединения

2 SCCRP (Start-Control-Connection-Reply) — Ответ на начало Управляющего соединения

3 SCCCEN (Start-Control-Connection-Connected) — Подтверждение начала Управляющего соединения

4 StopCCN (Stop-Control-Connection-Notification) — Сообщение о прекращении Управляющего соединения

5 Зарезервировано

6 (HELLO) Hello — Подтверждение актуальности туннеля

Управление вызовами (англ. *call*):

7 (OCRQ) *Outgoing-Call-Request* — Запрос исходящего вызова

8 (OCP) *Outgoing-Call-Reply* — Ответ на запрос исходящего вызова

9 (OCCN) *Outgoing-Call-Connected* — Подтверждение исходящего вызова

10 (ICRQ) *Incoming-Call-Request* — Запрос входящего вызова

11 (ICP) *Incoming-Call-Reply* — Ответ на запрос входящего вызова

12 (ICCN) *Incoming-Call-Connected* — Подтверждение входящего вызова

13 Зарезервировано

14 (CDN) *Call-Disconnect-Notify* — Сообщение о прекращении вызова

Сообщение об ошибках:

15 (WEN) *WAN-Error-Notify* — Сообщение об ошибке в сети *WAN*

Управление сессией PPP:

16 (SLI) *Set-Link Info* — Информация о соединении

На рис. 22.36 приведена общая схема функционирования протокола L2TP.

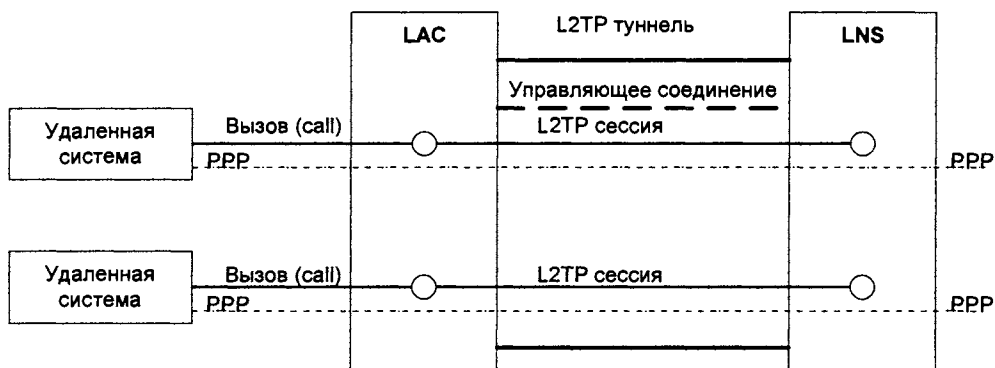


Рис. 22.36. Туннелирование PPP по протоколу L2TP

Инфраструктура открытых ключей

Для того чтобы два субъекта могли безопасно обмениваться информацией по незащищенным каналам связи, используются различные методы и алгоритмы криптографии, в том числе и алгоритмы с т. н. *открытыми ключами*. Теоретические основы этой технологии изложены в *главе 18*. Общий смысл подобных алгоритмов состоит в том, что каждый субъект создает и хранит пару связанных ключей шифрования — то, что зашифровано с помощью одного из них, может быть дешифровано только с помощью другого.

Предполагая в этой ситуации, что ключ шифрования (закрытый, или приватный ключ) сохраняется субъектом в строгом секрете, а ключ дешифрования (открытый, или публичный) распространен между остальными участниками информационного обмена, можно решать различные вопросы безопасности. Например, при помощи *электронно-цифровой подписи* (ЭЦП) данных субъекта на основе закрытого ключа можно обеспечить аутентификацию отправителя (только тот, кто имеет приватный ключ для данной ЭЦП, мог ее установить под присланным текстом) и необходимые условия для апеллируемости данных (только тот, кто имеет приватный ключ, мог создать текущий набор данных), если будут использованы дополнительные механизмы, в том числе организационного характера (например, контрактное закрепление обязательств по сохранению в тайне закрытого ключа или создание центра сертификации открытых ключей).

При этом вся схема безопасности работает только при одном важном условии — если данный участник обмена информацией обладает действительно тем самым открытым ключом, секретный ключ которого находится только у исходного субъекта. Это означает:

- что секретный ключ, соответствующий данному открытому, не должен быть скомпрометирован;

- открытый ключ, соответствующий данному секретному, доставлен участнику процесса информационного обмена надежным способом, т. е. не был подменен на ключ из другой пары, принадлежащей злоумышленнику.

Именно для решения второй задачи и разворачивается Public Key Infrastructure (PKI) — *инфраструктура открытых ключей*. Зачастую участники информационного обмена находятся от субъекта на другом конце земного шара, а их количество и время работы с субъектом таково, что невозможно отправить к каждому из них своего курьера с запечатанным конвертом, в котором находится дискета с открытым ключом субъекта.

В системе PKI вводятся два важных понятия — *цифровой сертификат* и *сертифицирующий уполномоченный* (англ. *Certification Authority — CA*). CA — это организация, которой априори доверяют все участники информационного взаимодействия и которая производит формирование цифровых сертификатов, обеспечивающих уверенность, что данный открытый ключ действительно принадлежит заявленному субъекту. Наиболее популярные известные во всем мире CA — это VeriSign, EnTrust и др. Конкретная организация сама может выступать CA по отношению к своим подразделениям или клиентам и выпускать собственные цифровые сертификаты для внутреннего, а возможно, и внешнего использования.

Цифровой сертификат — это электронный документ, состоящий из следующих частей:

- идентификатора владельца сертификата (уникально определяющего его как в реальном, так и в виртуальном пространстве);
- открытого ключа владельца сертификата;
- данные самого сертификата, например, срока действия;
- ЭЦП CA, подтверждающей, что указанные данные проверены и соответствуют действительности.

Каким образом открытый ключ субъекта будет надежно доставлен до CA и как CA проверяет корректность данных субъекта, — это вопрос политики формирования сертификатов CA. Тем не менее и в этом случае возможны различные варианты атак, такие как в марте 2001 года, когда группа злоумышленников получила для себя у одной из CA-компаний сертификаты на имя корпорации Microsoft Corp.

Однако после того, как получен корректный цифровой сертификат, его распространение между участниками будет очень простым. Субъект может публиковать свой сертификат на своем сервере, автоматически высылать его каждому новому участнику информационного обмена либо при инициализации каждой новой сессии поручить CA предоставлять свой сертификат по запросу пользователей.

При такой централизованной модели хорошо внедряется иерархическая система доверия. Если субъект имеет свой цифровой сертификат, подписанный СА, которому широко доверяют, то он сам может выступить в качестве СА для ряда подчиненных или независимых субъектов. В свою очередь каждый из этих субъектов, имея свой сертификат, заверенный СА второго уровня, может выступить как СА третьего уровня и т. д. Теперь участник процесса информационного обмена, получив от субъекта его сертификат, подтвержденный СА n -го уровня, о котором сам участник ничего не знает, должен подниматься вверх по иерархии, получая все сертификаты до того СА, которому доверяет данный участник (вплоть до первого СА).

Альтернативная, *децентрализованная система* обеспечения уверенности в принадлежности открытого ключа субъекту (англ. *KeyRing*) применяется в криптографическом программном обеспечении PGP (Pretty Good Privacy). Здесь организована система перекрестного доверия, т. е. субъект доставляет свой открытый ключ до других субъектов, которым доверяет он и которые доверяют ему. Эти субъекты формируют цифровой сертификат, подписывая его своей ЭЦП, и направляют его в адрес серверов хранения ключей. PGP автоматически анализирует количество поступивших сертификатов данного субъекта и, в зависимости от степени доверия подписавшим его субъектам, определяет степень доверия открытому ключу данного субъекта. Теперь каждый, в том числе и сам субъект, может получить свой сертификат с сервера хранения ключей, удостоверенный достаточно большим количеством доверенных субъектов.

Для любой схемы распространения открытых ключей необходимо выполнение следующих функций.

- ❑ Регистрация, т. е. первичное информирование СА о существовании субъекта, чей открытый ключ подлежит сертификации.
- ❑ Инициализация. Получение клиентами СА цифрового сертификата самого СА и другие процедуры.
- ❑ Сертификация, т. е. формирование цифрового сертификата субъекта с его открытым ключом.
- ❑ Генерация пары ключей. Это может быть сделано самим субъектом либо по договоренности с СА, который может сгенерировать ключи для субъекта.
- ❑ Восстановление ключей. Если ключи для субъекта были сгенерированы СА, то в случае утраты субъектом ключей, СА может восстановить их из своего доверенного хранилища.
- ❑ Обновление ключей. Как всякая постоянная основа для работы криптографических алгоритмов, ключи должны периодически обновляться, чтобы свести к минимуму риск реализации атак по взлому ключей. Соответственно цифровые сертификаты также должны быть обновлены.

- ❑ **Перекрестная сертификация.** СА, принадлежащие к различным доменам ответственности, могут столкнуться с задачей, когда их пользователям необходимо работать с ними одновременно. Для обеспечения соответствующего уровня доверия СА должны уметь взаимно сертифицировать друг друга.
- ❑ **Отзыв сертификатов.** В ряде случаев (компрометация секретного ключа, прекращение деятельности субъекта или изменение его параметров и т. п.) действующий сертификат должен быть аннулирован. Для этого используется механизм *Списка отозванных сертификатов* (*Certificate revocation List — CRL*), которой СА распространяется среди пользователей.

В качестве иллюстрации к жизненному циклу ключей приведем схему (рис. 22.37) из книги "Технологии управления ключами" (см. главу 13), очень детализированного материала по криптографии [Menezes]. Впрочем, данный рисунок с соответствующими поправками можно использовать и для иллюстрации жизненного цикла ключей симметричного шифрования. Подробные разъяснения можно найти там же [Menezes].

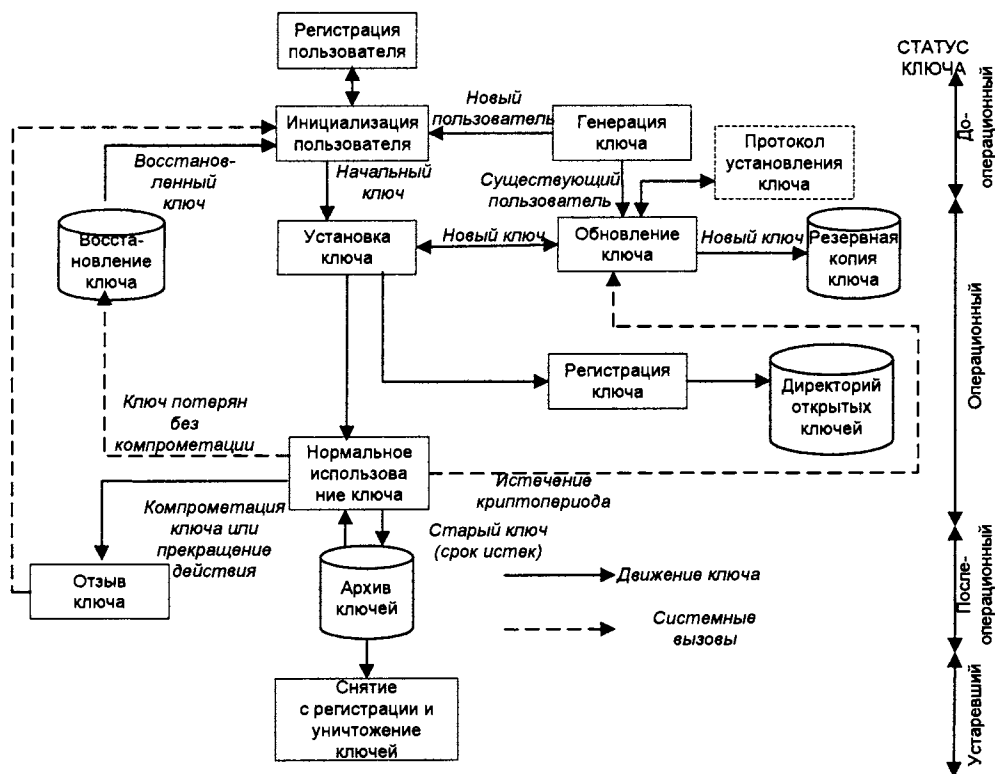


Рис. 22.37. Жизненный цикл ключей

Цифровые сертификаты X.509 v3

Что такое цифровой сертификат, рассматривалось в разд. "Инфраструктура открытых ключей" настоящей главы. В данном разделе будут рассмотрены подробнее структура и формат самого сертификата X.509 как одного из наиболее широко используемых протоколов. Различным аспектам данного вопроса посвящено большое количество документов (RFC 2459, 2510, 2511, 2527, 2528, 2559, 2560, 2585, 2587, 3029, 3039).

Повторим лишь то, что основной целью цифрового сертификата служит обеспечение уверенности пользователя открытого ключа данного субъекта в том, что указанный открытый ключ действительно принадлежит заявленному субъекту.

Рассмотрим подробнее структуру самого сертификата (рис. 22.38).

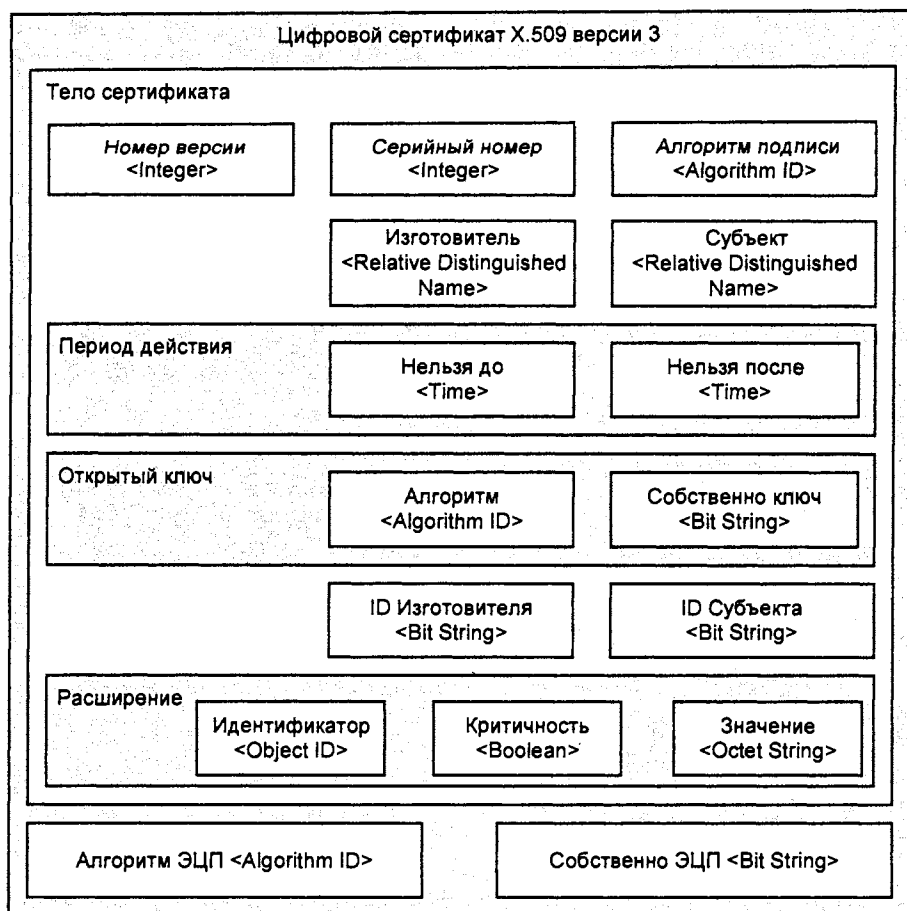


Рис. 22.38. Формат сертификата X.509 версии 3

Поле *Номер версии* — определяет номер версии используемого сертификата. Равен 0 (версия 1), если присутствуют только базовые поля, без расширений и идентификаторов субъекта/изготовителя. Если присутствуют идентификаторы — равен 1 (версия 2), а если присутствует расширение — 2 (версия 3).

Поле *Серийный номер* — уникальный номер, присваиваемый каждому сертификату.

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

Поле *Изготовитель* — идентифицирует того, кто выпустил и подписал сертификат, т. е., фактически сертифицирующего уполномоченного (CA — Certification Authority). Формат данного поля представляет Относительное отдельное имя (Relative Distinguished Name), используемое, например, в службах и протоколах каталогов (X.500 или LDAP) и состоящее из комбинации различных параметров, таких как идентификатор страны, области, организации, домен, субдомен, фамилия/имя или наименование, дающих в совокупности некий уникальный идентификатор. Стандартный набор таких параметров или атрибутов определен в протоколе X.500.

Поле *Субъект* — идентифицирует владельца сертификата: того, кому принадлежит открытый ключ, указанный в сертификате. Формат также представлен в виде Относительного отдельного имени.

Поле *Период действия* — временной интервал, в течение которого СА гарантирует поддержку статуса сертификата (например, своевременное сообщение о его отзыве). Представлено двумя полями — временем начала и завершения периода.

Поле *Открытый ключ* — два поля, первое из которых указывает, какой алгоритм был использован для генерации ключа, а второе представляет собственно открытый ключ в виде набора битов.

Поле *Идентификаторы субъекта и изготовителя* указаны на случай возникновения повторного использования одинакового имени субъекта или изготовителя.

Поле *Расширение* — это дополнительный атрибут, связанный с субъектом, изготовителем или открытым ключом и предназначенный для управления процессами сертификации.

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

Собственно поле *ЭЦП* — набор битов, составляющих электронно-цифровую подпись под данным сертификатом. В ее формировании участвуют поля, указанные в теле сертификата.

Отдельное внимание в описании протокола уделено возможным используемым расширениям. В список функциональных расширений входят:

- ❑ идентификация конкретной пары открытый/секретный ключ изготовителя сертификатов, в случае если изготовитель использует несколько различных ключей для подписания различных сертификатов;
- ❑ идентификация конкретного открытого ключа субъекта, в случае если субъект имеет несколько сертификатов, возможно, от разных изготовителей;
- ❑ цель использования ключа — для шифрования, подписи, формирования других сертификатов;
- ❑ уточнение периода использования — можно сократить период действия сертификата, указанный в поле *Период действия*;
- ❑ политики использования сертификата — указывается, для каких целей выпущен сертификат и приложения, использующие сертификат; могут принимать или отвергать этот сертификат на основе этого расширения;
- ❑ выбор соответствия политик использования сертификата для изготовителя и для субъекта, если имеются различные варианты;
- ❑ альтернативное имя субъекта или изготовителя, если таковое присутствует и отлично от указанного в основной части сертификата;
- ❑ определение, является ли сам субъект СА и насколько глубоко разворачивается сертификационный путь через этот СА.

Кроме того, существуют расширения для прочих задач, ознакомиться с которыми можно в документации.

Теперь рассмотрим структуру *Списка отозванных сертификатов* (англ. *Certificate Revocation List — CRL*) — рис. 22.39.

Поле *Номер версии* — определяет номер версии используемого CRL. Текущая используемая версия — вторая, она обязательна для указания, если CRL используется с расширениями.

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

Поле *Изготовитель* — идентифицирует того, кто выпустил и подписал CRL. Формат данного поля представляет Относительное отдельное имя (Relative Distinguished Name), используемое, например, в службах и протоколах каталогов (X.500 или LDAP) и состоящее из комбинации различных параметров, таких как идентификатор страны, области, организации, домен, субдомен, фамилия/имя или наименование, дающих в совокупности некий уникальный идентификатор. Стандартный набор таких параметров или атрибутов определен в протоколе X.500.

Поле *Выпущен* — указывает время выпуска настоящего CRL.

Поле *Следующий* — указывает время выпуска следующего CRL.

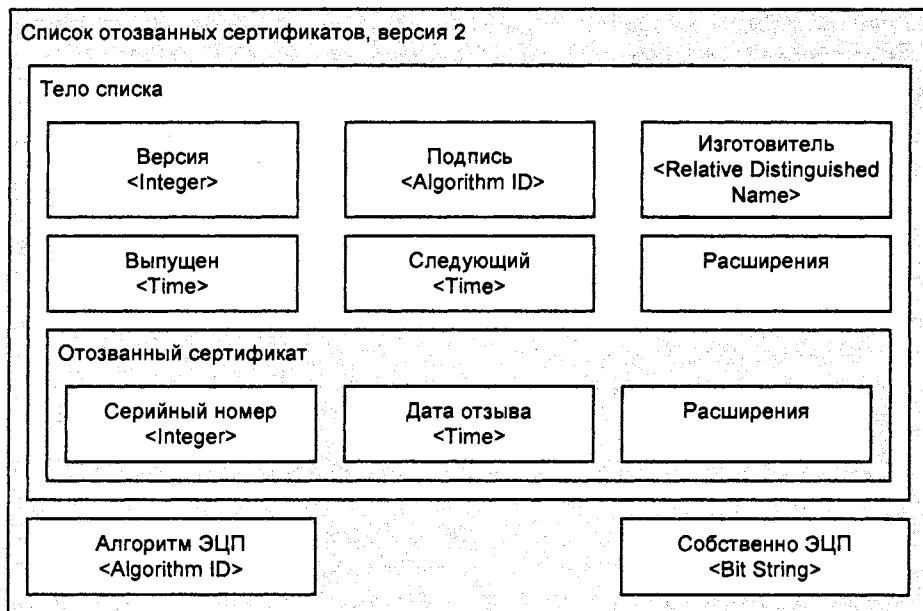


Рис. 22.39. Структура Списка отозванных сертификатов

Поле *Расширение* — это дополнительный атрибут, связанный с пользователем или открытым ключом и предназначенный для управления процессами сертификации.

Поле *Отозванный сертификат* — таких полей может быть несколько — столько, сколько сертификатов отзывается. Об отозванном сертификате приводится следующая информация:

- ☐ серийный номер;
- ☐ дата, с которого вступает в силу отзыв;
- ☐ расширения сертификата.

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

Собственно поле *ЭЦП* — набор битов, составляющих электронно-цифровую подпись под данным сертификатом. В ее формировании участвуют поля, указанные в теле CRL.

Расширения самого CRL включают схожую с сертификатом функциональность, а также дополнительную функциональность:

- ☐ обеспечивают последовательную нумерацию CRL для облегчения управления списками;

- ☐ обеспечивают накопление приложениями информации об отозванных сертификатах (так называемые, *delta CRL*);
- ☐ определяют пункт распространения CRL.

Расширения, связанные с сертификатами, имеют следующую функциональность, отличную от функциональности самого сертификата:

- ☐ определяют причину отзыва сертификата;
- ☐ определяют фиксированный набор инструкций — действий выполняемых с отзываемыми сертификатами;
- ☐ указывают дату компрометации секретного ключа или другой причины недействительности сертификата.

Полный список расширений и функций можно найти в соответствующей документации.

Нам осталось рассмотреть, какие алгоритмы и методы криптографии допустимы в работе с сертификатами.

Как известно, для формирования ЭЦП необходимо использование как минимум двух типов алгоритмов — однонаправленной хэш-функции и криптографического алгоритма с открытыми ключами.

В качестве хэш-функции X.509 использует алгоритмы MD2 (Message Digest 2), MD5 (Message Digest 5), SHA-1 (Secure Hash Algorithm). Алгоритмы с открытыми ключами — RSA (Rivest, Shamir, Adleman) и DSA (Digital Signature Algorithm). Субъектам разрешено использование для открытых ключей следующих алгоритмов: RSA, Diffie-Hellman, DSA.

Privacy Enhanced Mail

Описание протокола Privacy Enhanced Mail (PEM) — *Конфиденциальная почта* — состоит из четырех частей.

- ☐ Процедуры шифрования сообщений и аутентификации [RFC-1421].
- ☐ Управление ключами на основе сертификатов (RFC-1422).
- ☐ Алгоритмы, модули и идентификаторы (RFC-1423).
- ☐ Сертификация ключей и прочие сервисы (RFC-1424).

Протокол предлагает следующую функциональность обеспечения безопасности при электронном обмене информацией:

- ☐ конфиденциальность;
- ☐ аутентификацию;
- ☐ уверенность в целостности сообщения;
- ☐ апеллируемость источника.

Важно отметить, что используемые механизмы безопасности не выдвигают дополнительных требований к системе передачи сообщений или промежу-

точным узлам, участвующим в передаче сообщения. Это позволяет применять механизмы РЕМ, не влияя на функционирование остальных систем. Механизмы работы протокола должны быть интегрированы с почтовым протоколом *агента пользователя* (*User Agent — UA*). UA — это прикладной процесс, который взаимодействует с системой пересылки сообщений (*Message Transfer System — MTS*).

В описании протокола отдельно выделены области безопасности, которые не покрываются протоколом и, соответственно, должны быть учтены специалистами по безопасности и обеспечены защитой другими механизмами:

- ☐ контроль доступа;
- ☐ сокрытие объема трафика;
- ☐ точность списка адресации;
- ☐ управление маршрутизацией;
- ☐ работа на одном компьютере множества пользователей;
- ☐ уверенность в том, что сообщение получено, и обеспечение неотрекаемости факта получения сообщения;
- ☐ автоматическое связывание подтверждения на сообщение с самим сообщением;
- ☐ выявление дублированных сообщений.

В протоколе различают два вида криптографических ключей, выполняющих различные функции.

- ☐ Первый тип — это *ключи шифрования данных* (*Data Encryption Keys — DEK*), используемые для шифрования текста сообщения и формирования *кода целостности сообщения* (*Message Integrity Check — MIC*). Они выполняют функцию сеансовых ключей шифрования. При использовании асимметричной криптографии DEK применяется для шифрования подписи сообщения. DEK генерируется отдельно для каждого сообщения и не требует предварительного распространения между адресатами.
- ☐ Второй тип — *ключи обмена* (*Interchange Keys — IK*), используются для шифрования DEK, для того чтобы можно было отправлять их вместе с сообщениями. Один и тот же IK используется для общения между данным отправителем и данным получателем в определенный период времени, т. е. является мастер-ключом между двумя конкретными абонентами. Соответственно необходимо поддерживать увязку таких ключей для каждого субъекта с его отправителями/получателями. IK бывают различными в зависимости от использования симметричных или асимметричных криптографических механизмов:
 - для симметричных схем IK — это единичный симметричный ключ, предварительно распределенный между отправителем и получателем, который используется для шифрования DEK и MIC;

- для асимметричных схем для шифрования DEK используется открытый ключ получателя, а для шифрования MIC — секретный ключ отправителя (открытый ключ может быть извлечен из соответствующего сертификата).

Различают несколько типов PEM-сообщений.

- ❑ **Зашифрованное (Encrypted).** Это означает, что к данному сообщению применены все имеющиеся в распоряжении механизмы защиты (конфиденциальность, аутентификация, контроль целостности и апеллируемость), если используется асимметричная криптография.
- ❑ **Только MIC (MIC-Only).** Это означает, что к данному сообщению применены все те же механизмы защиты, что и к предыдущему типу, кроме обеспечения конфиденциальности. Для обеспечения защиты текста от воздействия нижележащих транспортных протоколов к сообщению применено кодирование (не путать с шифрованием).
- ❑ **Чистый MIC (MIC-Clear).** Это означает, что к данному сообщению применены все те же механизмы защиты, что и к предыдущему типу. При этом кодирование текста также пропускается. Такое сообщение сможет прочитать даже получатель, не имеющий соответствующего программного обеспечения для PEM. Однако отсутствие транспортной кодировки подразумевает, что MIC может быть подтвержден, только если MTS не вносила изменений в сообщение при передаче и приеме, либо такие изменения заранее определены.
- ❑ **Список Отзыванных сертификатов (Certificate Revocation List — CRL).** Специальное сообщение, используемое для рассылки листов отзыванных сертификатов. Формат этого сообщения описан в отдельном RFC-1422.

Формирование PEM-сообщения состоит из четырех фаз в следующем порядке:

1. **Локальная форма (Local form).** Применяется ко всем типам сообщения, кроме CRL. Сообщение создается в форме, присущей создающей системе, в том смысле что на этой стадии дополнительные механизмы кодирования не применяются.
2. **Каноническая форма (Canonical form).** Используется для тех же типов сообщения, что и локальная. Текст сообщения конвертируется в универсальную каноническую форму, такую же как представление inter-SMTP (RFC821 и 822).
3. **Аутентификация и шифрование (Authentication and encryption).** Аутентификация используется для тех же типов сообщения, что и предыдущая. При этом производится расчет MIC. Шифрование производится только для сообщения типа Encrypted.
4. **Кодирование для печати (Printable encoding).** Используется для типов сообщения Encrypted и MIC-Only. Используя в качестве входящих зна-

чений результат фазы 3, производится представление в символы, универсально представленные на всех интернет-сайтах.

Приведем список используемых криптографических механизмов.

- ☐ Алгоритмы шифрования сообщения:
 - алгоритм DES (Data Encryption Standard) в модуле CBC (Cipher Block Chaining) — (DES-CBC) с ключом 56 бит.
- ☐ Алгоритмы MIC:
 - алгоритм RSA-MD2, выдающий 128-битовое значение;
 - алгоритм RSA-MD5, выдающий 128-битовое значение.
- ☐ Алгоритмы управления симметричными ключами:
 - алгоритм DES-ECB (Electronic CodeBook);
 - алгоритм DES-EDE (Encrypt-Decrypt-Encrypt).
- ☐ Алгоритмы управления асимметричными ключами:
 - алгоритм RSA (Rivest-Shamir-Adleman).
- ☐ Алгоритмы асимметричного шифрования для DEK и MIC:
 - алгоритм RSA (Rivest-Shamir-Adleman).
- ☐ Алгоритмы для электронно-цифровой подписи:
 - алгоритм MD2 с RSA-шифрованием.

Secure/Multipurpose Internet Mail Extensions

Сам по себе протокол MIME (Multipurpose Internet Mail Extensions — Многоцелевые расширения интернет-почты) — это формат содержимого сообщений сети для обеспечения представления информации и обмена данными без потерь для сложных составных сообщений, текстовых и нетекстовых. Этот формат необходим для включения в одно сообщение различных объектов (таких как изображения или аудиоэлементы). Основные такие элементы следующие:

- ☐ text — текстовая информация как набор строковых значений;
- ☐ multipart — совокупность различных типов данных в одном сообщении;
- ☐ application — используется для передачи данных приложения или бинарных данных;
- ☐ message — используется для инкапсуляции в сообщение почтового послания;
- ☐ image — данные изображения (картинки);
- ☐ audio — аудиоданные, например, голос;
- ☐ video — видеоданные или движущиеся изображения, возможно, с аудиоэлементами как частью видеоформата.

Соответственно Secure/MIME (S/MIME) [RFC-2633] предназначен для обеспечения надежного пути для отправки и получения MIME-данных. Указанная надежность обеспечивается решением стандартных задач безопасности — аутентификации, целостности и аппелируемости за счет электронно-цифровой подписи, сокрытие данных за счет шифрования.

При этом S/MIME не ограничивается использованием только электронной почты, он может быть использован в других сетевых протоколах, таких как HTTP.

Требования к алгоритмам протокола в [RFC-2633] включают следующее:

- ❑ в качестве алгоритма формирования хэш-значения должна присутствовать поддержка SHA-1, опционально MD5;
- ❑ в качестве алгоритма формирования электронно-цифровой подписи должна присутствовать поддержка DSS, опционально RSA;
- ❑ в качестве алгоритма шифрования должна присутствовать поддержка 3DES, опционально RC2 с ключом 40 бит.

В описании протокола используется понятие *Синтаксис криптографического сообщения* (*Cryptographic Message Syntax — CMS*) — [RFC-2630], в котором описываются форматы работы с сообщениями, в которых используются криптографические механизмы. Для S/MIME-сообщений могут быть использованы следующие типы содержимого: *Data* (Данные), *SignedData* (Подписанные данные — обеспечивается целостность элемента), *EnvelopedData* (Запечатанные данные — обеспечивается конфиденциальность элемента).

В качестве наиболее интересного решения, используемого в данном протоколе, можно привести правила принятия решения о необходимости использования отправителем того или иного алгоритма шифрования. Таких правил три.

1. Известны возможности.

Если отправитель имеет список используемых возможностей (алгоритмов) принимающей стороны, тогда он должен использовать первую возможность из указанного получателем списка как наиболее предпочтительную.

2. Неизвестны возможности. — Известен использованный алгоритм.

В случае, если отправитель не имеет данных об используемых принимающей стороной возможностях, но имеет хотя бы одно принятое от адресата послание, и последнее из имеющихся таких сообщений имеет подтвержденную электронно-цифровую подпись, то исходящее сообщение должно использовать тот же алгоритм, что и в последнем поступившем письме.

3. Неизвестны возможности — неизвестна версия S/MIME.

Если отправитель не имеет данных ни об имеющихся возможностях, ни о версии S/MIME на принимающей стороне, он должен использовать

3DES, как наиболее сильный. В случае невозможности использования 3DES следует использовать RC2 с 40 битным ключом.

Secure Electronic Transaction

Общий принцип работы

Спецификация механизма Secure Electronic Transaction (*SET*) — *защищенного процесса электронной оплаты* — была разработана в сотрудничестве организациями VISA и MasterCard и представляет собой несколько объемных документов, основные из которых:

- ❑ SET Business Description (Производственное описание);
- ❑ SET Programmers Guide (Руководство программиста);
- ❑ SET Formal Protocol Definition (формализованное описание протокола).

На сайте <http://www.set.org/>, посвященном протоколу, можно найти и другие документы, такие как SET External Interface Guide (Руководство по внешним интерфейсам), полезное для практической реализации, но основными являются все же три перечисленные.

Поскольку уже только эти документы имеют объем около тысячи страниц, остановимся на наиболее общем и кратком SET Business Description.

В работе протокола предусматриваются следующие участники:

- ❑ *Владелец платежной карты* (обычно *кредитной* карты, однако не будем заострять внимание на различиях между дебетовыми и кредитными картами) или просто *владелец* использует карту, произведенную эмитентом, и взаимодействует с торговцем, защищенным образом.
- ❑ *Эмитент* — финансовая организация, открывшая для владельца счет и выпустившая соответствующую карту. Эмитент гарантирует выполнение платежа для авторизованной транзакции, произведенной с помощью карты в соответствии с правилами брэнда и соответствующим законодательством.
- ❑ *Торговец* предлагает владельцу товар или услугу в обмен на оплату с помощью защищенного электронного взаимодействия. Принимающий платежные карты торговец должен быть связан с получателем.
- ❑ *Получатель* — финансовая организация, открывшая счет для торговца и обрабатывающая платежи и подтверждения, произведенные с помощью платежной карты.
- ❑ *Платежный шлюз (шлюз)* — устройство, управляемое получателем или выделенной третьей стороной, обрабатывающее платежные сообщения торговца, в том числе платежные инструкции от владельца.

- *Брэнд (Фабрикант)* — финансовая организация, создавшая систему платежных карт, обслуживающая ее, устанавливающая правила работы и предоставляющая сеть для взаимодействия финансовых организаций в системе.
- *Третья сторона.* Иногда эмитент и получатель договариваются о передаче процесса обработки третьей стороне. В данном случае протокол не делает отличия между обработчиком платежей и финансовой организацией.

Не будем повторять те криптографические принципы, используемые протоколом SET, которые уже были изложены в *части V*. Остановимся только на новых понятиях.

Для обеспечения независимого "многоцелевого" заверения документов SET вводит технологию *сдвоенной подписи*. Для представления необходимости ее использования приводится следующий пример. Торговец высылает владельцу предложение на товар и подтверждение банку владельца для перевода денег, если владелец примет предложение. Но при этом торговец не хочет, чтобы банк был осведомлен об условиях предоставления товара, а владелец был осведомлен о платежных реквизитах торговца.

Для обеспечения в документе подобных свойств формирующий сообщение (в нашем примере — торговец):

1. Разделяет его на две части: первая (M_1) предназначена банку, вторая (M_2) — владельцу.
2. Для каждой части формирует хэш-значение (message digest):
 $H_1 = \text{hash}(M_1)$, $H_2 = \text{hash}(M_2)$.
3. Соединяет два хэш-значения вместе и формирует хэш-значение результата:
 $H = \text{hash}(H_1 \mid H_2)$.
4. Хэш-значение результата шифрует своим закрытым ключом $S = \text{sign}(H)$.

Теперь к каждой части сообщения он добавляет общую подпись и хэш-значение другой части. Формируются следующие пакеты: $P_1 = (M_1, S, H_2)$ — для банка и $P_2 = (M_2, S, H_1)$ — для владельца. Получатель конкретной части может сам вычислить хэш-значение полученной им части сообщения, сложить его с хэш-значением второй части и с помощью открытого ключа торговца проверить корректность сдвоенной электронно-цифровой подписи. При этом достигается тот факт, что, проверяя целостность всего сообщения, получатель имеет доступ к тексту только своей половины.

Участники процесса обмена электронными платежами — владелец, торговец, шлюз, получатель и эмитент — должны обладать таким механизмом обеспечения безопасности, как цифровой сертификат. Основная цель сертификатов — авторизовать принадлежность участника системе: получателя и эмитента — конкретному бренду, торговца и шлюза — получателю.

У владельца сертификат является электронной интерпретацией платежной карты, он только не содержит в явном виде номер счета и дату истечения. Эти данные вместе с некоторым секретным значением преобразованы в хэш-значение, содержащееся в сертификате. Таким образом, знание этой информации является подтверждением владения картой. Следует различать:

- ☐ сертификаты обмена ключами (*К-сертификат*), предназначенные для хранения открытого ключа, которым шифруется симметричный сессионный ключ, который в свою очередь участвует в шифровании набора данных (например, регистрационной формы);
- ☐ обычный сертификат, предназначенный для хранения открытого ключа, которым проверяется электронно-цифровая подпись владельца сертификата.

Протокол различает два варианта совершения сделок — с помощью *он-лайн каталога* и с использованием *электронного каталога*.

В первом варианте покупка осуществляется путем посещения владельцем Веб-сайта торговца. После выбора товара покупателю предлагается форма заказа для заполнения и подтверждения. После ввода данных в форму SET позволяет защищенным образом передать владельцу платежные инструкции, а торговцу — получить подтверждение и сам платеж.

Во втором варианте список товаров (каталог) распространяется на носителях в режиме офф-лайн (дискеты, компакт-диски и т. п.). Каталог сопровождается формой заказа. В случае выбора владельцем товара он может доставить заполненную форму до торговца различными способами — электронной почтой или прямым контактом с сайтом торговца, как в первом варианте.

SET рассматривает следующие действия, производимые участниками в процессе работы.

- ☐ Регистрация владельца.
- ☐ Регистрация торговца.
- ☐ Запрос на приобретение (заказ).
- ☐ Авторизация платежа.
- ☐ Проведение платежа (получение).

Регистрация владельца

Для начала работы в рамках протокола владелец должен быть зарегистрирован у Сертифицирующего уполномоченного (CA — Certification Authority). Этот процесс состоит из следующих шагов.

- ☐ Владелец (программное обеспечение от его имени) запрашивает копию CA-сертификата для обмена ключами (CA key-exchange certificate, К-сертификат).

- СА формирует ответ, создает для него электронно-цифровую подпись (ЭЦП) с помощью своего закрытого ключа и высылает владельцу вместе со своим К-сертификатом, содержащим открытый ключ.
- Владелец подтверждает К-сертификат, проверяя последовательно подписи на сертификатах от данного СА до корневого СА (подробнее об иерархии сертификатов в *разд. "Инфраструктура открытых ключей" данной главы*) и проверяет ЭЦП на ответе СА. Затем удостоверенный К-сертификат сохраняется, а владелец запрашивает регистрационную форму от СА, генерирует сессионный симметричный ключ и шифрует этим ключом запрос регистрационной формы. Затем симметричный ключ и номер счета владельца шифруются открытым ключом СА из сертификата СА. Полученные запрос, сессионный ключ и номер счета высылаются СА.
- СА дешифрует сессионный ключ и номер счета владельца своим закрытым ключом, а запрос на регистрационную форму — сессионным ключом. Затем СА определяет по первым 6—11 цифрам номера счета финансовую организацию, к которой относится владелец и выбирает соответствующую регистрационную форму (либо указание владельцу, где получить форму). СА высылает пописанную регистрационную форму и свой обычный (не для обмена ключами) сертификат владельцу.
- Владелец проверяет сертификат до корневого СА, так же как и К-сертификат проверяет ЭЦП на форме. Затем владелец генерирует для себя пару закрытый/открытый ключ, если это не было сделано ранее, и заполняет присланную СА регистрационную форму, которая послужит основанием для запроса на сертификат для владельца. Владелец генерирует случайное число N, которое будет использовано СА при формировании сертификата для владельца; формирует регистрационное сообщение (которое состоит из заполненной регистрационной формы и открытого ключа владельца) и подписывает его. На следующем этапе происходит генерация двух новых симметричных ключей, один помещается в регистрационное сообщение (им СА будет шифровать ответ), вторым шифруется все регистрационное сообщение. Второй ключ, номер счета, дата окончания срока действия карты и случайное число N шифруются открытым ключом СА из К-сертификата. Все перечисленное отсылается СА.
- СА дешифрует второй симметричный ключ и информацию по счету владельца своим закрытым ключом (для К-сертификата) и дешифрует запрос на сертификат вторым симметричным ключом. Затем СА проверяет ЭЦП запроса, используя открытый ключ владельца, и проверяет соответствие данных запроса, регистрационной формы и информации по счету владельца. Для формирования сертификата СА генерирует свое случайное число M, вместе с N, номером счета и датой срока действия карты преобразуются в хэш-значение, которое и будет указано в серти-

фикате. СА генерирует сертификат для владельца и подписывает своим закрытым ключом обычного сертификата, формирует ответ владельцу и подписывает его. После этого СА шифрует ответ, куда входит и число M , первым секретным ключом и отправляет владельцу вместе с сертификатом.

- ❑ Владелец проверяет свой сертификат до корневого СА и сохраняет его для дальнейшего использования. Затем дешифрует ответ СА первым симметричным ключом, проверяет его ЭЦП и извлекает число M . Сохраняет числа M и N для дальнейшего использования.

Владелец зарегистрирован и может начинать работу.

Регистрация торговца

Торговец перед началом работы также должен быть зарегистрирован у Сертифицирующего уполномоченного (СА — Certification Authority). Этот процесс состоит из нескольких шагов.

- ❑ Торговец (программное обеспечение от его имени) запрашивает копию СА сертификата для обмена ключами (CA key-exchange certificate, К-сертификат) и регистрационную форму. Здесь на лицо отличие от процесса регистрации владельца, где регистрационная форма получается позже.
- ❑ Получив запрос, СА определяет необходимую для данного торговца регистрационную форму, электронно подписывает ее обычным закрытым ключом и отправляет торговцу вместе с сертификатом и К-сертификатом.
- ❑ Торговец проверяет сертификат до корневого и сохраняет его для дальнейшего использования. Торговец генерирует (если до этого еще не сделал) две пары ключей — для обмена ключами (К-сертификата) и электронно-цифровой подписи (обычного сертификата) и заполняет регистрационную форму, в которую входит наименование, адрес и идентификатор. Затем торговец формирует запрос на сертификаты, в который входит регистрационная форма, добавляет свои открытые ключи для сертификата и К-сертификата, подписывает все это закрытым ключом обычного сертификата. Торговец генерирует симметричный ключ шифрования и шифрует им весь перечисленный набор данных, сам симметричный ключ и данные по счету торговца шифруются открытым ключом из К-сертификата СА. Все это отправляется СА.
- ❑ СА дешифрует своим закрытым ключом симметричный ключ и с помощью него дешифрует весь запрос, проверяет его подпись, используя присланные торговцем открытые ключи. Далее СА сверяет информацию из регистрационной формы и информацию о торговце, поступившую от получателя (способ доставки этой информации находится вне описания

SET). После этого СА генерирует соответствующие сертификаты торговца и новый симметричный ключ, которым шифрует сертификаты, а сам симметричный ключ шифруется открытым ключом торговца из его нового К-сертификата. Все это отправляется торговцу.

- ❑ Торговец дешифрует симметричный ключ своим закрытым ключом К-сертификата, а с помощью симметричного ключа — свои сертификаты. Проверяет их до корневого и сохраняет для дальнейшего использования.

Торговец зарегистрирован.

Запрос на покупку

Значимая деятельность владельца по протоколу начинается с запроса на покупку, который состоит из следующих шагов.

- ❑ SET вступает в действие после того, как владелец произвел заполнение полей формы заказа на покупку товара или услуги. Программное обеспечение владельца от его имени формирует начальный запрос и посылает его торговцу.
- ❑ Получив начальный запрос, торговец генерирует ответ и подписывает его своим закрытым ключом. Полученное сообщение направляется владельцу вместе с сертификатом торговца и К-сертификатом шлюза.
- ❑ Владелец проверяет полученные сертификаты до корневого и сохраняет их для будущего использования. На основе информации из заполненной формы владелец создает Информацию о заказе (ИЗ), а также создает Инструкцию о платеже (ИП), в которые помещает идентификатор данной транзакции, указанный торговцем. Затем владелец устанавливает двоячную электронно-цифровую подпись на ИЗ и ИП. На следующем этапе владелец генерирует случайный симметричный ключ (ключ-1), которым шифрует ИП, а сам ключ и номер счета владельца шифрует, используя ключ К-сертификата шлюза. Все полученное (ИЗ, ИП, ключ + номер счета) и сертификат владельца отсылаются торговцу.
- ❑ Торговец проверяет сертификат владельца до корневого, а затем с помощью самого сертификата проверяет подпись на ИЗ. Затем обрабатывает запрос, при этом отсылая ИП шлюзу. Далее торговец создает ответ на заказ, подписывает его своим закрытым ключом и отправляет владельцу вместе с сертификатом. Когда поступает авторизация (подтверждение) платежа, торговец выполняет заказ владельца (доставляет товар или услугу).
- ❑ Программное обеспечение владельца проверяет сертификат до корневого и проверяет ЭЦП. Затем выполняет необходимые действия (например, сообщает владельцу об успешном завершении или демонстрирует ответ) и сохраняет ответ.

Авторизация платежа

Для успешного выполнения транзакции, платеж торговцу должен быть авторизован. Процесс авторизации происходит между торговцем и шлюзом и складывается из следующего.

- ❑ Торговец формирует запрос на авторизацию, в который входит авторизуемая сумма и идентификатор транзакции из ИЗ, подписывает его своим закрытым ключом. Затем генерирует случайный симметричный ключ (ключ-2), шифрует этим ключом запрос, а сам ключ шифруется открытым ключом из К-сертификата шлюза. Все это, а также ИП владельца (сформированные при запросе на покупку), сертификат владельца, сертификат и К-сертификат торговца отправляется шлюзу.
- ❑ Шлюз проверяет сертификаты торговца до корневого. Затем шлюз дешифрует ключ-2 с помощью своего закрытого ключа от К-сертификата, а с помощью ключа-2 дешифрует запрос торговца на авторизацию и проверяет его ЭЦП с помощью сертификата торговца. Далее шлюз проверяет сертификат владельца до корневого, дешифрует ключ-1 с помощью своего закрытого ключа от К-сертификата, а с помощью ключа-1 дешифрует ИП и проверяет двоячную электронно-цифровую подпись владельца на ИП. После этого шлюз проверяет соответствие между ИП и запросом на авторизацию с помощью идентификатора транзакции и отправляет запрос на авторизацию финансовой организации владельца. Затем шлюз создает авторизационный ответ, подписывает его своим закрытым ключом, генерирует случайный симметричный ключ (ключ-3), шифрует им ответ, а сам ключ-3 шифрует открытым ключом из К-сертификата торговца. Шлюз опционально (по требованию получателя) создает фиксирующий токен и подписывает его своим закрытым ключом. Далее генерирует случайный симметричный ключ (ключ-4), которым шифрует токен, а сам ключ-4 и информацию по счету владельца шифрует своим открытым ключом К-сертификата шлюза. Все подготовленное и сертификат шлюза отправляется торговцу.
- ❑ Торговец проверяет сертификат шлюза до корневого, дешифрует своим закрытым ключом К-сертификата ключ-3, а ключом-3 — авторизационный ответ. Торговец проверяет электронно-цифровую подпись на ответе и исполняет заказ на товар/услугу. Фиксирующий токен, ключ-4 и информация по счету владельца сохраняются торговцем для будущего использования.

Фиксация платежа

После того, как владелец произвел заказ, торговцу требуется оплата. Между запросом на авторизацию платежа и запросом на оплату обычно проходит некоторое время.

- ❑ Торговец создает запрос на фиксацию платежа, который содержит общую сумму и идентификатор транзакции (из ИЗ) и подписывает его

своим закрытым ключом. Затем генерирует сессионный симметричный ключ (ключ-5), шифрует им запрос, а сам ключ шифрует открытым ключом из К-сертификата шлюза. Зашифрованный запрос, зашифрованный сессионный ключ, а также фиксирующий токен (если он был создан) и сертификаты (обычный и К-сертификат) торговца отсылаются шлюзу.

- Шлюз проверяет сертификаты торговца до корневого. Затем своим закрытым ключом для своего К-сертификата шлюз дешифрует ключ-5, с помощью ключа-5 получает фиксационный запрос и проверяет электронно-цифровую подпись открытым ключом из сертификата торговца. Далее, закрытым ключом для своего К-сертификата шлюз дешифрует ключ-4, с помощью ключа-4 получает фиксирующий токен и убеждается в соответствии фиксационного запроса фиксирующему токenu. Шлюз направляет фиксационный запрос финансовой организации владельца (Эмитенту). На следующем этапе шлюз формирует ответ на фиксационный запрос, подписывает его своим закрытым ключом, генерирует случайный симметричный ключ-6, шифрует им ответ, а сам ключ шифрует открытым ключом из К-сертификата торговца. Полученный результат отправляется торговцу.
- Торговец, получив ответ, проверяет сертификат шлюза до корневого, дешифрует своим закрытым ключом К-сертификата ключ-6, а ключом-6 дешифрует ответ на фиксационный запрос. Проверив электронно-цифровую подпись шлюза с помощью сертификата шлюза, торговец сохраняет ответ учета с платежом, пришедшим от получателя.

Процесс взаимодействия завершен.

Аудит использования сетевых протоколов безопасности



Если контролер производит инспекцию информационной системы путем беседы с соответствующим специалистом, а тот пытается сбить контролера с толку умными словами в виде конкретных названий протоколов безопасности, не следует пугаться. Необходимо аккуратно записать все названия и указанные специалистом области применения, а затем, обратившись к независимому консультанту либо прочитав техническую часть этого раздела, попытаться разобраться самостоятельно.

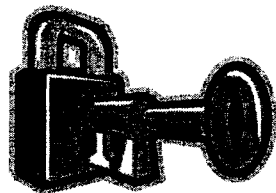
Если, наоборот, специалист не может внятно объяснить, какие протоколы используются либо указать причину, почему было решено их не использовать, можно сделать соответствующий вывод об уровне подготовки специалиста или уровне защищенности системы. Попытки сказать, что в системе используются секретные протоколы и закрытые алгоритмы, как показывает

практика, свидетельствуют скорее в сторону уязвимости, а не дополнительной надежности системы. Опубликованные алгоритмы и протоколы обычно подвергаются многократной проверке независимых экспертов, которые выясняют их уязвимости, что дает возможность разработчикам их устранять. Секретные механизмы могут иметь скрытые уязвимости и ошибки реализации, которыми может воспользоваться тот, кто обнаружит их раньше самого разработчика или авторизованного испытателя.

Выяснив набор используемых стандартных протоколов, следует произвести изучение их известных уязвимостей, а также определить рекомендуемые специалистами конфигурационные параметры (если подобные сведения имеются в наличии). Затем вопросы об уязвимостях и конфигурации можно переадресовать уполномоченному специалисту системы (либо при соответствующей квалификации и правах проверить самостоятельно), таким образом, подтвердив или опровергнув надежность системы.

Если все же вам пришлось инспектировать систему без дополнительного участия специалистов, которые могли ответить на ваши вопросы (например, только при использовании документации), то для облегчения понимания терминов, а также обозначений таких, как PAP (SPAP), CHAP (MS-CHAP), S/Key, TACACS (TACACS+), RADIUS, Kerberos, SHTTP (S-HTTP, SecureHTTP), SSL (Secure Sockets Layer), Secure Shell (SSH), SOCKS, IPSec, L2F, PPTP, L2TP, X.509, PKI (Public Key Infrastructure), LDAP вы можете использовать *разд. "Техническое описание протоколов" данной главы.*

Глава 23



Автоматизированные средства безопасности

Руководители, которым на подпись приносят проект приобретения нового автоматизированного средства по безопасности, порой находятся в растерянности. Только полгода назад они подписывали документ о существенных расходах на одно автоматизированное средство — теперь вот еще одно. Не обманывают ли их собственные подчиненные? Какие средства действительно необходимы предприятию?

Попробуем провести краткий экскурс в область автоматизированных средств защиты информации для руководителей, а также для всех, кто хочет узнать о продуктах информационной безопасности больше, но не слишком.

В этой главе рассмотрим:

- ☐ типовой набор, который должен присутствовать в каждой среднего размера организации;
- ☐ расширенные средства информационной безопасности для крупной корпорации;
- ☐ специализированные средства для помощи в отдельных направлениях, например, таких как обнаружение зловредных программ или борьба с сетевыми атаками;
- ☐ некоторые инструменты, которые не обязательны, но порой бывают очень удобны в работе.

Мы не хотим производить рекламу, явную или скрытую, какого-либо одного продукта или компании. Целью является скорее описать общую функциональность и возможности класса продуктов и моменты, на которые следует обратить внимание при их использовании.

Антивирусы



Обычно первым программным обеспечением по безопасности, которое появляется в компании, является антивирусный комплект. Его наличие еще не означает, что организация серьезно задумалась о безопасности, просто в современном мире присутствие антивируса (далее — А\В) на компьютере —

это такая же обычная и необходимая вещь, как присутствие текстового процессора. Критерием выбора A\B порой является не качество и надежность, а доступность пиратской копии, малый объем, возможность переноса на дискете или низкая стоимость.

Если перейти к более профессиональному подходу, то в первую очередь необходимо оценивать эффективность работы конкретного продукта. При этом не должны быть упущены из внимания следующие важные факторы.

- ❑ Список вирусов в текущей базе — насколько он велик? Обеспечивает ли он защиту от известных в настоящее время вирусов?
- ❑ С появлением новых вирусов насколько оперативно производится обновление базы поставщиком? В день уведомления разработчика A\B о новом вирусе, на следующий день, через неделю?
- ❑ Каким образом производится доставка обновленных баз распознавания вирусов до потребителя? Обычная практика — это периодическая загрузка их с сайтов производителей A\B в Интернете. А если у пользователя нет к нему прямого или регулярного доступа? Возможна ли доставка обновлений региональным партнером, дистрибьютором?
- ❑ На каком этапе A\B может распознать вирус и предотвратить его распространение? Возможно ли это на межсетевом экране или на почтовом сервере, т. е. "на лету"? При копировании файлов с внешних носителей? Или только при целенаправленном сканировании диска?
- ❑ Какие ресурсы требует A\B от хоста, на котором будет установлен? Особенно от процессора и оперативной памяти?
- ❑ Как организована архитектура работы программы? Это отдельный программный модуль или сервер и агенты? Как организовано автоматизированное обновление распределенного программного обеспечения?

Не рекомендуем определять качество работы A\B по количеству наград от различных изданий, которые имеет данный продукт. Порой складывается впечатление, что данное печатное или электронное издание было специально учреждено для того, чтобы выдать награду именно этому продукту (анти-вирусы не исключение). Вместо этого, лучше обратиться к известным независимым экспертам, например ICSA Laboratory (бывшая <http://icsa.net>, ныне <http://www.icsalabs.com>) — рис. 23.1. При этом следует обратить внимание, что A\B проходят сертификацию по различным номинациям — анализируйте именно тот раздел, который собственно вам интересен.

Эффективность лучше всего оценивать по опыту работы хороших знакомых (особенно, если они мигрировали с одного A\B продукта на другой, интересно узнать причины). Кроме того, можно регулярно отслеживать активность Интернет-сайтов производителей A\B в зависимости от появления новых вирусов, скорость их реакции, выпуск обновлений, рекомендаций.



Рис. 23.1. Сертификат ICSA Lab

При приобретении необходимо учитывать не только первоначальную стоимость продукта, но и совокупную стоимость владения. Во сколько обойдутся обновления и гарантийная поддержка; приобретение новой гарантийной поддержки после истечения текущей; приобретение новых лицензий на пользователей при увеличении их числа?

Учитывая, что в настоящее время получили развитие компании, предлагающие весь спектр автоматизированных средств по безопасности, следует задуматься о перспективах и выбрать антивирусный продукт как компонент, который в будущем будет интегрирован с межсетевым экраном, системой обнаружения атак и прочими составляющими. В противном случае, рано или поздно, можно столкнуться с проблемой состыковки продуктов различных производителей.

Межсетевой экран (брандмауэр, firewall)



О межсетевых экранах (МСЭ) организация обычно задумывается в двух случаях:

- ☐ при появлении постоянного прямого выхода в Интернет, имеющего соединение с локальной сетью;
- ☐ при организации взаимодействия со своими удаленными филиалами в режиме он-лайн с использованием сети широкого доступа (выделенных или коммутируемых телефонных линий, беспроводного или спутникового канала и т. п.).

Классификация МСЭ



Для того чтобы определить, какой МСЭ необходим, нужно знать принципы работы конкретных моделей, на которые можно ориентироваться. Существуют различные классы МСЭ, выделим три наиболее часто встречающиеся в настоящее время.

Пакетный фильтр (packet filter, screening filter)

Пакетный фильтр в чистом виде — это устройство, которое фильтрует (пропускает или отклоняет) сетевые пакеты на основе predetermined данных о сетевых (IP) адресах источника или получателя. Однако в виде независимых программно-аппаратных комплексов МСЭ этого типа уже давно не встречаются в основном по причине недостаточной функциональности (например, легкости подмены адресов — спуффинга — для протоколов без установления соединений: IP, UDP). В настоящее время пакетным фильтром можно назвать *маршрутизатор с функциями МСЭ* этого уровня (англ. *firewall option pack, firewall feature set*). При этом, возможно, сам маршрутизатор выполняет более сложные функции — анализ пакетов на транспортном уровне с учетом портов обращений, *сокрытие внутренних адресов сети*, на границе которой он находится (англ. *Network Address Translation*), *сокрытие портов* обращений (англ. *Port Address Translation*), формирование групп доступа по сетевым адресам с более сложным разграничением доступа (отдельных адресов к отдельным службам), расширенные возможности аутентификации (поддержка протоколов RADIUS, TACACS), дифференцирование сетевых интерфейсов, как "опасный"/"безопасный" и т. п. Иногда в подобный модуль встраиваются дополнительные "узконаправленные" анализаторы для предотвращения отдельных классов атак, например, на отказ в обслуживании (ping-of-death, SYN-flood и т. д.).

Перечисленные возможности являются базовыми для современных МСЭ и встречаются и в двух других классах МСЭ. Однако для данного уровня (пакетный фильтр) функциональность анализа трафика ограничивается транспортным уровнем. Фильтр обрабатывает каждый очередной пакет совершенно независимо: он не знает, какие пакеты проходили по каналу до него (рис. 23.2). Вся информация, на основе которой принимается решение разрешения/отказа, извлекается только из исследуемого в данный момент пакета: из его сетевого и транспортного заголовков. Это позволяет свести задействованные МСЭ ресурсы оперативной памяти и процессора к минимуму, но не позволяет излагать в правилах фильтрации законы и логику более высокого уровня.

Нельзя однозначно отметить данный тип МСЭ как недостаточно функциональный. Во-первых, необходимо проанализировать, для каких задач используется выход в Интернет, возможно, учитывая более низкую стоимость таких МСЭ; его будет вполне достаточно для работы организации. Во-вторых, его можно использовать в качестве первой линии обороны, которая будет предотвращать наиболее грубые атаки. А уже между ним и локальной сетью устанавливать другой МСЭ, либо принципиальной иной класс защитных устройств (например, агента системы обнаружения атак, см. раздел "*Системы обнаружения атак*" данной главы), которые возьмут на себя анализ пакетов на более высоких уровнях сетевого стека. В этом случае пакетный фильтр снижает общую нагрузку на вычислительные ресурсы анализаторов более высокого уровня и может защищать их от большого класса атак на отказ в обслуживании.

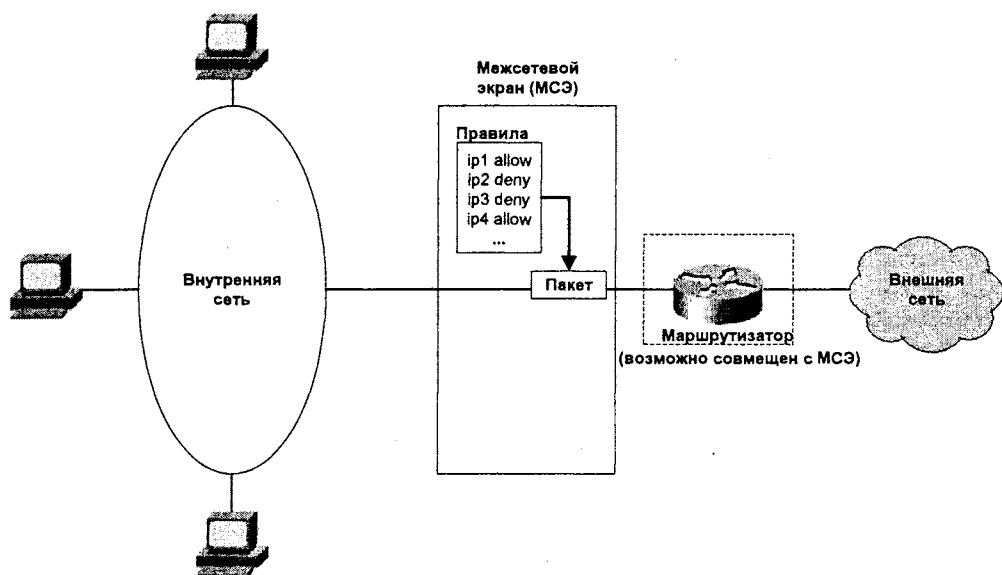


Рис. 23.2. МСЭ — пакетный фильтр

МСЭ с контролем соединения (virtual circuit control)

МСЭ данной категории производят более тонкий анализ проходящего трафика, а именно, рассматривают каждый пакет в принадлежности его к конкретному соединению, в том числе с учетом того, кем, когда и как было инициировано соединение и какая активность в рамках данного соединения была перед получением данного пакета. Причем, учитывая, что ряд протоколов работают без установки соединения (например, UDP), МСЭ производит работу с такими пакетами в рамках т. н. "виртуального" соединения — *потока* (англ. *flow*), и в этом случае рассматривая такие пакеты в едином контексте. При этом возможно использование информации не только о текущем соединении, но и предыдущих соединениях. Наименование современной технологии, использующей этот механизм, — *контроль состояния* (англ. *stateful inspection*) — рис. 23.3.

Обычно, для усиления контроля более высоких уровней сетевого стека в МСЭ данного типа используется дополнительный анализатор содержимого пакетов, исследующий активность данного соединения до уровня приложения. Таким образом, охваченным оказывается практически весь стек протоколов от сетевого до прикладного, что позволяет предположить высокую вероятность предотвращения любой известной на момент проектирования МСЭ атаки.

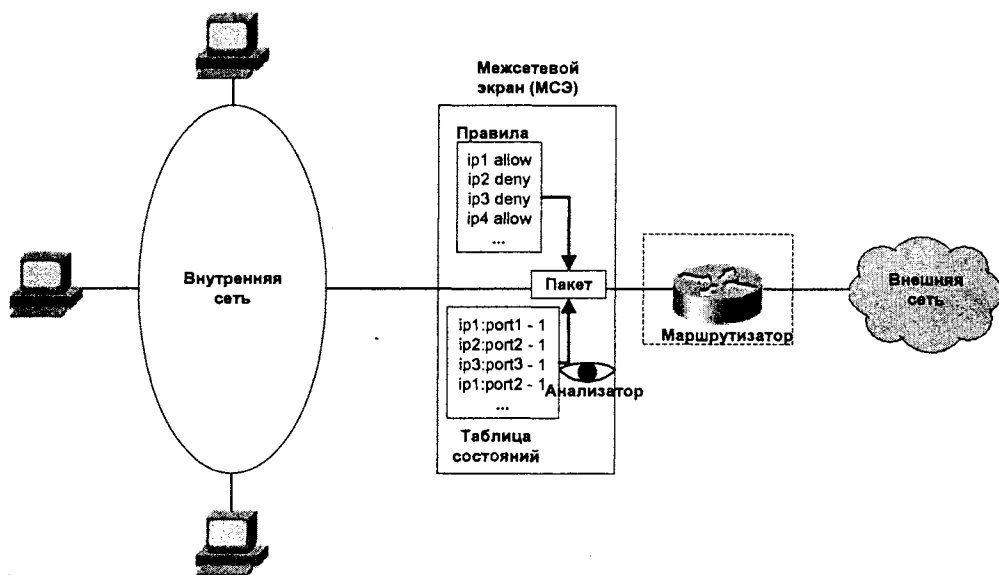


Рис. 23.3. МСЭ с контролем соединения

МСЭ — посредник приложения (application proxy, application layer gateway)

Технология посредника приложения заключается в том, что hosts во внутренней сети и hosts во внешней сети устанавливают соединения между собой не напрямую, а через виртуального "посредника" — отдельный сервис или демон в МСЭ, который общается с клиентом от имени сервера, а с сервером от имени клиента. Таким образом, МСЭ выступает как часть соединения и, соответственно, может анализировать все происходящие при соединении события, обнаруживая в том числе и возможные атаки (рис. 23.4).

Изначально прокси-службы разрабатывались исключительно для нескольких наиболее популярных протоколов прикладного уровня (HTTP, FTP и некоторых др.). В современных реализациях МСЭ данного класса есть возможность создавать посредников для собственных приложений, отличных от перечисленных. Общемировым стандартом подобного "универсального" прокси-сервиса стал протокол SOCKS (см. главу 22). Кроме того, в них также присутствуют все возможности пакетного фильтра.

Споры о том, какая из двух последних технологий лучше, продолжаются. Приверженцы обеих технологий говорят о достоинствах своей и недостатках противоположной. Считается, что оба принципа защиты достаточно надежны. Несовершенством МСЭ контроля соединения считается возможность прохода ряда атак на уровне приложений. Недостатком посредника приложений — замедление прохождения трафика за счет установления двойного соединения к посреднику.

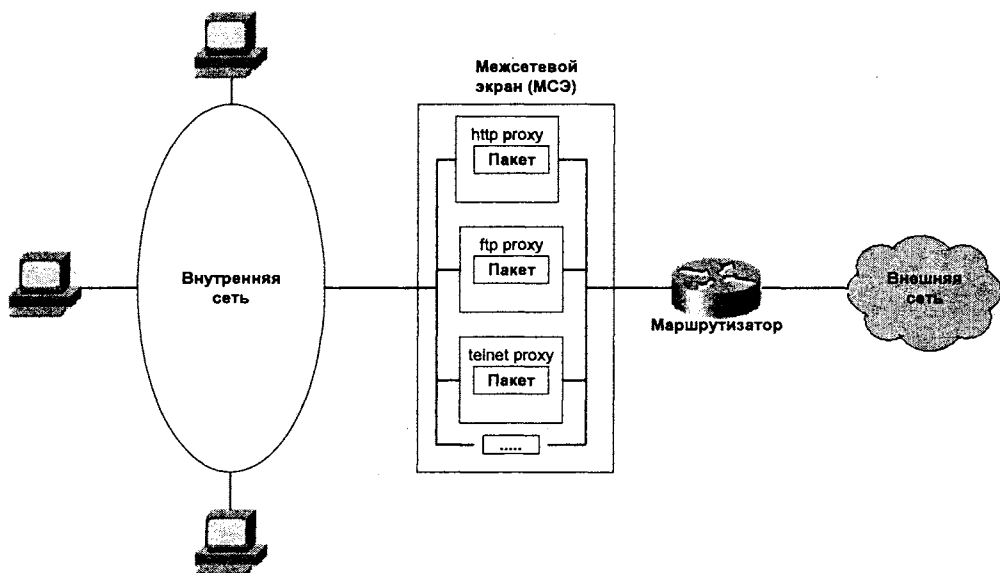


Рис. 23.4. МСЭ — посредник приложений

Если обратиться к независимому авторитетному источнику RFC 2196 (Site Security Handbook) [RFC-2196], то там можно встретить следующую рекомендацию: *"The current best effort in firewall techniques is found using a combination of a pair of screening routers with one or more proxy servers on a network between the two routers"*. ("Текущим наиболее мощным вариантом в технологии МСЭ считается использование комбинации двух фильтрующих трафик маршрутизаторов с одним или более сервером-посредником в сегменте сети между этими двумя маршрутизаторами"). Подобная схема расположения прокси-сервиса нашла реализацию в идеологии "демилитаризованной зоны", о которой будет рассказано в разделе "Демилитаризованная зона" данной главы.

В любом случае перед выбором конкретной модели следует изучить последние имеющиеся ошибки и уязвимости в ее функционировании (по информации на <http://www.cert.org/>, <http://www.securityfocus.com/>) и сравнить с другими моделями МСЭ, представленными на рынке.

Распределенные или персональные межсетевые экраны



Рассматривая технологии межсетевых экранов, необходимо упомянуть встречающиеся термины *распределенный*, или *персональный*, межсетевой экран (англ. *distributed firewall*, *personal firewall*, *desktop firewall*). Технология персонального МСЭ заключается в том, что вся функциональность фильтрации, или "проксирования", соединений выполняется модулем, расположенным непосредственно на персональном компьютере. Чаще всего модуль

реализуется в виде прослойки, модифицирующей системную очередь обработки сетевых пакетов. Персональный МСЭ имеет возможность запретить любой пакет на свое усмотрение в момент, когда тот передается приложением на отправку в сеть, либо в момент приема из локальной сети.

Выгоды такого подхода представим на следующем примере. Если одному из хостов локальной сети необходимо общаться с хостом во внешней сети по протоколу Telnet, то разрешение работы Telnet придется прописывать на общем межсетевом экране. Таким образом, потенциально в работу Telnet могут быть вовлечены все хосты локальной сети, а при появлении ошибки в данном сервисе на межсетевом экране все хосты сети подвергаются угрозе. При наличии персональных экранов каждый хост получает индивидуальную защиту, не зависимую от других хостов.

Отрицательными аспектами технологии являются:

- ❑ возможность влияния на функционирование МСЭ со стороны пользователя, работающего на данной персональной ЭВМ;
- ❑ более высокая стоимость владения, так как помимо закупочной цены необходимо еще учитывать стоимость распределенной поддержки (инсталляции, настройки, обновления и т. д.).

Однако для хостов, на которых размещаются информационные активы повышенной важности, можно задуматься о дополнительном усилении защиты за счет персональных экранов.

Интересное дополнительное свойство можно наблюдать у некоторых персональных экранов, которые имеют адаптивный пользовательский интерфейс настройки правил доступа. В случаях, когда такой экран сталкивается с сетевой активностью, предварительных правил фильтрации которой у экрана нет, он предлагает пользователю самому определить данную активность как разрешенную или несанкционированную для данного хоста. Таким образом, настраивая правила (либо, наоборот, не настраивая их достаточно тонко), можно получать представление о необычной сетевой активности, при этом персональный МСЭ проявляет функциональность агента системы обнаружения атак. Авторы хотели бы предостеречь против включения адаптивного режима настройки доступа на пользовательских станциях, так как рядовой сотрудник предприятия скорее всего не сможет по IP-адресу и номеру порта соединения принять решение о характере поступившего трафика. А вот активизация этого режима на компьютерах сотрудников сетевых служб и на серверах вполне может быть оправдана.

Демилитаризованная зона



Современное развитие бизнеса предполагает, что внутренние ресурсы организации не должны быть полностью закрыты. Ряд хостов, таких как Веб-сервер, FTP-сервер, почтовый сервер, должны быть в той или иной степени доступны для внешних пользователей, в том числе для тех, о ком нет ника-

кой предварительной информации. Где следует размещать такие хосты? Если во внешней сети, перед межсетевым экраном, это значит, что их защищенность будет зависеть только от схемы безопасности операционной системы и приложения, что, как показывает опыт, недостаточно. Если разместить их во внутренней сети, за межсетевым экраном, то тогда придется пропускать внешних пользователей во внутреннюю сеть, а это всегда небезопасно, даже при точной настройке правил доступа. Вполне логично напрашивается вывод — создать для подобных ресурсов отдельную подсеть, свободную от элементов внутренней и внешней сети. Данная технология получила название *демилитаризованной зоны (ДМЗ)* (англ. *demilitarized zone — DMZ*, или *screening subnet*).

Поскольку обычно межсетевые экраны имеют по два сетевых интерфейса (один во внутреннюю и один во внешнюю сети), то для ДМЗ необходим третий сетевой интерфейс. Отдельные правила, прописанные на межсетевом экране для доступа в ДМЗ, позволят, с одной стороны, обеспечить защиту информационных активов широкого доступа, а с другой стороны, не предоставят дополнительного доступа в локальную сеть (рис. 23.5). Очевидно, что в этом же сегменте наиболее логично располагать сервера прокси-служб. В этом случае мы получаем ту самую схему, описанную в [RFC-2196], с двумя фильтрующими МСЭ (их правила доступа объединены в одном физическом МСЭ на разных сетевых интерфейсах): "до" и "после" прокси-сервера.

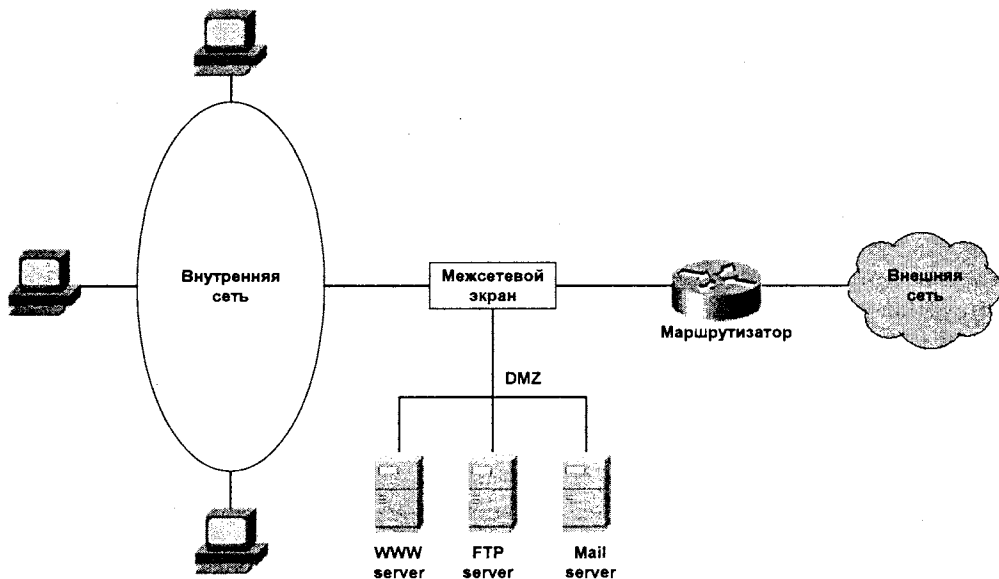


Рис. 23.5. Демилитаризованная зона

Транслятор сетевых адресов (NAT), транслятор портов



Говоря о современных межсетевых экранах, невозможно не упомянуть функцию *трансляции сетевых адресов* (англ. *network address translation — NAT*). Данная схема выполняет две полезные функции:

- ❑ актуальное для безопасности сокрытие схемы внутренней адресации локальной сети и частичную анонимность отправителя пакета;
- ❑ актуальное для сетевых технологий преобразование внутренних, т. н. "немаршрутизируемых" приватных IP-адресов (англ. *private network IP-address*) в разрешенный внешний интернет-адрес или адреса.

Технология заключается в том, что на МСЭ, который в данном случае играет роль шлюза, при выходе во внешнюю сеть во всех сетевых пакетах производится подмена внутреннего адреса на predeterminedный внешний адрес. При этом шлюз ведет таблицу соответствия отправленных пакетов таким образом, что для входящих пакетов из внешней сети производится обратная замена внешнего адреса на внутренний (рис. 23.6).

Различают *статический NAT*, когда за одним адресом внутренней сети жестко закреплен один внешний адрес, и более часто используемый *динамический NAT*, когда за множеством внутренних адресов закреплен некоторый, обычно небольшой, диапазон внешних адресов (а, возможно, и всего один IP-адрес). В случае динамического NAT возникает необходимость подмены на шлюзе кроме IP-адреса еще и номера порта соединения, так как потенциально два различных хоста из внутренней сети могут "выразить желание" общаться с одинаковым портом с одним и тем же удаленным объектом. Для данной технологии можно встретить термин *трансляция портов* (англ. *port address translation — PAT*), хотя обычно считается, что термин NAT уже содержит в себе эту функциональность.

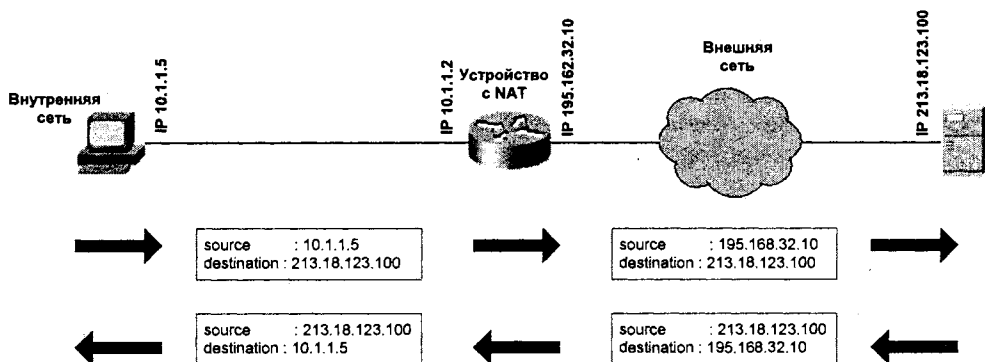


Рис. 23.6. Схема NAT

Правила доступа, листы доступа



Используемые термины *лист доступа* (англ. *access control list — ACL*) и *правило доступа* (англ. *access rule*) отображают основу настройки МСЭ, сценарий его поведения при получении пакета.

Обычно правило включает в себя:

- ☐ направление входа для данного пакета (номер или название сетевого интерфейса, с которого поступил пакет);
- ☐ направление выхода из шлюза (номер или название сетевого интерфейса, на который направляется пакет);
- ☐ адрес или группу принадлежности (группу адресов), куда отнесен источник, породивший пакет;
- ☐ адрес или группу принадлежности (группу адресов), куда отнесен получатель пакета;
- ☐ протокол или порт службы, от которой пришел пакет;
- ☐ протокол или порт службы, которой адресован пакет;
- ☐ набор действий, которые необходимо предпринять для данного пакета (пропустить, отвергнуть, переслать другой службе, проанализировать дополнительно).

Набор правил, возможно, с дополнительным определением интерфейсов, групп и ряда других параметров, составляют *лист доступа*.

Альтернативный доступ



Под альтернативным доступом понимается другой канал доступа в локальную сеть, кроме Интернета, например, по коммутируемым или выделенным линиям. В этом случае возможны два варианта работы.

1. Альтернативное подключение происходит также через МСЭ. В этом случае необходимо позаботиться о дополнительном оборудовании, например, сервере доступа. В зависимости от задач, возлагаемых на функциональность данного подключения, возможно, следует задуматься также об усилении безопасности такого соединения за счет использования специальных протоколов безопасности.
2. Подключение происходит в обход МСЭ, напрямую к одному из хостов локальной сети, например, используя персональный модем. Необходимо учесть, что если на хосте, к которому имеется альтернативное подключение, не предпринять дополнительных мер безопасности, то работа основного МСЭ может быть сведена на нет, так же как, проводя аналогии, при наличии мощной бронированной двери не производить установки решеток на окна. В качестве дополнительных мер в этой ситуации возможны и установка персонального МСЭ на данный хост, и включение

модема в сеть только на время работы (приема или отправки корреспонденции с дополнительной ее проверкой). Но в любом случае, этот вариант является выпадением из общей политики безопасности со всеми вытекающими последствиями. Если таких альтернативных подключений в организации существует несколько, возможно, следует задуматься о приобретении второго МСЭ для создания единого портала, например, коммутируемых подключений.

Уплотнение операционной системы



Безопасность работы МСЭ может зависеть не только от того, как работает сам экран, но и от безопасности операционной системы, на которой он установлен. Поскольку ряд операционных систем в той или иной мере уязвим, а, возможно, новые уязвимости будут обнаружены в будущем, качественно реализованный МСЭ должен при установке произвести *уплотнение операционной системы*. Уплотнение заключается в том, что все сервисы операционной системы, которые не требуются для работы МСЭ, отключаются, а ряд родных для операционной системы утилит, сервисов или демонов заменяются на соответствующие утилиты, сервисы и демоны МСЭ, которые функционируют более строгим в плане информационной безопасности или защищенным образом.

Бастион хост



Так называется любой хост, который находится под возможностью прямой атаки злоумышленником, без дополнительных устройств защиты между данным хостом и хостом злоумышленника. Обычно бастион хостом (англ. *bastion host*) является сам МСЭ. Однако в ряде случаев (например, при отсутствии МСЭ) бастион хостом могут выступать и другие хосты сети, например WWW-сервер, DNS-сервер, Mail-сервер. В этом случае обычный хост должен быть превращен в бастион, обычно за счет уплотнения операционной системы. На нем должны быть отключены сервисы, не являющиеся необходимыми, закрыты нефункциональные порты, не должно быть ресурсов, открывающих привилегированный доступ во внутреннюю сеть.

Приобретение МСЭ



Основным рекомендациям по выбору конкретного автоматизированного средства безопасности будет посвящена *глава 24*, а в этой главе хотелось бы коснуться вопросов, специфичных именно для МСЭ.

Для того чтобы полностью подготовиться к выбору того или иного МСЭ, необходимо учесть еще ряд важных моментов.

- ❑ Иногда качественные, защищенные МСЭ взламываются по причине ошибочной конфигурации. А это означает, что вместе с приобретением МСЭ необходимо "приобретать" и обучение администратора (или администраторов) МСЭ, а также гарантийную и техническую поддержку от

фирмы-производителя или локального поставщика, возможно, с прописыванием в договоре ответственности за инциденты, произошедшие по вине некачественного функционирования продукта.

- ❑ Если предполагается активный доступ работников во внутреннюю сеть организации извне, например, для работы из дома или из командировок, возможно, надо приобретать МСЭ с функциями VPN-сервера (о нем пойдет речь в *разд. "Технология VPN" данной главы*), который обеспечит защиту удаленного соединения.
- ❑ Для больших организаций необходимо подсчитать количество лицензий на пользователей (хосты), которые будут использовать МСЭ для работы. Причем рассчитывать эту величину надо с учетом перспективы роста количества пользователей. Необходимо ознакомиться с лицензионной политикой компании-производителя МСЭ, возможно, приобретение большого количества лицензий предусматривает дополнительные льготы, например приобретение VPN-сервера, бесплатных VPN-клиентов или персональные экраны и т. д.
- ❑ Необходимо подсчитать требуемое количество сетевых интерфейсов и другого оборудования с учетом ДМЗ, серверов доступа, модемных пулов и т. п.
- ❑ Следует предусмотреть интеграцию МСЭ и сервера антивирусного контроля (если таковой имеется), возможность управления с МСЭ маршрутизаторами и возможность управления экраном от системы обнаружения атак и т. д.

Что касается технической части МСЭ, то, кроме собственно качественно работающей функциональности, требуется учитывать:

- ❑ простой и понятный пользовательский интерфейс администратора, включающий возможность недопонимания настроек;
- ❑ правила по умолчанию (обычно запрещено все то, что явно не разрешено), порядок их применения и конфликтующие правила (поведение МСЭ в ситуации, когда сначала идет разрешающее, а затем запрещающее правила и наоборот);
- ❑ отчеты, выдаваемые МСЭ (их подробность, простота и четкость изложения);
- ❑ возможность фильтрации на прикладном уровне некоторых популярных протоколов (HTTP, FTP, SMTP, POP, IMAP), ведь с ростом числа пользователей Интернета в организации становится трудно контролировать использование данного ресурса сотрудниками предприятия, соответственно, может возникать необходимость запрета только ряда сайтов (URL) — игровых, порнографических и т. п.

В заключение необходимо отметить, что должность администратора МСЭ порой настолько важна, что выделяется многими организациями (особенно на Западе) в отдельную штатную единицу.

Технология VPN



В разделе "Приобретение МСЭ" настоящей главы были упомянуты термины VPN-сервер и VPN-клиент (VPN — *virtual private network* — *виртуальная частная сеть*). Этими терминами обозначают участников защищенного соединения, при этом предполагается, что механизмы защиты применяются на сетевом уровне, где в качестве транспорта используется протокол IP. Если речь идет о применении протоколов или схем безопасности более высоких уровней (транспортный, прикладной), то данный термин обычно не используется.

Современные технологии VPN ориентируются на использование стандартных протоколов безопасности (IPSec, PPTP, L2TP, L2P — см. главу 22), но конкретные производители могут предоставлять и схемы защиты собственной разработки. Сама технология заключается в применении криптографических методов для обеспечения конфиденциальности и целостности данных, пересылаемых между клиентом и сервером. Два основополагающих признака технологии VPN.

- ❑ Средой передачи данных обычно служат сети общего пользования, такие как Интернет, городская телефонная сеть и корпоративная сеть без дополнительных механизмов защиты (например, аппаратных).
- ❑ Криптографические механизмы накладываются на третьем (сетевом) уровне модели OSI или между третьим и вторым уровнем. Это создает у пользователя иллюзию изолированности от подавляющего большинства узлов сети общего пользования и создания "внутри нее" *виртуальной сети* из нескольких компьютеров, которые владеют одинаковыми VPN-средствами с одинаковыми криптографическими ключами.

Различают программные комплексы, где безопасность осуществляется только за счет усиления соответствующего сервиса или демона (например, ipd), и программно-аппаратные, примерами которых могут быть маршрутизаторы с функцией VPN или хосты с дополнительными платами безопасности. При разговоре о технологии следует дифференцировать две возможные схемы применения механизмов защиты.

- ❑ Схема "сеть-сеть", когда протоколы безопасности применяются только к пакетам, выходящим из локальной сети, и прекращают свое действие при входе пакета в удаленную локальную сеть (если обмен данных идет между двумя хостами в двух локальных сетях, например, в двух филиалах одной организации). В этом случае необходимо учитывать, что внутри локальных сетей пакеты не защищены.
- ❑ Схема "точка-сеть" — обычно используется при удаленной работе сотрудника с сетью организации. При этом как типовой вариант предполагается, что клиент (например, с мобильного компьютера по модемной линии) подключается к серверу удаленного доступа, связь между которым и локальной сетью назначения идет через компьютерную сеть общего пользования. В этом случае необходимо учесть, что в ряде протоколов механиз-

мы VPN активизируются только между сервером доступа и локальной сетью назначения — т. е. трафик, идущий между сервером доступа и удаленной станцией (ноутбуком), может быть не защищен.

Технология VLAN



Virtual Local Area Network — это псевдосеть, организованная на базе существующей локальной сети. На одной физической локальной сети возможно организовать несколько виртуальных сетей — VLAN, при этом субъектам данной виртуальной сети (хостам, пользователям) не будут видны участники других виртуальных сетей, как если бы данная виртуальная сеть и была единственной сетью, т. е. собственно локальной сетью.

Средством, позволяющим создавать VLAN, является коммутатор, поддерживающий соответствующие механизмы. В зависимости от конкретной реализации коммутатора, могут быть реализованы следующие типы виртуальных сетей.

- ❑ На основе портов коммутатора или, фактически, сетевых сегментов. В этом случае на коммутаторе указывается принадлежность того или иного порта к данному VLAN. Таким образом, устройства, присоединенные к порту коммутатора, смогут взаимодействовать только с устройствами, присоединенными к тем портам, которые объединены в данную виртуальную сеть.
- ❑ На основе MAC-адресов или групп физических устройств. На коммутаторе указываются адреса сетевых карт — сетевых интерфейсов (MAC-адреса), и коммутатор проверяет поступающие на него кадры и направляет их только тем устройствам, чьи MAC-адреса прописаны для данного VLAN.
- ❑ На основе сетевого протокола или групп логических устройств. При этом на коммутаторе указываются адреса или номера подсетей (для хостов — IP подсеть) и коммутатор, в данном случае, работает, практически, как маршрутизатор.
- ❑ На основе типов протоколов. Коммутатору указываются инструкции, в каких полях кадров искать указания на протоколы и объединять данные VLAN по принципу принадлежности к протоколам (например, VLAN для DecNet и VLAN для NetBIOS).
- ❑ На основе комбинации критериев или на основе правил. В этом случае для создания VLAN могут быть использованы комбинации перечисленных вариантов, например, совмещением MAC-адреса, адреса подсетей и типа протоколов.
- ❑ На основе тегов или IEEE 802.1Q. К Ethernet-кадру добавляется идентификатор принадлежности к тому или иному VLAN, а коммутатор производит сортировку кадров на основе анализа этих идентификаторов.

- ❑ На основе аутентификации пользователей. В этом случае коммутаторы будут работать как межсетевые экраны, требуя у пользователей аутентификационных данных (имя/пароль) и производя фактически группировку пользователей и ресурсов, а не устройств.

Криптографические средства



Несмотря на возможность сильной криптографической защиты информации при пересылке по открытым каналам связи (которой была посвящена *Участь*) в организации остаются обычно еще задачи сохранения информации в защищенном виде на рабочих станциях пользователей и серверах, а также обеспечение аутентифицированности информационных единиц (например, целостность и неотрекаемость почтовых сообщений). Эти задачи вполне эффективно можно решить с помощью соответствующих приложений — криптографических программ. Хотя существуют и другие способы.

На первом этапе необходимо определиться, какая из задач является первоочередной:

- ❑ обеспечение конфиденциальности данных на месте хранения или при пересылке;
- ❑ защита целостности данных на месте хранения или при пересылке;
- ❑ обеспечение неотрекаемости данных при хранении или пересылке.

Возможно, придется решать все задачи в комплексе и, соответственно, ориентироваться на одно средство, которое решало бы все задачи, а не плодить дополнительные сущности, которые затем трудно поддерживать и интегрировать.

Учитывая, что при этом, возможно, придется оперировать созданием, хранением и распространением цифровых сертификатов или открытых ключей, следует обратить особое внимание на технологию PKI (Public Key Infrastructure — инфраструктура открытых ключей) и, соответственно, ориентироваться на средства, использующие данную технологию.

Имеющиеся в настоящее время средства криптографической защиты можно разделить на следующие категории.

- ❑ Средства, требующие установки дополнительного аппаратного обеспечения в виде плат, активаторов портов и т. д. В обмен на некоторую громоздкость решения предлагают одно значительное преимущество — возможность выноса идентификатора-аутентификатора на внешний носитель (токен). Теоретически внешний носитель в виде дискеты можно предусмотреть и в других средствах, но дискетные варианты имеют отрицательную черту в виде возможности копирования.
- ❑ Средства защиты, предоставляемые операционной системой. Обычно это встроенная файловая система с возможностью криптографической

защиты. Представляется достаточно мощным средством, при одном существенно отрицательном аспекте. Поскольку операционная система — это сложный механизм с предоставлением пользователям и других возможностей, кроме безопасности (в частности, обратную совместимость с предыдущими версиями, возможно, имеющими более слабую схему безопасности), то уязвимость или компрометация одного из элементов операционной системы может привести к возможности взлома криптографической защиты.

- ❑ Средства расширения функциональности (англ. *Application Programming Interface — API*), имеющиеся в операционной системе или в *дополнительных утилитах* системы (англ. *option pack, development kit* и т. д.). В отличие от предыдущего варианта предполагает возможность включения разработчиками дополнительных контрольных механизмов, позволяющих снизить возможность компрометации взлома средств защиты. Однако данный вариант требует наличия квалифицированного программиста-разработчика.
- ❑ Средства собственной разработки. Требуют от разработчика серьезного математического образования и подробного знакомства с современными криптографическими алгоритмами (при этом следует учесть, что опубликование криптографического алгоритма не всегда обязательно означает возможность его свободного использования для бизнес-целей организации — возможно нарушение авторских прав создателей алгоритма). Кроме того, при собственных разработках следует очень тщательно контролировать работу программ на низком уровне — например, работу с памятью, когда работа с криптографическими ключами и паролями должна производиться в защищенных областях памяти, исключающих возможность несанкционированного копирования данных в файл (*дамп памяти*), необходимо помнить о необходимости очистки памяти и т. д.
- ❑ Средства, встроенные в коммерческие приложения. Например, некоторые почтовые системы имеют встроенные возможности шифрования сообщений и установки электронно-цифровой подписи. Положительной стороной является отсутствие необходимости дополнительных инвестиций (если почтовая система уже приобретена или планируется к приобретению вне зависимости от требований безопасности). Отрицательной стороной может быть ограниченная функциональность (в данном случае нет возможности защиты других данных, кроме почтовых сообщений). Кроме того, функциональность безопасности также может присутствовать не полностью в связи с ограничениями на вывоз криптографических средств из страны производителя.
- ❑ Свободно распространяемые средства. Данные средства достаточно легко найти в Интернете, однако при их использовании следует учитывать следующие вопросы. Свободное использование программного обеспечения для частного лица не означает автоматического разрешения на свободное

использование программного обеспечения организацией, особенно если это использование связано с получением прибыли. За ошибки и сбои, возникшие при использовании свободно распространяемых средств, разработчик обычно ответственности не несет (мы не будем рассматривать крайние примеры, когда в такие программы специально встраивают троянские программы, программные закладки и т. д.). Кроме того, иногда для крупной компании использование свободно распространяемого программного обеспечения не всегда идет на пользу престижу.

- ❑ **Коммерческие программные средства.** В зависимости от конкретной реализации могут предоставить весь спектр преимуществ и избежать отрицательных черт, за исключением, конечно, стоимости. В качестве дополнительных плюсов можно рассмотреть регулярное обновление версии (а следовательно, усиление защиты) при гарантийной поддержке, а также возможность апелляций в случае сбоев/взломов.

На использование криптографических средств защиты стоит обращать внимание также при приобретении любых продуктов информационных технологий, например, СУБД (см. позицию *"Средства, встроенные в коммерческие приложения"* приведенного списка).

При выборе криптографических средств защиты следует учитывать используемые алгоритмы. Во-первых, на предмет надежности работы самого алгоритма. Не всегда закрытость алгоритма означает его стойкость. Чаше наоборот — опубликованные и публично протестированные алгоритмы оказываются гораздо более устойчивыми к взлому, чем засекреченные алгоритмы, где авторами могла быть допущена ошибка, которая не была выявлена тестированием сторонних экспертов. Во-вторых, криптографический алгоритм (возможно, с конкретной длиной ключа) могут быть объектами запретительных акций при вывозе за пределы государства разработчика. Вне зависимости от алгоритма, пристальное внимание должно быть уделено длине ключа, ведь малая длина ключа может быть преодолена простым прямым перебором, вне зависимости от теоретической стойкости самого алгоритма. В настоящее время разговоры о ключах шифрования следует вести от 128 бит для симметричных алгоритмов и от 1024 бит для асимметричных. Для оценки качества алгоритма проще всего обратиться к исследованиям независимых экспертов, которые доступны в публичной печати.

Сканеры уязвимости



Ряд компаний-производителей средств информационной безопасности относят сканеры уязвимостей к элементам системы обнаружения и предотвращения атак, однако, на наш взгляд, их следует выделить в отдельный раздел.

В свете огромного количества определений будем называть *уязвимостью* характеристику информационной системы, использование которой злоумыш-

ленником (которым может быть любой субъект сети) преднамеренно или непреднамеренно может привести к реализации угрозы.

Существуют различные классификации уязвимостей. Например, в [Лукацкий1] уязвимости классифицируются как уязвимости проектирования, реализации, эксплуатации. Возможно классифицировать их по сфере применения — сетевые, операционной системы, базы данных, приложения. Для обеспечения единого языка обсуждения уязвимостей среди профессионалов принят стандарт их обозначения Common Vulnerabilities and Exposures (CVE), ознакомиться с существующими идентифицированными уязвимостями и кандидатами в список можно на сайте <http://cve.mitre.org/cve/>.

Использовать тот или иной сканер означает определить, насколько уязвима информационная система, которая подвергается сканированию. Именно поэтому такие сканеры — это инструменты следующих категорий пользователей:

- ☐ специалистов по безопасности, которые хотят регулярно проверять уровень уязвимости (либо наоборот, защищенности) информационных систем своей организации;
- ☐ специалистов по сертификации информационных систем с точки зрения информационной безопасности;
- ☐ организаций, предоставляющих услуги по анализу защищенности информационных систем;
- ☐ злоумышленников, использующих сканеры поиска основы для реализации атаки.

Сами продукты могут быть свободно распространяемыми или коммерческими. Среди коммерческих продуктов можно найти продукты с бесплатным ограниченным сроком использования. Третий вариант — обратиться к компании, которая произведет однократное сканирование информационных систем организации за сумму гораздо меньшую, чем стоимость приобретения и поддержки сканера как собственности организации. Необходимо только учесть, что различные продукты могут работать на всевозможных операционных платформах и анализировать только ограниченный класс информационных систем.

Решение использовать или не использовать сканер зависит от задач и средств организации. Безусловно, применение сканеров полезно и способствует усилению информационной безопасности предприятия. Однако не следует ожидать от сканера гениальных технических решений и выработки сложных схем безопасности. Большинство рекомендаций, выданных в отчете, после анализа информационной системы скорее всего сведется к необходимости установки программной заплатки от производителя информационной системы. А, скажем, обнаружение установленного модема, возможно, будет проклассифицировано как серьезная угроза с предложением убрать

данный модем, без учета необходимости его использования в организации и без дополнительных рекомендаций по усилению безопасности.

Кроме того, формирование баз уязвимостей, их тестирование и распространение производителем по заказчикам требует определенного времени, а, следовательно, всегда запаздывает по отношению к новым уязвимостям, обнаруженным злоумышленниками. Если, скажем, информация о новой уязвимости на он-лайн службах (таких как <http://www.cert.org>) появляется в течение нескольких дней, иногда даже часов, то обновление баз уязвимостей сканера может занять недели.

Тем не менее, полученный отчет обычно дает представление о статистике имеющихся уязвимостей по категориям (очень опасные, средние, мало опасные) и соответственно о необходимости мероприятий по усилению эффективности работ по информационной безопасности.

Следует также учесть, что некоторые операционные системы (Windows 2000) уже несут в себе некоторую функциональность сканера. Скажем, анализ политики безопасности (Security Configuration and Analysis), выбранный для наиболее строгой политики (hisecdc.inf), выдаст отчет по обнаруженным уязвимостям в настройке операционной системы, весьма схожий с соответствующим отчетом сканера уязвимости.

В любом случае перед приобретением конкретной модели стоит получить у производителя временную, или ограниченную, копию для экспериментов.

Системы обнаружения атак



Собственно полное название таких систем — это *системы обнаружения и предотвращения атак*, так как именно в возможности автоматизированного противодействия атакам заключается одно из основных преимуществ таких систем, по сравнению, например, со средствами, основанными на человеческом факторе. Однако мы будем продолжать использовать устоявшееся название — *система обнаружения атак* (COA), англ. — *intrusion detection system (IDS)*.

Принцип работы COA заключается в постоянном анализе (в режиме реального времени) активности, происходящей в информационной системе, и при обнаружении подозрительного информационного потока предпринимать действия по его предотвращению и информированию уполномоченных субъектов системы.

COA используют различные механизмы в своей работе. Приведем несколько схем их функционирования по классификации корпорации Cisco Systems, Inc.

Технология сравнения с образцами

Анализируется наличие в пакете некоторой фиксированной последовательности байтов — шаблона (англ. *pattern*) или сигнатуры (англ. *attack signature*). Обычно производится анализ пакетов, предназначенных определенному сервису (порту), таким образом, снижается количество пакетов и шаблонов для анализа. Соответственно, пакеты, предназначенные нестандартным портам или пакеты, непредопределенных протоколов, могут не подлежать анализу и пропускаться.

Рассмотрим пример работы механизма.

Если пакет сетевого протокола IPv4, транспортного протокола TCP, предназначен порту 2222 и содержит в разделе данных строку "foo", то это считается атакой.

Можно указывать порты отправителя и получателя и отдельные флаги. Это, безусловно, упрощает работу механизма, но оставляет его несколько примитивным.

Положительные стороны:

- ☐ наиболее простой метод обнаружения атак;
- ☐ позволяет жестко увязать образец с атакой;
- ☐ сообщение об атаке достоверно (если образец верно определен);
- ☐ применим для всех протоколов.

Отрицательные стороны:

- ☐ если образец определен слишком общё, то вероятен высокий процент ложных срабатываний;
- ☐ если атака нестандартная, то она может быть пропущена;
- ☐ для одной атаки, возможно, придется создавать несколько образцов;
- ☐ метод ограничен анализом одного пакета и, как следствие, не улавливает тенденций и развития атаки.

Технология соответствия состояния

Поскольку сетевое взаимодействие, в том числе и атака, — это большее, чем единичный пакет, то данный метод работает с потоком данных, а не с отдельным пакетом. Производится проверка ряда пакетов из каждого текущего соединения, прежде чем принимается решение о наличии или отсутствии атаки.

Если проводить сравнение с предыдущим методом, то можно привести пример, когда строка "foo" направляется атакуемому в двух пакетах, как "fo" и "o". В этом случае первый метод пропустит атаку, а метод контроля потока пакетов выявит ее.

Положительные стороны:

- ☐ в применении метод лишь ненамного сложнее метода сравнения с образцами;
- ☐ позволяет жестко увязать образец с атакой;
- ☐ сообщение об атаке достоверно (если образец верно определен);
- ☐ применим для всех протоколов;
- ☐ уклонение от атаки более сложно (по сравнению с методом сравнения с образцами).

Отрицательные стороны:

- ☐ если образец определен в слишком общем виде, то вероятен высокий процент ложных срабатываний;
- ☐ если атака нестандартная, то она может быть пропущена;
- ☐ для одной атаки, возможно, придется создавать несколько образцов.

Анализ с расшифровкой протокола

В этом случае метод предполагает анализ атак применительно к отдельным протоколам с учетом их специфики. После того, как вид протокола определен, применяются правила существующих стандартов для протокола с учетом обнаружения несоответствия этим правилам, таких, например, как значений специфических полей протокола, длины полей, числа аргументов и т. п.

Если рассматривая наш пример, предположить, что поверх протокола TCP будет работать протокол, в котором есть поле переменной длины, а следом за ним поле, одним из значений которого может быть параметр "fool", то при использовании методов сравнения с образцами и соответствия состояния мы получим вариант ложного срабатывания, причем, учитывая переменную длину первого поля, затруднительно будет настроить правило, которое пропускало бы такие пакеты без срабатывания.

В качестве примера пропуска реальной атаки можно привести случай, когда гипотетический протокол допускает игнорирование значений NULL, тогда строка ('f' 00₁₆ 'o' 00₁₆ 'o' 00₁₆) будет пропущена в двух предыдущих методах, но обнаружена в анализе с расшифровкой протокола, который произведет удаление NULL-значений.

Положительные стороны:

- ☐ снижает вероятность ложных срабатываний, если протокол точно определен;
- ☐ позволяет жестко увязать образец с атакой;
- ☐ позволяет улавливать различные варианты на основе одной атаки;

- ☐ позволяет обнаружить случаи нарушения правил работы с протоколами.

Отрицательные стороны:

- ☐ если стандарт протокола допускает разночтения, то вероятен высокий процент ложных срабатываний;
- ☐ метод сложен для настройки.

Статистический анализ

Использует алгоритмизированную логику для определения атаки. Применяются статистические данные для оценки трафика. Примером выявления такой атаки будет распознавание сканирования портов. Для правила определяется предельное значение портов, которые могут быть использованы на одном хосте (возможно, и с одного хоста). В этом случае одиночные легальные соединения дадут в сумме проявление атаки.

Положительные стороны:

- ☐ некоторые типы атак могут быть обнаружены только этим методом.

Отрицательные стороны:

- ☐ алгоритмы распознавания могут потребовать тонкой дополнительной настройки.

Анализ на основе аномалий

Предназначен не для четкого выявления атак, а для определения подозрительной активности, отличающейся от нормальной. Основная проблема метода заключается в том, чтобы определить критерий нормальной активности. Необходимо также установить допустимые отклонения от нормального трафика, которые еще не будут считаться атакой.

Подкатегорией такого метода будет анализ на основе профилей, когда нормальное поведение определяется для отдельных субъектов (пользователей/систем).

Иногда элементы такого анализа встречаются и в других методах, скажем, в расшифровке протокола, когда выявлен элемент, не принадлежащий предопределенным протоколам или нарушающий правила использования протоколов.

Системы, основанные на методе аномалий, считаются достаточно перспективными, но еще развивающимися и находящимися в стадии исследования.

Положительные стороны:

корректно настроенный анализатор позволит выявлять даже неизвестные атаки, не потребует дополнительной работы по вводу новых сигнатур и правил атак.

Отрицательные стороны:

- ❑ не может представить описание атаки по элементам, скорее сообщает, что происходит что-либо подозрительное;
- ❑ отношение полезной информации (на основе которой делаются выводы) к бесполезной очень невелико в большинстве случаев;
- ❑ значительно зависит от среды функционирования как определяющего фактора нормального поведения

Рассмотренная классификация достаточно подробна, иногда производят разделение только на два типа противоположных методов (и, соответственно, систем):

- ❑ на основе сигнатур (образцов и правил), куда включаются первые три метода;
- ❑ на основе аномалий.

Топология систем обнаружения атак

По топологии размещения в защищаемой сети СОА обычно разделяются на два вида или представляют двуединую сущность из следующих составляющих: СОА на хосте (COAX) и СОА на сети (COAC) (англ. *host-based* и *network-based*). В качестве пролога рассмотрим логические составляющие СОА.

- ❑ *Агент* — логическое устройство, собирающее информацию об активности, которая подлежит анализу на предмет присутствия атаки. Для активных СОА дополнительно — это элемент воздействия на атаку (инициатор разрыва соединения, закрытия порта и пр.).
- ❑ *База данных* — набор информации о принципах и правилах определения атаки и признаках атаки. Возможно также, это сервер анализа информации от агентов и устройство принятия решения по определению атаки и ответной реакции.
- ❑ *Менеджер* — устройство, обеспечивающее слаженную работу распределенных агентов. Функциями могут быть обновление программного обеспечения на агентах, признаков атак в базах данных, предоставление информации от агентов на консоль и т. д.
- ❑ *Консоль* — устройство управления системой в целом. С консоли производится настройка системы, и здесь получаются отчеты по работе системы.

В зависимости от реализации системы различные логические сущности могут быть объединены, например, агент с базой или база с менеджером или менеджер с консолью.

В САОХ агент размещается прямо на защищаемом хосте, его не интересуют другие хосты сети. В данном случае агент занят работой с сетевой и локаль-

ной активностью, специфической именно для хоста, на котором он установлен — сетевые атаки на данную операционную систему, подозрительные попытки работы с консоли хоста (подбор пароля, установка нового программного обеспечения) и т. п.

В СОАС агент будет размещаться в некотором сегменте сети, подлежащем защите, на отдельном сетевом устройстве или на рабочей станции пользователя (в этом случае она должна быть достаточно производительной). Агент перехватывает и анализирует все кадры, поступающие на сетевой интерфейс устройства, на котором установлен агент. Возможен случай, когда устройство агента снабжено двумя сетевыми интерфейсами и является шлюзом, анализирующим сетевую активность в двух сегментах и строго препятствующим распространению атаки из одного сегмента в другой. В случае СОАС агент нацелен на работу с информацией, специфичной для сетевых протоколов.

Если планирование внедрения СОАХ достаточно просто (количество приобретаемых единиц продукта равно количеству хостов, необходимых для защиты), то для сетевого варианта возможно появление ряда сложностей.

В самом начале работы по проекту СОАС для своей организации специалист, знакомясь с материалами того или иного производителя систем, скорее всего увидит достаточное количество красивых иллюстраций (пусть даже технически достаточно подробных), описывающих работу СОАС. Однако необходимо учитывать, что иллюстрации чаще всего отражают устройство типовой или идеальной сети, а реальная сеть организации может представить некоторые проблемы для эффективного внедрения СОАС.

Сильно сегментированная сеть (в сегменте 10—15 рабочих станций или группа серверов) может быть присоединена на отдельный порт коммутатора и потребовать очень большого количества агентов для полного охвата (рис. 23.7). Даже при условии фокусирования направления только на критически важные участки сети прямое решение может представлять из себя вариант, требующий больших финансовых затрат.

Причины этого обычно заключаются в том, что СОАС приобретается уже для "живой", работающей сети, а не сеть проектируется для эффективной работы СОА. Возьмем крайний случай. 20 серверов присоединены к 20 портам 24-портового коммутатора (например, для организации нескольких VLAN). Каким образом обеспечить эффективное внедрение СОА для такого варианта?

Учитывая, что обычно в коммерческих продуктах агент (лицензия на агент) для СОАС стоит дороже, чем для СОАХ, наиболее дорогостоящим решением будет установка отдельного агента СОАС для каждого порта коммутатора (рис. 23.8). Если брать во внимание необходимость сохранения производительности серверов, то необходимо приобретать еще и хосты, на которых будут размещены агенты СОАС.

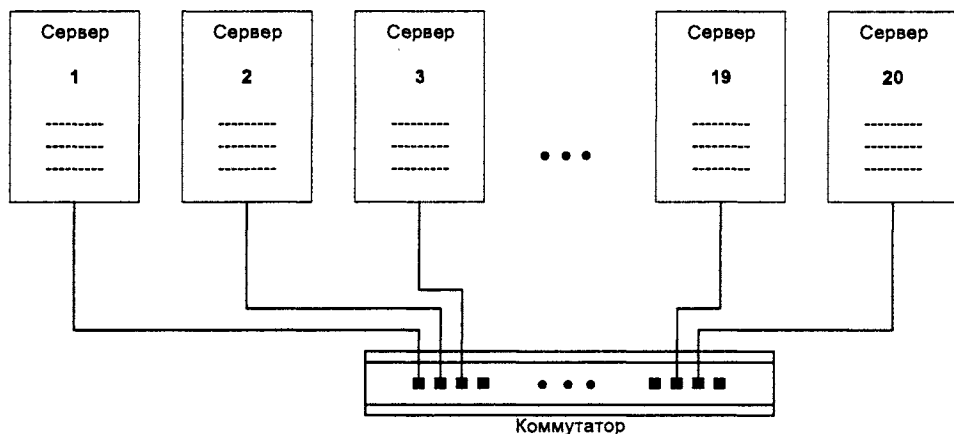


Рис. 23.7. Сегментированный участок сети

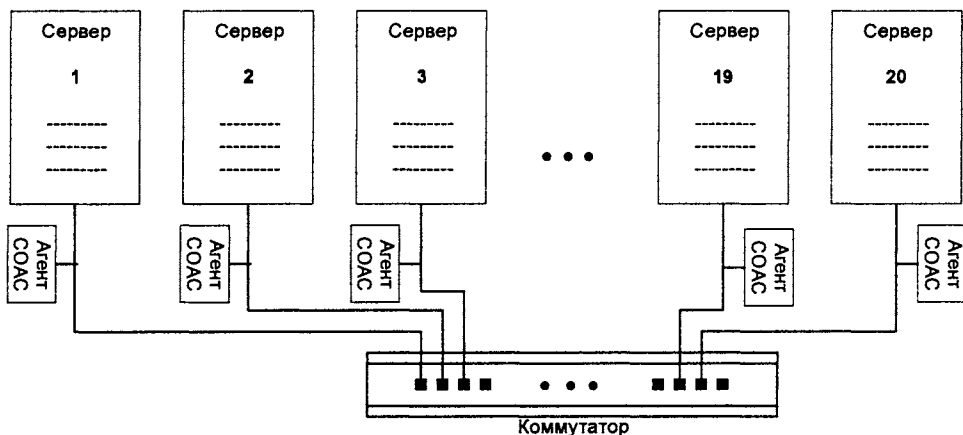


Рис. 23.8. Вариант с несколькими агентами СОАС

Другим вариантом решения будет размещение на серверах 20 агентов СОАХ (рис. 23.9). При этом также следует учитывать некоторое снижение производительности работы серверов.

Наиболее оптимальным будет решение по установке одного агента СОАС таким образом, чтобы он смог производить мониторинг всех портов коммутатора (рис. 23.10). Это возможно:

- ❑ если сам коммутатор предоставляет возможности "зеркалирования" информации со всех портов на один *технология SPAN (Switched Port Analyzer)*;
- ❑ приобретением дополнительного устройства (устройств) — например, устройства копирования трафика (англ. *Test Access Point — TAP*), собирающего информацию с сегмента.

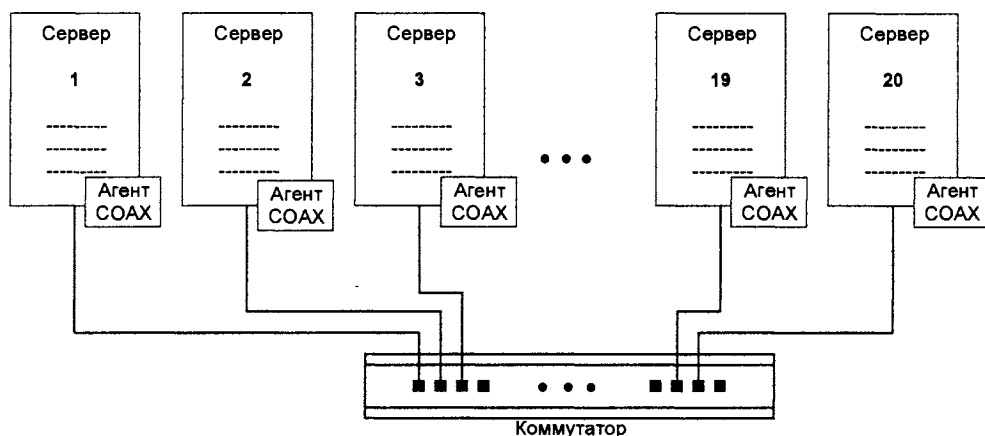


Рис. 23.9. Вариант с несколькими агентами COAX

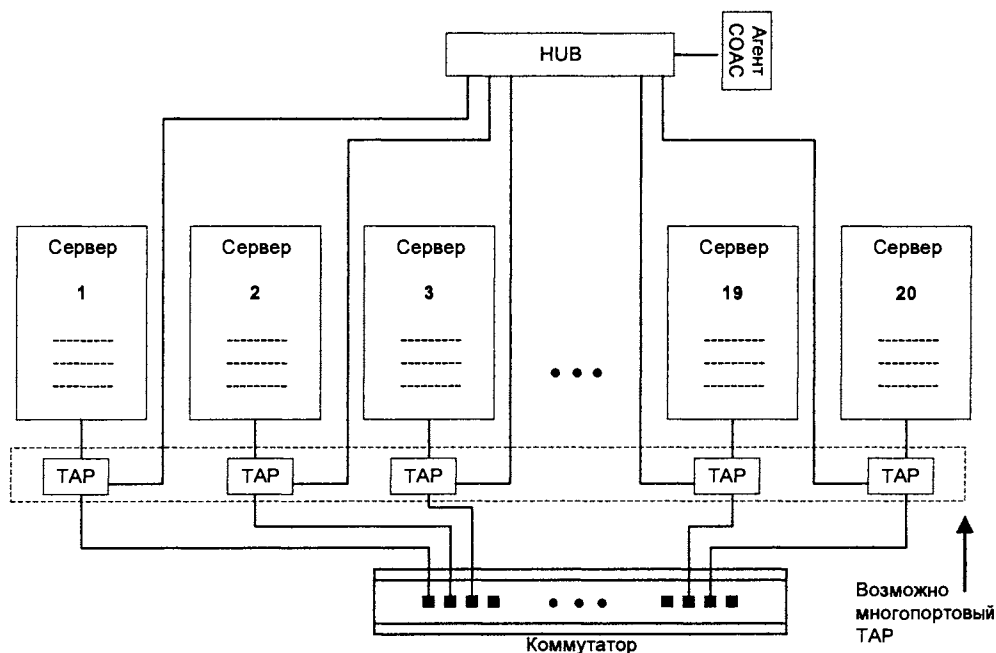


Рис. 23.10. Вариант с одним агентом COAX

Следует подробно ознакомиться с техническими характеристиками приобретаемых устройств, так как могут возникнуть дополнительные сложности, например, такие как:

- ☐ многопортовый TAP не позволяет принимать информацию одновременно со всех портов, а только с одного порта за промежуток времени;

- ❑ несколько ТАР, соединенных в хаб, при интенсивной работе вызывают коллизии, что приводит к потере данных;
- ❑ ТАР может работать только по принципу half-duplex, т. е. прием и передача в сегменте не могут выполняться одновременно.

Все это следует учесть перед тем, как формировать спецификацию для приобретения оборудования, подсчета количества агентов, расчета лицензионной политики по СОА и других значимых действий.

Всех, кто интересуется системами обнаружения атак на еще более детальном уровне, авторы адресуют к фундаментальному труду Алексея Лукацкого "Обнаружение атак" [Лукацкий2], вышедшему в издательстве "БХВ-Петербург".

Автоматизированные средства управления политикой безопасности



Наличие автоматизированных средств управления политикой безопасности (англ. *policy manager, security manager*) в организации подразумевает обычно присутствие следующих факторов:

- ❑ развитой инфраструктуры безопасности, так как данное программное обеспечение должно управлять именно некими механизмами безопасности, а при их отсутствии бесполезно;
- ❑ достаточной интеграции между элементами инфраструктуры безопасности, так как средство автоматизированного управления должно иметь возможность общаться с устройствами безопасности для получения от них информации и направления им управляющих воздействий;
- ❑ квалифицированного персонала для управления безопасностью, так как без детального представления принципов работы механизмов безопасности автоматизированное управление безопасностью бесполезно и рискованно.

Приведем список возможностей идеального средства управления политикой безопасности, хотя не все коммерческие продукты могут обладать этим списком возможностей в полном объеме. Принципиальная характеристика таких продуктов — это выработка единой политики для данного типа устройств, сегмента сети, операционной системы, группы пользователей в разрезе различных областей безопасности, и применение этой политики в корпоративной сети.

- ❑ Разрешение способов аутентификации на данном объекте для данного субъекта. Здесь могут формироваться как указания конкретных аутентификационных протоколов, разрешенных к использованию, и серверов аутентификации, так и определение парольной политики для пользова-

телей — размер и составляющие паролей, срок действия, парольная история и т. п.

- ❑ Разрешение на прохождение данного типа трафика (аналогия с межсетевым экраном) по различным его параметрам — источник, получатель, время, защищенность, содержание.
- ❑ Проверка целостности системных файлов и настроек, особенно настроек безопасности. Проверка на отсутствие уязвимостей в операционной системе, базе данных, приложениях (аналогия со сканером безопасности).
- ❑ Определение способов аудита, детальности и расположения регистрационных журналов.

Кроме перечисленных возможностей, следует учитывать, что консоль подобного программного обеспечения обычно становится единым местом, куда стекаются все отчеты и сообщения о состоянии информационной безопасности предприятия.

Дополнительные автоматизированные средства



Перечислим некоторые средства, которые могут и не являться напрямую средствами обеспечения информационной безопасности, но тем не менее помогают обеспечивать тот или иной ее аспект.

- ❑ Средства управления конфигурацией или инвентаризацией. Обычно такое средство является инструментом службы поддержки работы пользователей и состоит из некоторого агента, установленного на хостах пользователя и центральной базы с консолью. Средство позволяет обнаруживать новые аппаратные устройства, появившиеся на хосте со времени предыдущего анализа, новое установленное программное обеспечение и изменения, внесенные в существующие настройки среды хоста.
- ❑ Средства управления проектами. Помогают эффективно распределять ресурсы при решении конкретного проекта. Поскольку внедрение многих средств информационной безопасности (даже таких, как проработка документации и руководств для пользователей и администраторов) носит характер ИТ-проектов, данное средство может увеличить эффективность работы.
- ❑ Средства оценки рисков. Обычно представляют собой интерактивный вопросник, но (в зависимости от цены) могут иметь и интеллектуальный элемент анализа текущей сетевой инфраструктуры. Данное средство хорошо тем, что при изменении ряда параметров информационного пространства позволяет оперативно произвести новую оценку без необходимости повторных работ по определению влияния изменившихся параметров на остальные объекты.

Инструменты злоумышленников

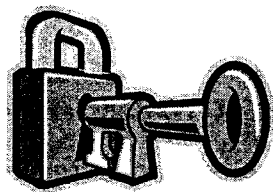


Иногда для эффективного обеспечения информационной безопасности специалистам организации приходится использовать некоторые средства, которые обычно используются злоумышленниками для взлома информационных систем — т. н. *хакерские инструменты*. Можно долго спорить о первоначальных целях, которые ставились разработчиками таких средств, возможно, они были самыми благими. Однако, глядя в глаза реальному миру, можно утверждать, что большая часть случаев использования таких средств все же относится к попыткам несанкционированного использования информационных ресурсов. В отношении этого класса программного обеспечения провести четкую грань между полезным и опасным, поощряемым и запрещенным очень и очень сложно.

Интерес для сотрудников службы безопасности представляют.

- ❑ **Инструмент определения подключенного к телефонному каналу модема (англ. *war-dialer*).** Представляется целесообразным периодически проверять телефонные номера организации на предмет подключенных и/или оставленных активными модемов. Частоту проверок следует увеличивать, если сотрудники организации замечены в ночное время работающими или посещающими Интернет из дома за счет канала организации.
- ❑ **Сканер портов (усиленный вариант — свободно распространяемый сканер уязвимостей).** Скорее всего актуален для использования в организации, где не внедрен автоматизированный сканер уязвимостей. Позволяет выявить незакрытые порты (службы) системы, которые в дальнейшем могут быть использованы для атаки.
- ❑ **Взломщик паролей.** Позволяет периодически с соответствующими полномочиями проверять пароли пользователей на их уязвимость. В случае обнаружения паролей, уязвимых к атаке полным перебором, целесообразно принуждать пользователей к их замене. Ряд средств может быть использован для восстановления, например, электронных документов и таблиц, которые были защищены шифрованием с последующей утерей пароля (ключа шифрования). Следует обратить особое внимание на оформление этой процедуры, с тем чтобы специалист, производящий санкционированный взлом, не получил в свое распоряжение пользовательские пароли и, таким образом, не расширил свои привилегии.
- ❑ **Перехватчик сетевых пакетов (англ. *sniffer*).** Может использоваться службами сетевой поддержки для детального мониторинга сети. В службе безопасности применяется для выявления подозрительной активности, например, работы замаскированной троянской программы на хосте пользователя или большого количества широковебательных запросов, как потенциальную угрозу атаки на отказ в обслуживании.

Глава 24



Выбор конкретных автоматизированных средств безопасности

В *главе 23* уже был затронут ряд вопросов, на которых необходимо сосредоточить внимание при выборе той или иной системы. Однако, когда дело доходит уже до приобретения конкретного продукта в виде запроса на коммерческие предложения производителей, организации тендера или конкурса, то качество полученной информации, на основе которой возможно принимать решение, в значительной степени зависит от того, насколько качественно составлено техническое задание на систему, планируемую к использованию, и насколько специалисты, производящие анализ, компетентны в данной сфере. В этой главе будут представлены ссылки и выдержки из стандартов и документов некоторых известных экспертов в области анализа и тестирования систем безопасности, доступных в настоящее время в Интернете, а также приведены технические разделы из реальных технических заданий на приобретение средств безопасности.

Следует учесть, что из рассмотрения упущены многие нетехнические аспекты, такие как цена, условия поставки, стоимость гарантийной поддержки, обучения специалистов предприятия и т. д., которые тем не менее могут сыграть существенную роль в принятии решения.

Нормативные акты и аналитические материалы



Межсетевые экраны. Показатели защищенности от несанкционированного доступа к информации

Полное название основного документа, на который мы будем ссылаться: Средства вычислительной техники. Межсетевые экраны. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации: Руководящий документ / Государственная техническая комиссия при Президенте Российской Федерации [ГТК2].

Данный источник, также как и связанные с ним другие руководящие документы, как уже упоминалось, доступен для подробного изучения в Интернете. Кроме того, с кратким содержанием этого и других российских стандартов можно ознакомиться в *главе 29*.

Термины и определения

В документе [ГТК1], вышедшем несколько ранее указанного [ГТК2], предлагается классификация автоматизированных систем (АС) применительно к обработке конфиденциальной информации. Вводятся три группы АС:

1. Многопользовательские АС, обрабатывающие информацию различных уровней конфиденциальности.
2. Многопользовательские АС, в которых все пользователи имеют равный доступ ко всей обрабатываемой информации, расположенной на носителях разного уровня конфиденциальности.
3. Однопользовательские АС, в которых пользователь имеет полный доступ ко всей обрабатываемой информации, расположенной на носителях разного уровня конфиденциальности.

В первой группе выделяют 5 классов защищенности АС: 1А, 1Б, 1В, 1Г, 1Д, во второй и третьей группе — по 2 класса защищенности: 2А, 2Б и 3А, 3Б соответственно. Класс А соответствует максимальной, класс Д — минимальной защищенности АС.

Под сетями ЭВМ, распределенными автоматизированными системами, в документе [ГТК2] понимаются соединенные каналами связи системы обработки данных, ориентированные на конкретного пользователя.

Межсетевой экран (МСЭ) — это локальное (однокомпонентное) или функционально-распределенное программное (программно-аппаратное) средство (комплекс), реализующее контроль за информацией, поступающей в АС и/или выходящей из АС. МСЭ обеспечивает защиту АС посредством фильтрации информации, т. е. ее анализа по совокупности критериев и принятия решения о ее распространении в (из) АС на основе заданных правил, проводя таким образом разграничение доступа субъектов из одной АС к объектам другой АС. Каждое правило запрещает или разрешает передачу информации определенного вида между субъектами и объектами. Как следствие, субъекты из одной АС получают доступ только к разрешенным информационным объектам из другой АС. Интерпретация набора правил выполняется последовательностью фильтров, которые разрешают или запрещают передачу данных (пакетов) на следующий фильтр или уровень протокола.

Правила фильтрации — перечень условий, по которым с использованием заданных критериев фильтрации осуществляется разрешение или запрещение дальнейшей передачи пакетов (данных) и перечень действий, произво-

димых МЭ по регистрации и/или осуществлению дополнительных защитных функций.

Межсетевой экран может строиться с помощью экранирующих агентов, которые обеспечивают установление соединения между субъектом и объектом, а затем пересылают информацию, осуществляя контроль и/или регистрацию. Использование экранирующих агентов позволяет предоставить дополнительную защитную функцию — сокрытие от субъекта истинного объекта. В то же время субъекту кажется, что он непосредственно взаимодействует с объектом. Обычно экран не является симметричным, для него определены понятия "внутри" и "снаружи". При этом задача экранирования формулируется как защита внутренней области от неконтролируемой и потенциально враждебной внешней.

Сетевые адреса — адресные данные, идентифицирующие субъекты и объекты и используемые протоколом сетевого уровня модели международной организации по стандартизации взаимодействия открытых систем (OSI). Сетевой протокол выполняет управление коммуникационными ресурсами, маршрутизацию пакетов, их компоновку для передачи в сети. В этих протоколах решается возможность доступа к подсети, определяется маршрут передачи и осуществляется трансляция сообщения. Управление доступом на сетевом уровне позволяет отклонять нежелательные вызовы и дает возможность различным подсетям управлять использованием ресурсов сетевого уровня. Поэтому в данных протоколах возможно выполнение требований по защите в части проверки подлинности сетевых ресурсов, источника и приемника данных, принимаемых сообщений, проведения контроля доступа к ресурсам сети.

Экранирование — функция МСЭ, позволяющая поддерживать безопасность объектов внутренней области, игнорируя несанкционированные запросы из внешней области. В результате экранирования уменьшается уязвимость внутренних объектов, поскольку первоначально сторонний нарушитель должен преодолеть МСЭ, где защитные механизмы сконфигурированы особенно тщательно и жестко. Кроме того, экранирующая система в отличие от универсальной может и должна быть устроена более простым и, следовательно, более безопасным образом, на ней обязаны присутствовать только те компоненты, которые необходимы для выполнения функций экранирования. Экранирование дает также возможность контролировать информационные потоки, направленные во внешнюю область, что способствует поддержанию во внутренней области режима конфиденциальности. Помимо функций разграничения доступа, экраны осуществляют регистрацию информационных обменов.

Пять классов защищенности МСЭ

Самый низкий класс защищенности — пятый, применяемый для безопасного взаимодействия АС класса 1Д с внешней средой, четвертый — для 1Г, третий — 1В, второй — 1Б, самый высокий — первый, применяемый для безопасного взаимодействия АС класса 1А с внешней средой.

Для АС класса 3Б, 2Б должны применяться МСЭ не ниже 5 класса.

Для АС класса 3А, 2А в зависимости от важности обрабатываемой информации должны применяться МСЭ следующих классов:

- ☐ при обработке информации с грифом "секретно" — не ниже 3 класса;
- ☐ при обработке информации с грифом "совершенно секретно" — не ниже 2 класса;
- ☐ при обработке информации с грифом "особой важности" — не ниже 1 класса.

Показатели защищенности сведены в табл. 24.1.

Обозначения:

" — " — нет требований к данному классу;

" + " — новые или дополнительные требования;

" = " — требования совпадают с требованиями к МСЭ предыдущего класса.

Таблица 24.1. Показатели защищенности для классов

Показатели защищенности	Классы защищенности				
	5	4	3	2	1
Управление доступом (фильтрация данных и трансляция адресов)	+	+	+	+	=
Идентификация и аутентификация	-	-	+	=	+
Регистрация	-	+	+	+	=
Администрирование: идентификация и аутентификация	+	=	+	+	+
Администрирование: регистрация	+	+	+	=	=
Администрирование: простота использования	-	-	+	=	+
Целостность	+	=	+	+	+
Восстановление	+	=	=	+	=
Тестирование	+	+	+	+	+
Руководство администратора защиты	+	=	=	=	=
Тестовая документация	+	+	+	+	+
Конструкторская (проектная) документация	+	=	+	=	+

Руководства для приобретающих межсетевой экран и приобретающих систему обнаружения атак

Данные материалы разработаны лабораторией ICSA — исследовательским подразделением компании TruSecure. Наименования в оригинале "Firewall Buyers Guide" и "IDS Buyers Guide" соответственно. Документы интересны в полном объеме, так как позволяют получить представление о системах безопасности и используемых технологиях, но в плане практических рекомендаций наибольший интерес представляют следующие разделы:

Для межсетевых экранов — раздел *"Firewall Buyer's Assessment Form"* (Форма оценки покупателя межсетевого экрана). Предполагается, что предприятие, заполнив данную форму и выслав ее производителю, позволит последнему сформировать наиболее качественное предложение для покупателя. Заполнение данной формы, впрочем, и без отправки ее кому-либо, позволит организации лучше понять, какое решение ей необходимо.

1. Контактная информация — адрес и ответственные лица.
2. Бизнес-среда работы:
 - количество и расположение отдельных учреждений (зданий) предприятия;
 - указание подразделений и информации ограниченного характера и информации, для которой важна доступность данных для взаимодействия подразделений, их размещение;
 - указание внешних партнеров, с которыми необходимо организовать взаимодействие;
 - описание сервисов, открытых публично;
 - требования к организации удаленного доступа во внутреннее информационное пространство организации;
 - сервисы электронных служб, использующих публичные каналы связи (например, электронная коммерция).
3. Планируемые изменения в бизнес-среде по перечисленным параметрам.
4. Информационная среда:
 - количество пользовательских рабочих станций с указанием аппаратного обеспечения, системного и прикладного программного обеспечения;
 - структура сети с указанием топологии, среды передачи данных, используемых устройств и протоколов;
 - структура удаленного доступа с указанием используемых устройств, а также методов аутентификации;

- количество серверов с указанием аппаратного обеспечения, системного и прикладного программного обеспечения;
 - существующая система поддержки информационных систем со стороны поставщиков с их указанием и границами сферы деятельности;
 - антивирусные системы и другие системы контроля программного обеспечения;
 - технологии управления сетью и информационными системами;
 - аутентификационные технологии — список и описание.
5. Планируемые изменения в информационной среде по перечисленным параметрам.
6. Связь с Интернетом:
- тип интернет-соединения;
 - существующие межсетевые экраны (если есть);
 - средства связи с внешней средой, используемые внутренними системами;
 - внутренние системы и сервисы, доступные извне;
 - серверы электронной коммерции и других транзакционных систем;
 - указание на наличие утвержденной политики безопасности доступа и использования Интернета.
7. Планируемые мероприятия (для которых приобретается межсетевой экран):
- изменения в способах доступа к Интернету и в политике безопасности предприятия;
 - появление новых протоколов, которые необходимо поддерживать отдельно для внутренних пользователей, пользователей с удаленным доступом или специальных пользователей, доступных публично.
8. Требуемая функциональность межсетевого экрана:
- по контролю доступа;
 - выдаваемым сообщениям;
 - аутентификации;
 - управлению конфигурацией;
 - контролю содержимого проходящего трафика;
 - регистрационным журналам;
 - распознаванию атак;
 - сетевым опциям (количество интерфейсов, способ доступа);
 - удаленному администрированию;

- системным требованиям (под ключ, интеграция с другими продуктами и т. д.);
- поддержке виртуальных частных сетей (VPN).

9. Прочие условия:

- предполагаемая стоимость межсетевого экрана (сколько организация может потратить);
- предполагаемая дата начала работы продукта;
- требования к наличию у продукта сертификатов;
- требования к предполагаемому администратору продукта и к службе поддержки;
- специальные условия контракта с поставщиком (если есть);
- другие замечания, которые не включены в данную форму.

Для системы обнаружения атак наиболее интересен раздел "*Guidelines for Selecting Products*" ("Руководства по выбору продукта"). Документ акцентирует внимание на процессе принятия решения и проектирования использования СОА. В разделе представлена текстовая информация, а также алгоритмически структурированные схемы, разделяющие процесс на три части — решение руководства по необходимости внедрения продукта, техническое решение по сетевой среде, анализ критических Веб-серверов.

Практика технологий систем обнаружения атак

Данный материал (название в оригинале — "State of the Practice of Intrusion Detection Technologies") создан Институтом разработки программного обеспечения — Software Engineering Institute. Документ интересен для подробного знакомства с системами обнаружения атак, а для определения критериев выбора системы важен раздел "Appendix F: Candidate IDS Selection Criteria" (Критерии выбора кандидатов СОА).

Возможности обнаружения

Время, которое пройдет между тем, как произойдет событие атаки и его автоматическим распознаванием.

Хранящаяся информация об атаках — полнота и расширяемость базы знаний об атаках в продукте, включая общую работу, которая потребуется для формулирования нового подхода к обнаружению и установку его на работающий продукт.

Профилирование активности пользователей и систем — способность разрабатывать настраиваемые профили пользователей и систем как основу для распознавания аномального поведения.

Соотнесение информации от разных источников — способность СОА устанавливать связь между трафиком для мониторинга или сообщениями от различных источников и в различное время.

Операционные возможности

Тестирование и верификация. Включает требования к представлению документов о проведении тестирования продукта.

Системная безопасность — степень, с которой СОА защищает свои ресурсы от атак на безопасность.

Поддерживаемые сетевые среды — отражает типы сетевых носителей (интерфейсов, протоколов, скоростей), на которых система может работать с требуемой пропускной способностью, не теряя при этом сетевые пакеты.

Пользовательский интерфейс — тип интерфейса, который СОА предоставляет администратору для конфигурирования и мониторинга.

Системы обнаружения атак:

Групповой тест

Документ (название в оригинале — Intrusion Detection Systems, Group test) разработан компанией NSS Group и представляет собой более чем 200-страничное описание квалифицированного экспертного тестирования различных производителей СОА по наименованиям продуктов с указанием преимуществ и недостатков, выявленных при тестировании продуктов. В текущем издании были протестированы следующие продукты:

- ☐ Cisco Secure IDS V2.5;
- ☐ CA Entrust Intrusion Detection V1.4.5;
- ☐ Cybersafe Centrax 2.4;
- ☐ Enterasys Dragon Sensor 4.2;
- ☐ Enterscept 2.01;
- ☐ Securenet Pro 4.0, Intrusion Inc.;
- ☐ Real Secure 5.0, ISS;
- ☐ LanGuard S.E.L.M.;
- ☐ BlackIce Sentry 2.1, Network Ice;
- ☐ NFR NID-200 V1.1;
- ☐ NSecure NPatrol IDS V1.3.2;
- ☐ Snort 1.8.1;

- ☐ Intruder Alert 3.5, Symantec;
- ☐ Net Prowler 3.51, Symantec;
- ☐ TripWire for Servers V2.4.2;

Таким образом, с одной стороны, можно получить представление о лидерах рынка COA, а с другой стороны — о способах и методах тестирования таких систем.

Примеры технических заданий

Выдержка из технического задания на межсетевой экран

Представлена часть технического задания, на основе которого был проведен конкурс по приобретению межсетевого экрана.

Требование к функциональности программного продукта

Программное обеспечение межсетевого экрана должно представлять собой продукт известного производителя и марки, зарекомендовавшей себя на рынке информационной безопасности, имеющей положительные отзывы независимых экспертов и сертификационных компаний. Продукт должен обладать следующими функциональными возможностями:

- ☐ анализировать входящий и/или исходящий информационный поток от сетевого до прикладного уровня модели OSI включительно;
- ☐ иметь встроенный редактируемый и пополняемый набор правил для авторизации, фильтрации/разграничения доступа из внешней/во внешнюю сети/сеть во внутреннюю сеть/из внешней сети;
- ☐ иметь возможность автоматической проверки создаваемых правил, в том числе на целостность, непротиворечивость и соответствие общей политике;
- ☐ возможность включения в правила контроль загружаемых пользователями файлов (запрет файлов с расширениями: exe, com, zip и т. п.), контроль java апплетов и скриптов, контроль спецсимволов в URL;
- ☐ возможность создания черных списков сетевых адресов (запрет на доступ пользователей к конкретным URL);
- ☐ возможность управления доступом пользователей (время дня, день недели, парольная политика и т. п.);
- ☐ возможность начать работу сразу после установки с минимальными требованиями к администрированию, иметь средства автоматического восстановления работоспособности после сбоев;

- ❑ иметь возможность управлять двумя и более межсетевыми экранами с одного рабочего места во внутренней сети при едином графическом пользовательском интерфейсе;
- ❑ иметь возможность создания демилитаризованной зоны (DMZ) в выделенной подсети для обслуживания Веб-сайта, FTP-сервера и т. п.;
- ❑ иметь возможность интеграции с антивирусным пакетом для межсетевого экрана (желательно) или быть совместимым с ним (обязательно);
- ❑ поддерживать основные сетевые протоколы безопасности, включая, но не ограничиваясь HTTPS, SSL, RADIUS, TACACS;
- ❑ иметь возможность создания виртуальной частной сети (VPN) с аналогичным межсетевым экраном или с удаленным пользователем на основе стандартных протоколов безопасности (IPSec);
- ❑ контролировать целостность и защиту операционной системы, на которую установлен продукт;
- ❑ обеспечивать формирование регистрационных журналов в дружелюбном, интуитивно понятном виде и с возможностью фиксации всех значимых событий.

Требование к аппаратному и системному программному обеспечению

Аппаратное обеспечение межсетевого экрана должно представлять собой продукт известного производителя и марки, зарекомендовавший себя на рынке информационных технологий.

Аппаратное обеспечение должно:

- ❑ быть снабжено необходимой периферией для обеспечения консольной работы (видеомонитор с экраном не менее 17", клавиатура, манипулятор мышь) и возможности консольной установки (CD-ROM, дисковод для гибких дисков 3,5");
- ❑ соответствовать требованиям к поддержке операционной системы межсетевого экрана на уровне не ниже рекомендуемого производителем операционной системы;
- ❑ обеспечивать возможность работы без снижения производительности для:
 - 100 внутренних пользователей сервиса Интернета одновременно;
 - 25 внешних пользователей внутреннего Веб-сайта одновременно;
- ❑ иметь минимум три сетевых интерфейса Ethernet (UTP);
- ❑ иметь возможность работы через Dial-Up интерфейс;

- ❑ обеспечивать уровень отказоустойчивости не ниже RAID 1 (дуплексирование);
- ❑ обеспечивать сохранение регистрационных журналов межсетевого экрана при максимальной активности и детальности — не менее месяца.

Операционная система должна быть представлена с необходимым количеством лицензий.

Операционная система должна быть сертифицирована до уровня не ниже C2 NCSC-TG-005 "Красная книга" или аналогичного уровня для страны производителя и/или иметь возможность уплотнения ядра программным обеспечением межсетевого экрана (замена IP-стека, отключение сервисов, не требующихся в работе межсетевого экрана).

Выдержка из технического задания на систему обнаружения атак

Приводится сетевая часть технического задания, на основе которого был проведен конкурс по приобретению системы обнаружения атак.

Общие требования

Акцент в СОА должен быть сделан на обнаружение сетевых атак. Если система подразумевает функциональное разделение этих типов по агентам на сетевые и системные, то необходимо предложить N сетевых агентов.

СОА должна функционировать на основе сигнатур атак.

Система должна быть поставлена с соответствующим аппаратным обеспечением и функционировать на операционной системе Windows NT/2000.

Участники конкурса должны представить предложения с соответствующим техническим описанием и документацией, раскрывающей специфику выполнения перечисленных далее требований.

Требования к логической части СОА

Определяющие требования

1. Каждый агент обнаружения атак (далее агент) должен быть легко масштабируемым, т. е. с увеличением числа рабочих станций и узлов сегмента производительность должна зависеть только от параметров аппаратного обеспечения агента. Производительность самого агента не должна зависеть от архитектуры, программного обеспечения, организации базы данных агента или управляющей консоли.
2. Все агенты должны иметь возможность управляться с одной консоли (при наличии у предприятия нескольких консолей — с любой из них).

Каждый агент должен иметь возможность удаленного конфигурирования, управления, обновления. При этом в случае отсутствия связи с управляющей консолью агент должен иметь возможность самостоятельной работы в рамках имеющихся у него правил и накопления информации для отправления ее консоли после восстановления связи.

3. База сигнатур атак должна покрывать все основные имеющиеся на данный момент сетевые атаки, а также обладать возможностью определения самостоятельных сигнатур по стандартным используемым протоколам, например, определения в качестве атаки пакетов сетевой компьютерной игры. Система должна анализировать все основные имеющиеся на данный момент сервисы и протоколы, включая, но не ограничиваясь следующим списком (IP, ICMP, TCP, UDP, HTTP, FTP, TELNET, SMTP, SNMP, IRC, POP3, rshell, rlogin).
4. СОА должна предусматривать механизм настройки правил прекращения сетевой атаки в случае ее обнаружения и оперативного информирования об атаке уполномоченного администратора.
5. СОА должна быть сертифицирована известным уполномоченным органом или быть участником гласного независимого экспертного исследования 2000—2001 года (в этом случае необходимо представление дополнительных материалов или предоставления ссылок для их получения).
6. Трафик обмена данными между агентом и консолью должен быть защищен от прослушивания или модификации.
7. Система должна иметь возможность работы в скрытом режиме, т. е. работа сетевых агентов системы не должна обнаруживаться стандартными сетевыми утилитами.
8. Система не должна показывать существенного замедления функционирования в условиях возрастания интенсивности трафика сети до предельных значений.

Предпочтительные требования

1. Обновление баз данных сигнатур атак должно производиться без останова или перезагрузки агента или консоли.
2. Доступ к консоли управления СОА должен иметь дополнительные ограничения доступа, кроме средств, предлагаемых операционной системой.
3. Система должна иметь возможность интеграции с межсетевым экраном, в смысле возможности посылки управляющих сигналов на блокирование портов, изменения правил и т. п.
4. Система должна предоставлять простой и интуитивно понятный интерфейс управления, а также простую и интуитивно понятную систему предоставления отчетов.

Дополнительные пожелания

1. Система должна предусматривать дополнительные возможности извещения администратора, кроме сообщения на управляющую консоль, такие как сообщения на пейджер, SNMP агенту, посылки сообщения по электронной почте.
2. Должно быть предусмотрено обновление базы сигнатур атак на группе сетевых сканеров одновременно, обновление сигнатур атак автоматизировано, через Интернет.

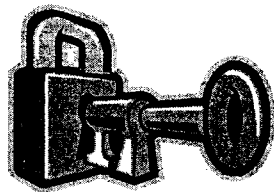
Требования к аппаратному обеспечению

Аппаратное обеспечение должно обеспечивать необходимую производительность системы.

При этом поставка аппаратуры пользовательского ввода/вывода (монитор, клавиатура, мышь) должна быть только для консоли управления. Агенты могут быть поставлены только для удаленного управления, но при этом допускать возможность подключения аппаратуры пользовательского ввода/вывода заказчика.

Аппаратное обеспечение агента должно гарантировать его полную функциональность, согласно перечисленным требованиям. Например, если для предотвращения сетевой атаки агент должен быть шлюзом для данного сегмента, необходима поставка агента с двумя сетевыми интерфейсами.

Глава 25



Разработка приложений

Разработка программного обеспечения (ПО) с точки зрения информационной безопасности охватывает обычно три следующих аспекта:

1. Проектирование механизмов, предназначенных для поддержания модели безопасности, заложенной в системе.
2. Контроль жизненного цикла программного обеспечения как информационной системы от внесения несанкционированных изменений или наличия недокументированных возможностей.
3. Защита самого программного обеспечения как интеллектуальной собственности или объекта распространения авторского права.

Защита интеллектуальной собственности программного обеспечения

Начнем с третьего аспекта. Хотя этот вопрос не всегда является объектом внимания службы информационной безопасности, но тем не менее конкретная организация может иметь другое мнение и ввести данный аспект в сферу компетенции службы. Защита программного обеспечения как интеллектуальной собственности складывается из формирования внутри программного обеспечения механизмов защиты от пиратского копирования и распространения и от обеспечения соответствия данного объекта интеллектуальной собственности правовому пространству.

Работы по совершенствованию программных механизмов защиты от пиратского копирования и распространения ведутся постоянно, и в настоящее время авторам не известно ни одного надежного на 100% способа такой защиты. Противостояние разработчиков и сообщества *взломищиков программ* (англ. *code-crackers*) носит характер "спора" снаряда и брони, а также допускает возможность применения афоризма: что один человек сделал, другой завсегда разобрать сможет. Огромное количество взломанного программного обеспечения (англ. *warez*) на соответствующих сайтах Интернета, в том числе продуктов известных компаний, а также компаний, ориентированных на информационную безопасность, тому подтверждение.

Косвенным подтверждением тому, что надежных методов такой защиты в настоящее время нет (и по-видимому, в ближайшем будущем не появится), является ориентация ряда крупных компаний на законодательные меры и нормативные акты более, чем на программные механизмы защиты.

Если организация также предполагает использовать правовые методы защиты программного обеспечения, то следует учесть два важных вопроса.

Во-первых, законодательство страны должно определять программное обеспечение как объект применения законов об интеллектуальной собственности. Если таковые законы отсутствуют, необходимо проанализировать существующее законодательство на предмет возможности применения альтернативных определений типа "нематериальный актив" и т. д.

Во-вторых, необходимо обеспечить ситуацию, когда существующие законы могут быть применены, т. е. фактически заключение лицензионного (или иного) соглашения между производителем/продавцом программного обеспечения и пользователем/покупателем программного обеспечения. При этом следует учесть, что стандартные формулировки типа "нажимая клавишу ВВОД, вы тем самым подтверждаете согласие..." или "вскрытие данной упаковки равносильно признанию лицензионного соглашения...", не являются абсолютными аргументами в суде, возможно, потребуются более серьезные доказательства.

Проектирование механизмов безопасности

Механизмы информационной безопасности в программном обеспечении

Проектирование механизмов безопасности требует от разработчика не только квалифицированного программирования, но и способности к постановке задачи именно с точки зрения безопасности. Даже использование встроенных производителями в операционную систему или в средство разработки возможностей расширения функциональности безопасности (API — Application Programming Interface, интерфейс прикладного программирования) и его использование в программном обеспечении не означает сохранение целостной, непротиворечивой и достаточно неуязвимой модели безопасности.

Поэтому в группе (проекте) разработки серьезного программного обеспечения, на наш взгляд, должен присутствовать специалист по безопасности, способный как абстрактно, так и достаточно конкретно описать модели:

- ☐ идентификации субъектов;
- ☐ аутентификации и авторизации;
- ☐ контроля доступа к объектам;

- ☐ обеспечения конфиденциальности и целостности информации;
- ☐ апеллируемости действий;
- ☐ а также всех тех аспектов безопасности, о которых идет речь на протяжении всей книги.

Если при этом специалист способен составлять и читать объектно-ориентированные модели и разбираться в написанном коде, это будет значительным плюсом в усилении безопасности создаваемого программного обеспечения.

Различные механизмы безопасности, встраиваемые в программы, в значительной степени ориентированы на использование криптографических средств (им посвящена *часть V данной книги*), поэтому знание основ криптографии необходимо при создании модели безопасности. Однако, на наш взгляд, наиболее специфическим, ориентированным именно на бизнес-цели организации и задачи конкретного программного обеспечения, будет такой элемент информационной безопасности, как *контроль доступа* (англ. *access control*). Действительно, возможно абстрактно говорить о том, что практически в любой задаче потребуются сохранение паролей в зашифрованном виде, ЭЦП, журнал аудита и т. п. Но определить, к каким объектам или их частям, в каком порядке и с какими правами должен иметь доступ субъект, практически невозможно без знания задач, поставленных перед программой.

Абстрактные модели доступа

В связи с этим опишем ряд классических абстрактных моделей доступа, используемых в информационной безопасности. Знакомство с этими моделями поможет понять принципы развития теорий информационной безопасности в целом и разграничения доступа в частности.

Решетчатая модель

Каждому объекту и субъекту присваивается определенный класс доступа. После этого субъект может получать доступ только к объекту равного или более низкого класса. Поскольку обычно количество объектов с увеличением класса доступа уменьшается (в том смысле, что обычно объектов с широким доступом больше, чем объектов с ограниченным доступом), то для субъекта картина доступа выглядит в треугольном или пирамидальном виде. Поскольку такая классическая модель (*Lattice model*) может не всегда удовлетворять требованиям доступа для конкретной организации, требующей более тонкой настройки, иногда можно встретить расширение данной модели за счет добавления т. н. "сфер интересов", в которые объединяются объекты, а сами сферы интересов привязываются к субъектам. В этом случае субъект не сможет получить доступ к объекту с более низким классом, если он не находится в сфере интересов субъекта.

Модель Белл-ЛаПадула

Модель Белл-ЛаПадула (Bell-LaPadula, 1976) предназначена предотвращать несанкционированные, в том числе неумышленные, попытки раскрытия информации с ограниченным доступом. В самом деле, если субъект имеет доступ к объекту равного класса, он может получить данные этого объекта, а поскольку субъект имеет доступ к объектам более низкого класса, он может переместить полученные данные в более низкий класс. Данная модель, как и предыдущая, запрещает читать объект с уровня выше, чем класс субъекта, но при этом, она еще и запрещает записывать данные в объект уровня ниже, чем класс субъекта. При рассмотрении графических интерпретаций модели можно столкнуться со следующими обозначениями:

- " — " правило "no read up" — запрет чтения вышестоящего уровня;
- " — * " правило "no write down" — запрет записи на нижестоящий уровень.

Модель Биба

Модель Биба (Biba, 1977) считается расширением модели Белл-ЛаПадула с понятия *конфиденциальность* на понятие *целостность*. В данном случае субъект не может производить модификацию объектов более высокого класса и не может вызывать на исполнение модули из объектов более низкого класса, а также не может запрашивать сервис от субъекта более высокого класса.

Модель Гогена-Мезигера

Модель Гогена-Мезигера (Goguen-Meseguer, 1982) основана на теории автоматов (виртуальных механизмов, следующих predetermined инструкциям). Согласно этой теории система может при каждом действии переходить из одного разрешенного состояния только в ограниченное количество других разрешенных состояний. Объекты в данной модели защиты разбиваются на домены (наборов объектов, к которым пользователь может иметь доступ), а субъекты — на группы, в соответствии с определенными для них доменами. Переход системы из одного состояния в другое выполняется только в соответствии с т. н. таблицей разрешений, в которой указано, какие операции может выполнять субъект. В данной модели при переходе системы из одного разрешенного состояния в другое используются транзакции, что обеспечивает общую целостность системы, а действия одного субъекта не влияют на действия другого.

Модель Сазерленда

Модель Сазерленда (Sutherland, 1986) уделяет отдельное внимание возможностям использования *скрытых каналов* (англ. *covert channels*) для воздейст-

вия на процесс. Так же как и в модели Гогена-Мезигера, здесь используется машина состояний с некоторым набором начальных состояний и множеством разрешенных функций перемещения, которые переводят устройство от начального состояния в последующие. Механизм безопасности обеспечивается наложением ограничений на объекты и потоки информации между ними.

Модель Кларка-Вильсона

В модели Кларка-Вильсона (Clark-Wilson, 1987) вводится понятие *тройки* (англ. *triple*), которая состоит из субъекта/программы/объекта, т. е. субъекта и объекта, как сущности, находящиеся во взаимоотношениях (модификации, добавления и т. п.), и программы — как посредника, осуществляющего эти взаимоотношения. Программа должна производить работу в соответствии с predetermined правилами, такая процедура иногда называется *корректное действие* (англ. *well-formed transaction*), она обеспечивает безопасность, преимущественно целостность данных объекта.

Поскольку эта модель получила широкое распространение, в частности, Microsoft Windows NT имеет сходную модель безопасности [Blake], рассмотрим некоторые понятия модели подробнее.

Данные в ней разделены на две части — к одним данным (англ. *constrained data items* — *CDI*) применяются правила контроля целостности, к другим (англ. *unconstrained data items* — *UDI*) правила не применяются. К данным применяются два вида процедур — процедура подтверждения целостности (англ. *integrity verification procedure* — *IVP*), используемая для того, чтобы убедиться, что данные находятся в корректном состоянии, и процедура перехода (пресловутая *well-formed transaction*, или *transformation procedure* — *TP*), которая переводит данные из одного корректного состояния в другое. В модели вводятся наборы правил для мониторинга целостности (сертификационные правила — *certification rules*), применяемые администратором, и для сохранения целостности (принудительные правила — *enforcement rules*), исполняемые системой.

Сертификационные правила.

- ❑ C1 — Система должна иметь процедуру подтверждения целостности (IVP) для каждого набора данных с контролем целостности (CDI).
- ❑ C2 — Применение процедуры перехода (TP) к любому набору данных с контролем целостности (CDI) должно поддерживать целостность данного CDI. Все CDI должны быть проверены на то, что производные от них — это корректные CDI.
- ❑ C3 — Любой набор данных с контролем целостности (CDI) может быть изменен только процедурой перехода (TP). Все TP должны быть проверены на то, что они применяют принципы разделения обязанностей (separation of duties) и минимизации привилегий (least privilege).

- ❑ С4 — Все процедуры перехода (TP) должны быть проверены на то, что их действия регистрируются.
- ❑ С5 — Процедуры перехода (TP), работающие с набором данных без контроля целостности (UDI), должны быть проверены на то, что их результатом будут корректные наборы данных с контролем целостности (CDI).

Принудительные правила.

- ❑ E1 — Только проверенные процедуры перехода (TP) могут работать с наборами данных с контролем целостности (CDI).
- ❑ E2 — Пользователи могут получать доступ к наборам данных с контролем целостности (CDI) только через процедуры перехода (TP), для которых они авторизованы.
- ❑ E3 — Система должна аутентифицировать каждого пользователя, пытающегося выполнить процедуру перехода (TP).
- ❑ E4 — Только администратор может определять авторизацию процедуры перехода (TP).

Дополнительная информация

При ознакомлении с моделями безопасности можно встретить термин Основная теорема безопасности (Basic Security Theorem), которая гласит:

"Если система начала функционирование из безопасного состояния и все переходы между состояниями безопасны, то любое промежуточное состояние системы будет безопасным, вне зависимости от информационных потоков, поступающих в систему".

Кроме перечисленных моделей, при изучении документации по схемам контроля доступа скорее всего придется столкнуться еще с рядом специальных терминов.

Mandatory access control — MAC (обязательное управление доступом) и *discretionary access control* — DAC (переназначаемое управление доступом). Различие их в том, что первая модель является строго закрепленной ("защитой" в систему) и никто, даже администратор системы с полным доступом, не может ее изменить (скажем, невозможно убрать нажатие клавиш <Ctrl>+<Alt>+<Del> при загрузке Windows NT); вторая модель позволяет уполномоченному субъекту переназначать правила доступа к объектам для себя и других субъектов, в том числе увеличивать привилегии, даже если по умолчанию они были минимальными (скажем, разрешить обычному пользователю изменение системного времени для определенных целей).

Доступ по правилам (англ. *rule-based access*) и *доступ по ролям* (англ. *role-based access*). В доступе по правилам, способы распределения доступа применяются для всех субъектов (как, скажем, в моделях Biba или MAC).

В доступе по ролям способы доступа различаются в зависимости от того, к какой группе принадлежит субъект (вариант DAC) — эту модель удобно применять, когда в системе присутствует множество субъектов, которых можно разбить по группам, объединяющим, например, единые функциональные обязанности в системе.

Типовое техническое задание по безопасности программного обеспечения

В качестве примера хотелось бы привести общую постановку задачи (конкурсное задание) по безопасности для приобретаемого коммерческого программного обеспечения — Автоматизированной банковской системы (АС). Здесь указаны общие широкие принципы безопасности. Если же данный документ адаптировать для постановки задачи по безопасности для программного обеспечения собственной разработки, то объем текста может вырасти до размеров брошюры или даже книги, похожей на классическое ТЗ (техническое задание) НИИ советского периода. В этом случае требования обязательно должны пополниться заданиями для программистов по запрету недокументированных возможностей, очистке локальных переменных, защите переполнения буфера и многого другого.

Идентификация и аутентификация

Работа любого субъекта (пользователя или процесса) в АС должна быть идентифицирована системой.

Основным средством аутентификации пользователей в АС должна быть схема "имя пользователя/пароль", при этом система должна допускать возможность расширения процедур аутентификации на основе смарт-карты, touch memory, дискеты. При этом должна присутствовать возможность дифференцированного считывания информации с токена — пароля или ПИН при аутентификации, ключа при шифровании или электронно-цифровой подписи.

Парольная политика должна предусматривать возможность настройки по следующим параметрам:

- ☐ минимальная длина пароля;
- ☐ минимальный срок жизни пароля;
- ☐ максимальное количество ошибок при вводе пароля;
- ☐ время блокировки счета пользователя после достижения заданного количества ошибок;
- ☐ поддержка истории паролей;
- ☐ определение примитивных и слабых паролей;

- ☐ требование ввода старого пароля при замене новым паролем;
- ☐ возможность наличия нескольких (до трех) паролей у одного бюджета пользователя.

К дополнительным требованиям к работе с бюджетами пользователя можно отнести следующие:

- ☐ возможность изменить пароль пользователя, кроме него самого, должен иметь администратор безопасности, однако он не должен иметь возможность получить информацию о старом, заменяемом пароле;
- ☐ при входе пользователя в систему необходимо предупреждение о запрете использования чужих паролей и несанкционированного доступа;
- ☐ при входе пользователя в систему ему должна быть представлена информация о предыдущем успешном входе в систему (дата и время), а также количество ошибок при вводе пароля между двумя последними успешными входами в систему (если таковые были) с указанием даты и времени;
- ☐ бюджет пользователя должен иметь возможность привязки к конкретному рабочему месту (сетевой адрес), к конкретному времени работы по дням недели и часам (отдельно для интерактивной работы и запуска процессов, а также для удаленной работы);
- ☐ бюджет пользователя должен иметь уникальный системный идентификатор, который должен формироваться автоматически при регистрации нового пользователя и должен быть доступен уполномоченному администратору только на чтение;
- ☐ бюджет пользователя должен иметь возможность ручной блокировки уполномоченным администратором без изменения пароля бюджета, а также возможность выставления признака требования немедленной смены пароля пользователем либо запрета смены пароля;
- ☐ бюджет пользователя должен иметь поля для описательной текстовой информации и возможность расширения для привязки дополнительных полей, например, изображения.

Авторизация и доступ

Для определения единообразного уровня доступа субъектов к информационным объектам, являющимся первичной информационной единицей, система должна предусматривать возможность следующего распределения доступа:

- ☐ к группе или нескольким группам объектов;
- ☐ к объекту;
- ☐ к набору частей объекта.

Например, если информационными объектами являются клиенты, то должна присутствовать возможность определения единообразного доступа к клиентам для конкретного пользователя по следующим критериям:

- ☐ конкретный клиент;
- ☐ все клиенты типа А и Б;
- ☐ все клиенты, у которых пользователь является ответственным контролером или у которых ответственные контролеры принадлежат к той же группе, что и данный пользователь;
- ☐ только ряд полей в карточке клиента.

Доступ к информационным объектам должен включать следующие виды ограничений:

- ☐ нет доступа к объекту;
- ☐ доступ (к объекту/части объекта) только на чтение;
- ☐ доступ (к объекту/части объекта) только на изменение — отдельно с чтением и без чтения;
- ☐ удаление информационного объекта;
- ☐ добавление информационного объекта;
- ☐ нет доступа на чтение;
- ☐ нет доступа на изменение;
- ☐ нет доступа на добавление;
- ☐ нет доступа на удаление.

Кроме доступа к конкретным информационным объектам, должно быть предусмотрено разделение доступа к функциям, выполняющим процессы над объектами. Отдельно должны быть выделены функции, выполняющие массовые изменения данных в информационных объектах.

Схема определения доступа должна предусматривать возможность группового или ролевого доступа, т. е. создание абстрактного профиля прав пользователя. При включении нового пользователя в конкретную группу он автоматически должен наследовать все права, присущие пользователям данной группы. При этом суммарные права пользователя должны проверяться на непротиворечивость, должен применяться принцип поглощения более широкими разрешительными правами более узких (суммирование прав) и принцип преимущества запретительных прав над разрешительными. Также должна присутствовать возможность копирования (наследования) прав пользователя с возможностью последующей корректировки.

Для каждого информационного объекта должен существовать владелец объекта (с возможностью в дальнейшем заменить одного владельца на другого) с полными правами доступа. Система должна контролировать обяза-

тельное наличие хотя бы одного пользователя (владельца или администратора) для каждого информационного объекта.

Пользовательский интерфейс настройки прав доступа должен быть прост и интуитивно понятен.

Доступ к банковским счетам

АС должна поддерживать возможность отдельного распределения доступа пользователя:

- ☐ к карточке (реквизитам) счета (тип счета, балансовый код Главной книги, описание, процентные реквизиты, настройки типов комиссий, уровень неснижаемого остатка, лимит овердрафта и другие параметры);
- ☐ к остатку счета (в частности, к операциям со счетом).

Виды доступа к карточке счета: чтение, изменение, открытие счета, закрытие счета, блокировка, снятие блокировки и т. д. Возможно дополнительное разделение карточки счета на несколько непересекающихся групп реквизитов и свойств с независимой настройкой прав на них.

Доступ к остатку должен включать следующие независимо настраиваемые уровни: чтение остатка, дебетовые операции, кредитовые операции, возможность просмотра истории оборотов по счету (с отдельной настройкой прав на просмотр дебетовых и кредитовых оборотов с полными реквизитами корреспондентов или без них, а также на авторизацию операций). Для каждого вида операций система должна предусматривать возможность ограничения максимально разрешенной суммы, установленной индивидуально по этой операции для данного счета, а также возможность скрыть от пользователя информацию о клиенте счета, в том числе и при сохранении права на операции со счетом.

При распределении доступа пользователей к счетам (как к карточке, так и к остатку) должен использоваться удобный интерфейс, позволяющий быстро настраивать права на отдельный счет и на целые группы счетов. Группировать счета можно:

- ☐ по типам счетов (перечень);
- ☐ по кодам Главной книги (в частности, по маскам кодов);
- ☐ по типам и группам клиентов;
- ☐ по валютам;
- ☐ по филиалам;
- ☐ по подразделениям организации;
- ☐ по уровню ответственности (свои счета, т. е. счета, у которых данный пользователь является ответственным контролером, или счета группы, т. е. счета, у которых ответственные контролеры состоят в одной группе с данным пользователем).

Пользовательский интерфейс распределения доступа должен предусматривать и механизмы для указания исключений по аналогичным правилам (например, все счета группы А, кроме счетов, принадлежащих группе В).

Дополнительные права

Система должна предусматривать дополнительные возможности по установке разграничений доступа к информационным объектам или их частям. Примером такого разграничения является понятие овердрафта (принижение минимально допустимого остатка). Необходимо разграничивать права использования овердрафта следующим образом:

- ☐ разрешен ли овердрафт как таковой;
- ☐ размер овердрафта для конкретного счета, для группы счетов;
- ☐ размер овердрафта для данного пользователя;
- ☐ размер овердрафта для данной операции;
- ☐ размер овердрафта для вида валюты.

Конфиденциальность данных безопасности и прочих данных

Информация о критических атрибутах профиля пользователя (пароли, ключи) должна сохраняться в виде, исключающем возможность прямого доступа к ним пользователей, минуя средства системы, причем как для постоянной, так и для оперативной памяти. Это означает, что указанные данные при постоянном хранении должны быть защищены средствами криптографии с предоставлением доступа либо только самому пользователю, либо процессам подсистемы безопасности, либо администратору, уполномоченному подсистемой безопасности. При хранении данных в оперативной памяти должна исключаться возможность получения несанкционированного доступа к ним.

При этом пароли должны сохраняться в виде хэш-значений, а ключи — иметь возможность аварийного доверенного восстановления.

Система должна обеспечивать возможность настраиваемой криптографической защиты данных (вплоть до полного сокрытия всех данных) как при архивном, так и при операционном хранении.

Системы при передаче данных по каналам связи от места хранения до конечного пользователя в рамках локальной сети должны использовать только защищенные соединения, при этом защита должна быть осуществлена минимум на транспортном, предпочтительнее — на сетевом уровне.

Для распределенного варианта работы организации, когда основная база данных находится в центре, а конечный пользователь — в удаленном филиале, требуется наличие у организации защищенных каналов связи. Автоматизированная банковская система при этом должна обеспечивать защиту соединения минимум на прикладном, предпочтительно — на транспортном уровне.

Целостность данных

Автоматизированная банковская система должна поддерживать логическую целостность бизнес-данных, т. е. изменения должны носить характер транзакционно-ориентированных, выполняющихся в целом от начала до конца либо, в случае сбоя, не выполняющихся совсем. При этом каждая транзакция должна проверяться на непротиворечивость в соответствии с настраиваемыми в АС правилами. Прохождение транзакции в АС должно быть разбито на этапы, сопровождаемые настраиваемым количеством подтверждений (ввод, авторизация и т. п.)

Способы авторизации транзакций должны настраиваться по выбору организации:

- ☐ несколько авторизаторов для одного пользователя;
- ☐ один авторизатор для группы пользователей;
- ☐ один авторизатор для данного типа операций.

Автоматизированная банковская система должна иметь настройку на обеспечение апеллируемости на ряд критически важных транзакций путем использования электронно-цифровых подписей. Должна присутствовать возможность варьировать количество подписей на одну транзакцию в вариантах:

- ☐ без подписи;
- ☐ одна подпись;
- ☐ две подписи;
- ☐ три подписи.

Также должна присутствовать возможность включения/исключения полей записей транзакции в/из группы параметров для цифрового подписывания.

Система должна предусматривать формирование шаблона транзакции, когда ряд значимых полей заполняется автоматически, без возможности корректировки пользователем.

Система должна иметь собственную встроенную или внешнюю подсистему проверки целостности файлов, модулей и системных данных самой АС на предмет их несанкционированной модификации.

Никакие системные или прикладные данные не должны удаляться в рамках АС без следов. Должна присутствовать настройка на запрет удаления отдельных категорий данных. Система также должна фиксировать информацию, необходимую для идентификации факта, объекта и субъекта процесса удаления. Система должна предусматривать возможность восстановления удаленных данных в течение определенного промежутка времени.

Доступность данных

Автоматизированная банковская система должна предусматривать возможность устойчивой работы при появлении сбоев, т. е. возможность конфигурации для работы кластера из двух и более географически распределенных узлов. При этом выход из строя некритического количества узлов не должен сказываться на общей функциональности системы.

Процессы, требующие от системы монопольного использования ресурсов и препятствующие интерактивной работе пользователей, должны быть перенесены на время наименьшей активности (ночь) и в сумме занимать не более 10—15% суточного рабочего времени системы.

Система должна быть реализована в трехуровневой архитектуре клиент-сервер с тем, чтобы вывод из строя рабочего места пользователя или получение злоумышленником несанкционированного доступа к нему не сказывался на работе серверной части системы, а сбой сервера приложений не влиял на состояние данных системы.

Система должна обладать возможностью балансирования нагрузки между отдельными узлами, модулями и функциями, а также иметь подсистему предупреждения о работе в режиме, приближенном к максимальной загрузке.

Автоматизированная банковская система должна иметь возможность резервного копирования и восстановления средствами самой системы. Резервное копирование должно осуществляться на различные носители по выбору организации, в том числе в дискретном (разорванном) виде, т. е. общий объем резервной копии может быть распределен между двумя или более носителями меньшего объема.

Резервное копирование должно предусматривать как сохранение прикладных, так и системных данных, и осуществляться в режиме с криптографической защитой по заранее установленному графику от одного до нескольких раз в день либо вручную по команде оператора. При этом процедура не должна забирать ресурсы системы в монопольное использование.

В системе следует предусмотреть механизм облегченного восстановления потерянных в результате сбоя операционных данных со времени последнего резервного копирования.

Удаленная работа

Система должна предусматривать возможность работы пользователей в удаленном режиме по выделенным, коммутируемым каналам либо средствами Интернета. Для этого она должна иметь либо собственную встроенную систему создания виртуальной частной сети, либо возможность тесной интеграции с межсетевым экраном.

В случае реализации защиты соединения на уровне выше сетевого система должна обладать защитой системы маршрутизации сетевых пакетов.

Системы настройки аутентификационных механизмов, авторизации, прав доступа и прочих элементов схемы безопасности должны четко различать варианты локальной и удаленной работы и, соответственно, иметь возможность применения различных правил в зависимости от режима доступа.

Аудит и мониторинг

Система аудита АС должна различать бизнес-значимые и системно-значимые события. Для бизнес-значимых событий должны фиксироваться параметры, которые не отражаются напрямую в прикладных данных системы, например такие, как время запуска процедуры массовой работы со счетами, имя пользователя и сетевой адрес его рабочей станции. Для системно-значимых событий должна происходить фиксация любых изменений, которые так или иначе могут отразиться на работе системы. Такой список должен включать следующие события (но не ограничиваться ими):

- ☐ добавление, удаление пользователя или изменение атрибутов пользователя;
- ☐ изменение настроек криптографической защиты данных;
- ☐ изменение настроек контроля целостности;
- ☐ изменение настроек удаленного доступа;
- ☐ изменение настроек резервного копирования;
- ☐ изменение настроек, влияющих на работоспособность и производительность системы;
- ☐ запуск, остановка процедур системного характера и обмен данными с процедурами системного характера;
- ☐ сообщения администратору, имеющие отношение к безопасности системы.

При этом информация, фиксируемая в регистрационных журналах, должна обеспечивать однозначную идентификацию субъекта, причины, времени, местоположения и результата произведенного действия.

Все бизнес-значимые действия пользователей в АС, как успешные, так и неудачные, начиная от попытки установления связи и до завершения сессии, должны быть зафиксированы. Журнал будет использован для исследования корректности работы АС, мониторинга действий пользователей, расследования сложных или подозрительных событий и т. д.

Все события должны быть разбиты на типы (открытие счета, регистрация клиента, корректировка карточки счета и т. п.) и категории (клиенты, счета, пользователи, системные события, безопасность и т. д.).

Каждый тип событий характеризуется уровнем доступа или тем, интересы каких пользователей могут затронуть события данного типа и для кого они будут доступны при просмотре журнала. Возможны следующие уровни:

- ☐ система в целом — уровень центрального администратора системы (доступны все события в системе);
- ☐ филиал — уровень администратора филиала организации (доступны все события, относящиеся к данному филиалу);
- ☐ сотрудник — уровень сотрудника филиала (доступны только те события, которые относятся к нему или затрагивают его интересы).

Каждое событие должно обладать как минимум следующими реквизитами:

- ☐ автор (пользователь, который породил это событие);
- ☐ время (календарная дата и время, когда событие произошло);
- ☐ категория события;
- ☐ событие;
- ☐ объект события;
- ☐ описание события (конкретные подробности события и объекта);
- ☐ филиал организации, затронутый событием.

При работе с журналом событий должна быть обязательно предусмотрена возможность фильтрации событий, т. е. возможность показывать не все события, соответствующие данному периоду, а только те, которые требуются в настоящий момент. Список параметров фильтрации должен включать все указанные реквизиты событий.

Регистрационные журналы системы должны обладать возможностью защиты от изменений со стороны любого субъекта, кроме процесса его формирования (доступ только на чтение уполномоченным администратором). Доступ на чтение регистрационного журнала должен зависеть от установленного для данного администратора фильтра прав.

Регистрационный журнал должен обладать механизмами обеспечения его юридической доказательности (например, системная или административная цифровая подпись).

Аудит событий должен включать возможность формирования шаблонов событий, подозрительных, похожих на злоумышленные. При наступлении событий, соответствующих такому шаблону, система должна выдавать оповещение уполномоченному лицу.

Система должна предусматривать мониторинг и управление работой пользователей, включая следующие возможности:

- ☐ отображение текущих сессий пользователей в системе;
- ☐ отправку системного или административного сообщения пользователю или группе пользователей;

- ☐ автоматическое завершение сессии пользователя при достижении определенного времени бездействия в системе;
- ☐ ручное принудительное завершение сессии пользователя (группы пользователей) администратором.

Бизнес-отчеты

Система формирования бизнес-отчетов должна укладываться в схему прав пользователя, т. е. пользователь не должен получать в рамках отчета информации больше, чем доступно ему в рамках разрешения на чтение.

Разрешение на получение консолидированных (сводных) отчетов может определяться отдельно.

Интерфейсы

Для обмена данными с другими системами АС должна обеспечивать наличие соответствующих интерфейсов таким образом, чтобы подсистема безопасности АС могла:

- ☐ продолжать контроль экспортируемых данных в рамках правил, определенных в АС, до тех пор, пока контроль над данными не возьмет на себя система, в которую производится экспорт;
- ☐ принимать на себя контроль импортируемых данных от другой системы после получения от этой системы сигнала о готовности импортируемых данных;
- ☐ устанавливать на импортируемые данные правила доступа, авторизации, контроля целостности и т. п. в соответствии с предопределенным шаблоном для импортируемых данных.

Контроль жизненного цикла программного обеспечения

Контроль жизненного цикла программного обеспечения, разрабатываемого внутри организации, необходим для того, чтобы обеспечить уверенность в том, что информационная система на основе данного программного обеспечения функционирует точно так, как предполагалось в техническом задании и не имеет недокументированных возможностей. В связи с этой проблемой возникает такой термин информационной безопасности, как *программная закладка* (англ. *backdoor, logical bomb*).

Широко известны случаи, когда в процессе разработки в программе возникали недокументированные возможности, которые в дальнейшем приводили или могли привести к появлению уязвимости. Иногда такие возможности возникают случайно, по причине не очень квалифицированного програм-

мирования, порой они вводятся на этапе тестирования программы, а затем их просто забывают удалить, иногда они вносятся разработчиком в рабочую версию программного продукта сознательно для последующего использования в собственных целях. Причем, в последнем случае, разработчик не обязательно может преследовать злоумышленные цели.

В качестве примера можно привести вариант, когда разработчику приходится выступать одновременно в качестве консультанта или представителя службы поддержки своего продукта для учреждения организации, географически удаленного от офиса разработчика. Столкнувшись со сложностью удаленной консультации пользователя или администратора, особенно в случае низкой квалификации последних, разработчик может прийти к решению, когда проще всего временно передать значительные полномочия по администрированию системы или работы в ней удаленному абоненту. Для этого в программе следует содержать бюджет пользователя с большими полномочиями, который активизируется секретным паролем или хитроумной комбинацией нажатия клавиш. К сожалению, если подобные механизмы не описаны в документации к программе и их использование соответствующим образом нормативно не упорядочено — это серьезная уязвимость программы, как бы хитро не был замаскирован способ получения таких привилегий.

Естественно, что в случае, когда разработчик вынашивает планы собственного обогащения или мести организации, то он имеет больше возможностей для реализации своих идей в случае неадекватного контроля над его деятельностью. Здесь в качестве примеров можно привести ставшие классическими случаи реализации подобных атак.

Первый — *логическая бомба* — предназначена в основном для мести организации. В этом случае разработчик закладывает в программу механизм, который активизируется в случае, например, его увольнения. Механизм может остановить действие программы, уничтожить данные, внести помехи в работу пользователей. Наиболее опасный способ — внесение мелких, но необратимых изменений в данные, похожих на результаты действий легитимных пользователей.

Второй — имеющий собственное название *атака саями* — заключается в том, что в операциях внутри системы, связанных с разделением денежно-значимых активов на части, производится дополнительное отделение минимальной, практически незначимой части в пользу разработчика. Имевшие место известные случаи из банковской практики — от перевода средств между двумя банковскими счетами, с имеющейся дробной частью от единицы валюты (скажем, в 12 345,67 доллара) отнимался 1 цент в пользу счета, открытого разработчиком в системе для своих целей. При большом количестве таких операций счет разработчика пополнялся существенными суммами.

Каким способом организация может обезопасить себя от возможного произвола разработчиков? Решение не демонстрирует ничего нового — это из-

вестные принципы безопасности *минимальных привилегий* (англ. *least privileges*) и *разделения обязанностей* (англ. *separation of duties*) применительно к разработке и использованию ПО как к полноценному информационному процессу. В данном случае это означает, что процесс разработки программного обеспечения должен быть отделен от распространения, установки, эксплуатации и сопровождения системы.

Типовым сценарием для обеспечения такой работы может быть следующий общий план.

- ❑ В организации создаются отдельные службы: разработки, хранения эталонных копий, внедрения и поддержки.
- ❑ Первая рабочая копия передается службой разработки службе хранения в виде исходных текстов, где проходит тщательный анализ на отсутствие недокументированных возможностей.
- ❑ После успешной верификации отсутствия недокументированных возможностей служба хранения производит компиляцию исходных текстов в исполняемые модули, возможно, подписывает их с помощью ЭЦП, рассчитывает и фиксирует контрольную сумму каждого модуля как файла данных либо применяет другие механизмы контроля целостности программного обеспечения.
- ❑ Скомпилированные модули и средства контроля целостности передаются службе внедрения и поддержки. Исходные коды и эталоны программных модулей сохраняются в специальном хранилище, куда имеют доступ только уполномоченные сотрудники службы, ни в коем случае ни разработчики, ни служба поддержки.
- ❑ Группа внедрения и поддержки обеспечивает регулярную проверку рабочих копий системы на целостность.
- ❑ В следующем цикле разработки, когда группа программирования представляет новую версию с обновлениями/изменениями и соответствующей документацией, служба хранения производит проверку только измененных составляющих, после чего процесс компиляции, фиксации и передачи в работу повторяется. Контроль изменений проще всего осуществлять, если разработка ведется объектно-ориентированным способом или с применением специализированных средств поддержки версий.

Теоретически, в рамках данной схемы злоупотребления могут произойти в службе хранения, если несанкционированные изменения будут внесены в программный код после проверки на недокументированные возможности, но до компиляции. Усилить работу схемы возможно, если служба разработки будет передавать исходные коды службе хранения, а уже скомпилированные модули, службе поддержки (рис. 25.1). Далее возможны два варианта.

- ❑ Служба поддержки передает копии модулей службе хранения. Служба хранения после проверки кодов производит их компиляцию и сравнива-

ет полученные модули с модулями, поступившими от службы поддержки (контрольные суммы, подписи или другие механизмы). В случае полного совпадения служба хранения дает разрешение службе поддержки использовать оставшиеся у них копии модулей в работе.

- ❑ Служба хранения после проверки кодов производит их компиляцию и также передает копии модулей в службу поддержки. Служба поддержки сравнивает обе копии модулей и в случае их совпадения производит фиксацию контрольных параметров и т. п., использует модули в работе.

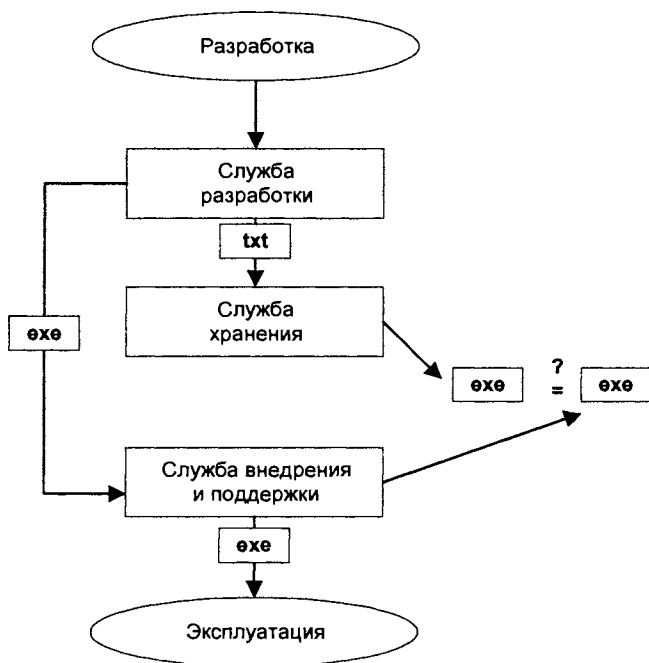
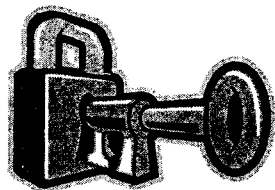


Рис. 25.1. Примерная схема процесса разработки/хранения/внедрения

Если на каком-либо этапе было обнаружено несовпадение модулей, несоответствие кодов документации, несоответствия контрольных сумм или цифровых подписей, необходимо проведение расследования на предмет несанкционированного изменения функциональности системы.

При обеспечении строгого следования указанной схеме единственной возможностью для осуществления злого умысла, разработчику остается несанкционированная подмена рабочего модуля на месте работы. В этом случае измененный модуль будет функционировать до следующей проверки модулей на целостность службой поддержки. Таким образом, сокращая периоды проверки модулей (скажем, до ежедневных, перед началом работы пользователей) возможно практически полностью избежать данной угрозы.

Глава 26



Работа с персоналом и оборудованием

Методическая работа с персоналом и тщательная проверка всего парка используемого оборудования являются очень важными профилактическими мероприятиями в плане информационной безопасности. Рассматриваемые в данной главе вопросы частично пересекаются с материалом других глав, однако авторы решили выделить данное направление. Это связано с тем, что речь в основном пойдет не о специализированных продуктах информационной безопасности, а о знакомых каждому пользователю принтерах, мониторах и другом неспецифичном оборудовании.

Работа с персоналом

Общие вопросы

Говоря о безопасности персонала, необходимо разделять два понятия — безопасность самого персонала как человеческого и информационного ресурса и защиту от персонала как источника или основы злоумышленных воздействий на информационные системы.

Вопросам защиты персонала в случаях стихийных бедствий будет уделено внимание в *главе 31*. Нападения, угрозы, шантаж и прочие воздействия злоумышленников на персонал — это объект уголовного права и в общем случае находится в компетенции уполномоченных государственных органов, с которыми служба информационной безопасности организации может сотрудничать, в любом случае играя второстепенную роль. Для конкретной организации ситуация может быть иная, и такие вопросы будут решаться собственной службой безопасности, однако все равно используемые в данном случае оперативные, розыскные и иные мероприятия выходят за рамки книги.

Иной взгляд авторы имеют на персонал как возможный источник злонамеренных воздействий на информационную безопасность предприятия либо, когда некомпетентность, ошибки, халатность или иные действия персонала

послужили основой для угрозы безопасности организации. Причем угрозы безопасности по причине некомпетентности и низкой квалификации персонала в зависимости от региона, социального состава и других причин, могут иметь значимость не меньшую, а иногда и большую, чем угрозы безопасности по причине злого умысла.

Прием на работу



Начало анализа персонала начинается, соответственно, с этапа приема на работу. Подразделение по работе с персоналом имеет собственные критерии отбора людей, но служба безопасности может и должна принимать в этом активное участие. На наш взгляд, такая оценка складывается из двух частей — предыдущего опыта и текущей квалификации.

Что касается предыдущего опыта, то здесь должны быть отработаны системы сбора информации (профессиональный опыт, знания, персональные качества, характеристики с прежних мест работы как официальные, так и неформальные), возможно, контакты со службой безопасности с прежнего места работы кандидата; анализа информации — какие качества являются положительными, а какие отрицательными. Скажем, готовность кандидата поделиться особенностями организации производства на прежнем месте работы — это положительная или отрицательная характеристика? С одной стороны, от него ожидается, что весь накопленный опыт он применит на новом месте, с другой — где гарантия, что при очередной смене места работы он не будет рассказывать о той организации, куда теперь принимается на работу?

Текущий статус кандидата должен проверяться с точки зрения его технической подготовки и способности работать с информационными системами, а также психологического склада и черт характера для выявления тенденций развития личности (в том числе как злонамеренных, так и просто создающих неудобства при общении в коллективе). Возможно участие в этом процессе штатного или приглашенного психолога.

Одним из примеров простого теста для проверки способностей кандидата является тест на пароль. Для него необходимо небольшое программное обеспечение, которое имитирует приглашение к вводу пароля в информационную систему, сохраняет пароль и может продемонстрировать его представителю службы безопасности. Возможно, перед испытанием кандидату следует дать ознакомиться с требованиями организации к паролям пользователей. Далее кандидату предлагается придумать и ввести собственный пароль (например, мотивируя тем, что ему будет предоставлена возможность тестовой работы в системе). После ввода пароля можно попросить кандидата еще раз войти в систему, но при этом система выдаст ему сообщение об ошибке с предложением обратиться в службу безопасности, либо сообщит, например, что время его последнего входа не соответствует тому, ко-

торое реально было сделано только что, или что со времени последнего входа было две-три попытки ввода неверного пароля. Настораживающими факторами в данном случае являются следующие.

- ❑ Слишком долгие размышления по придумыванию пароля. Комбинацию из букв, цифр и спецсимволов нормальный пользователь с опытом работы в информационных системах сможет придумать за 30—60 секунд либо вообще использовать один из своих старых паролей. Долгие раздумья скорее всего означают, что кандидат впервые сталкивается с такими требованиями к безопасности и, возможно, в дальнейшем он будет испытывать трудности с их соблюдением. Правда, долгие раздумья могут означать, что он действительно изобретает сложный пароль, но это, возможно, будет проверять после его ввода.
- ❑ Попытки запроса подсказок пароля типа "Помогите мне подобрать пароль". Это означает, что пользователь не осознает важность сохранения пароля в секрете. Дополнительной проверкой может послужить провокация проверяющего, например: "Скажите, какой пароль вы сейчас ввели?" — ведь кандидат предупрежден, что получает вход в систему, пусть и в тестовом режиме.
- ❑ Ошибки при подтверждении пароля. Если пользователь не может сосредоточиться настолько, чтобы два раза ввести одинаковую комбинацию из 10—15 символов, то это может быть негативным знаком. При этом следует делать поправку на возможное волнение кандидата при проверке.
- ❑ Слишком простой введенный пароль. Если кандидат, ознакомившись с требованиями организации к паролям, все же игнорирует их, надеясь, что пароль не будет никому известен, то это также негативная тенденция.
- ❑ Отсутствие реакции на тревожные сообщения системы при последующем входе в нее. Если система проинформировала пользователя об ошибке при вводе пароля, а затем пропустила его, а также дала неверную дату/время последнего входа или сообщила о неверных вводах пароля, которых не было со стороны пользователя, — это всё сигналы, о которых кандидат должен проинформировать службу безопасности.

Если кандидат принят на работу и становится реальным пользователем — субъектом информационного пространства, то служба информационной безопасности должна обеспечить себе правовое пространство для дальнейшей работы с данным пользователем, в том числе для случаев возможного преследования данного пользователя при выявлении с его стороны нарушений. Для этого необходимо произвести следующие мероприятия.

- ❑ Ознакомить с существующими нормативами (правилами, инструкциями, методическими указаниями и т. п.) в сфере информационной безопасности и обеспечить уверенность, что в дальнейшем он не сошлется на их незнание, в том числе взять подписку.

- ❑ Если кандидат заключает с организацией контракт, то в него без сомнения должен войти ряд положений по информационной безопасности, особенно если они не отражены адекватно в законодательстве государства. Например, что субъект обязуется признавать цифровую подпись руководителя наравне с обычной подписью на бумаге.
- ❑ Если организация выдвигает повышенные требования к сохранению конфиденциальности своей информации, возможно, это также необходимо включить в список подписываемых субъектом положений. При этом следует соблюдать действующее законодательство страны, с тем чтобы возлагаемые на субъекта обязательства не нарушали его конституционных прав.
- ❑ Если организация собирается использовать элементы вмешательства в частные разделы рабочей деятельности сотрудника, например, регулярно или выборочно перлюстрировать его электронную почту или отслеживать посещения интернет-сайтов, то работник должен быть как минимум об этом предупрежден.
- ❑ Если сотруднику предоставляется в его распоряжение персональный компьютер, то необходимо довести до сотрудника, что данная единица техники остается собственностью организации, а значит, организация будет определять политику его использования. При этом установка на компьютер игр, обучающих программ, музыки и видеофильмов чаще всего не входит в цели организации. А поскольку данный сотрудник становится ответственным за данный компьютер, он должен обеспечивать в рамках своих возможностей соответствие требованиям нормативов, в том числе по информационной безопасности.
- ❑ Если за данным персональным компьютером будет работать два и более человек, то целесообразно назначить одного, наиболее подготовленного, ответственным за (или контролирующим) соблюдение требований информационной безопасности на данном компьютере.

Методическая работа с персоналом



Несмотря на любую квалификацию пользователя, поскольку развитие информационного пространства предприятия не стоит на месте, необходимо регулярное обучение сотрудников как работе в новых информационных системах, так и мерам безопасности в этих системах. Кроме того, развитие злоумышленной части мирового информационного пространства может потребовать от пользователей дополнительных более или менее квалифицированных знаний по ряду атак (например, как в случае с известным вирусом Kournikova — отличать расширения jpg от jpg.vbs). Это означает, что следует продумать систему оперативного оповещения сотрудников о необходимых мероприятиях, а также возможное проведение регулярных семинаров для всех или наиболее подготовленной части сотрудников. Для больших, рас-

пределенных корпораций желательно применять способ подготовки подробных методических материалов, их рассылки и обязательного ознакомления сотрудников с ними.

Как бы старательно и качественно не работали сотрудники, периодически приходится проверять их деятельность. Проверки могут быть выборочными или регулярными, их диапазон может простираться от анализа регистрационных журналов данной конкретной информационной системы до полного сканирования содержимого жесткого диска и внешних носителей пользователя. При таких проверках, по всей видимости, будут периодически обнаруживаться какие-то нарушения в работе пользователей, в том числе и по их вине. О том, как производится расследование инцидентов, речь пойдет в *главе 33*, а вот степень зависимости вины пользователя в том или ином нарушении и размер наказания должны определить уполномоченные лица конкретной организации.

Хотелось бы дать один совет: не всегда суровое наказание даже за маленький проступок — это хорошее решение проблемы. По опыту известно, что даже при информированности пользователей о том, что их деятельность — это предмет наблюдения соответствующих служб, значительная часть из них все равно будет пытаться искать мелкие выгоды для себя в процессе своей рабочей деятельности. Возможно, это будет посещение развлекательных сайтов в Интернете или установка компьютерных игр. В таких случаях напоминание в виде электронного письма с примерным содержанием: *"Вчера вы посетили сайт www.xxx.com и провели на нем 42 минуты рабочего времени. Рекомендуем вам во избежание неприятностей не посещать сайты, не связанные с вашей рабочей деятельностью. Служба безопасности"* — будет хорошей профилактикой развития злоупотребления корпоративными ресурсами.

Однако даже если данный незначительный инцидент формально "прощен", тем не менее он должен быть зафиксирован в некоем "черном списке", куда вносятся все нарушения информационной безопасности пользователем с первого дня работы. Такой документ полезен, во-первых, поскольку пользователи склонны забывать о своих нарушениях (в этом случае данный документ будет полезен при некоем переполнении списка мелких нарушений, условно прощенных для этого пользователя). Во-вторых, с помощью данного документа можно отслеживать тенденции в развитии работы пользователя с точки зрения информационной безопасности.

Мелкой, но тем не менее важной деталью является вопрос идентификации работника при межличностном общении. В организации, насчитывающей несколько десятков человек, специалист по безопасности может знать каждого сотрудника лично, но в большой организации, где персонал составляет 1000 и более человек, вопросы распознавания личности, особенно для новых работников, станут уязвимостью, которая может быть использована, например, средствами социальной инженерии, о которой шла речь в *главе 10*.

Ближайшим примером является обращение пользователя по телефону с просьбой замены пароля по той или иной причине. Каким образом проверить соответствие реального пользователя названному username? Если пользователь находится неподалеку, наиболее простой способ — пригласить его прибыть к администратору с документами, подтверждающими личность. Кстати, для этих целей хорошим корпоративным стандартом считается ношение на груди опознавательного пропуска с необходимой информацией (желательно продумать вопросы воспрепятствования их подделке). Но что, если пользователь географически удален от администратора, а вопросы бизнеса требуют обеспечения немедленного начала работы в информационной системе? В этом случае вступает в действие система временных паролей для однократного входа (с немедленной заменой), для которой в Интернете разработаны два основных варианта решения проблемы.

- ❑ **Контрольный вопрос.** При регистрации в учетную запись пользователя, доступ к которому есть только у администратора, заносится вопрос и соответствующий ответ, которые известны только данному пользователю. Таким образом, перед сменой пароля администратор аутентифицирует пользователя по ответу на контрольный вопрос. Если система позволяет, следует сохранять несколько вопросов для одного пользователя, также следует заменять вопросы, после их использования.
- ❑ **Альтернативная обратная связь.** После обращения пользователя администратор фиксирует номер телефона и, проверив корректность номера, перезванивает пользователю сам. Кроме того, временный пароль высылается пользователю по электронной почте, возможно, даже не самому пользователю, а его непосредственному руководителю.

Следует только учесть важность того, чтобы временный пароль был достаточно уникальным, с тем чтобы в каждый конкретный момент не было известно о том, какой временный пароль использован для данного работника.

Бывший кадровый состав предприятия



Другой важный момент — взаимоотношения с сотрудником, который собирается покинуть организацию. Естественно, должны быть проведены некоторые технические мероприятия по блокированию или удалению учетных записей данного сотрудника как пользователя информационных систем. Но для этого служба безопасности как минимум должна быть оповещена о предстоящем увольнении сотрудника. Также в этой ситуации необходим контакт с подразделением по работе с персоналом.

Для ряда увольняющихся сотрудников, а возможно и для всех, целесообразно проводить собеседования, в ходе которых напомнить о продолжении действия обязательств, взятых на себя сотрудником по отношению к организации (если заключенные контракты и договоренности не теряют своей силы при увольнении сотрудника). Возможно, остаются еще какие-либо

дополнительные заинтересованности уходящего сотрудника в организации. В любом случае предпочтительнее сохранить хорошие отношения между уходящим сотрудником и организацией.

Служба безопасности организации должна решить, как она будет поступать с запросами на характеристики бывших сотрудников организации, устраивающихся на работу на новое место. Возможно, предприятие считает, что оно не должно предоставлять такие сведения, но тогда следует учесть, что и на аналогичный запрос самого предприятия на нового кандидата может прийти отказ в ответе. По общим соображениям целесообразнее организовать взаимодействие между службами безопасности предприятий отрасли, предварительно определив перечень данных, которые могут быть сообщены коллегам. Подобное сотрудничество при правильной организации позволит только улучшить состояние информационной безопасности отдельного предприятия.

Работа с оборудованием



Безопасность оборудования также складывается из вариантов, когда оборудование само представляет угрозу и когда оборудование является объектом, на который направлена угроза.

Когда организация приобретает оборудование информационных технологий, она вынуждена ему доверять, так как в большинстве случаев организация не имеет достаточного количества ресурсов для анализа сложной техники на отсутствие недокументированных возможностей — *аппаратных закладок* (англ. *hardware backdoor*). Причины возникновения таких закладок — это вопрос конкуренции между предприятиями или политики отдельных государств, но в любом случае он вне рассмотрения данной книги. Даже если в организации присутствуют квалифицированные специалисты по информационным технологиям или информационной безопасности, они не возьмут на себя смелость подтверждать или опровергать наличие аппаратных закладок. В отличие от методик и средств обнаружения программных закладок в большинстве случаев для анализа аппаратного обеспечения необходимы специальные лаборатории и персонал отдельной высокой квалификации.

По этой причине оборудование необходимо приобретать только после сертификации уполномоченным органом государства, либо если таковой орган отсутствует — у известных производителей (брендов), которые дорожат репутацией и не заинтересованы в скандалах по поводу аппаратных закладок в их технике. В данном случае увеличение цены оборудования — это плата за гарантию отсутствия "пятой колонны" внутри предприятия.

Если предположить, что в оборудовании нет встроенных со злым умыслом недокументированных возможностей, в нем тем не менее может присутст-

вывать уязвимость для информационной безопасности. Чаще всего в связи с такой уязвимостью употребляют термин *ПЭМИН* — *побочное электромагнитное излучение и наводки* (англ. *side-channel*). Данное явление основывается на том, что при своей работе оборудование выделяет в окружающее пространство неконтролируемым образом некоторую информацию в том или ином виде, причем эту информацию можно перехватить, обработать и использовать несанкционированным образом. Наибольшее внимание в информационных технологиях привлекают ПЭМИН, возникающие в средах передачи данных (проводах, кабелях и т. д.) и генерируемые при работе мониторов пользователей. Однако прежде чем приобретать сложное и дорогостоящее оборудование для защиты от ПЭМИН следует произвести анализ рисков.

- ❑ Возможно, достаточно обеспечить шифрование данных при передаче по линиям связи, в этом случае не только сложная аппаратура улавливания электромагнитных наводок, но и прямое внедрение перехватчика пакетов ничего не даст злоумышленнику.
- ❑ Для удаленного перехвата информации с экрана монитора требуется очень дорогостоящее и сложное оборудование. А что получит самый злобный конкурент, если он даже сможет заглядывать через плечо наиглавнейшему системному администратору? Возможно, он сможет получить эту информацию более легким и простым путем.

Следует учесть, что обычно анализируются риски для среднего предприятия. Если говорить, например, об учреждениях министерства обороны, где в электронном виде хранится информация класса "государственная тайна", то рассуждения будут, конечно, совсем другими.

Допустим, обеспечена ситуация, когда все оборудование организации функционирует надежным и доверенным способом. Однако присутствует возможность внедрения в информационное пространство предприятия постороннего оборудования, принесенного сотрудниками и посетителями.

В данном случае оборону необходимо эшелонировать двумя ступенями: контролировать внос оборудования в здание организации и проверять отсутствие посторонних включений в информационную сеть. И для первого, и для второго способов существуют различные средства и методы — от визуального осмотра до автоматизированного обнаружения.

Отдельного внимания заслуживают в данном случае внешние носители (которые также можно отнести к оборудованию), а также средства для чтения/записи на внешние носители и порты для внешнего подключения таких устройств. В данном случае внешние носители могут быть средством доставки зловредных информационных объектов/субъектов в информационное пространство предприятия и средством изъятия конфиденциальной информации из информационных объектов предприятия.

Наиболее надежным, хотя и непростым решением будет физическое отключение/удаление устройств считывания/записи и блокировка/отключение свободных портов (COM, LPT, USB, инфракрасная связь и т. п.). При этом необходимо создание отдельной службы, в распоряжении которой останется одно или несколько устройств работы с внешними носителями. Физически размещать эту службу целесообразно недалеко от входа в здание, с тем чтобы все вносимые и выносимые внешние носители могли быть представлены ее сотрудникам. В функции службы будет входить:

- ☐ проверка отсутствия на принесенном внешнем носителе зловредных элементов (вирусов, троянских программ, постороннего программного обеспечения);
- ☐ копирование или электронная доставка информации с принесенного внешнего носителя на рабочую станцию или сервер, в распоряжение того пользователя, для которого информация предназначена;
- ☐ получение от пользователя информации, предназначенной к выносу из организации на внешнем носителе;
- ☐ проверка отсутствия в информации для выноса конфиденциальных данных и запись информации на внешние носители.

Единственным допустимым исключением для этой службы может быть только служба регулярного резервного копирования критической информации с отдельным регламентом работы.

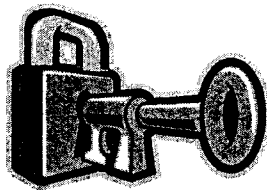
Заканчивая разговор о внешних носителях, необходимо упомянуть о проблеме стирания информации с внешних носителей, после того как информация была доставлена по назначению. Вопрос затрат на возможное восстановление якобы уничтоженной информации с внешнего носителя, — это вопрос стоимости самой информации, которую необходимо восстановить. В любом случае следует учесть, что многократная перезапись или физическое разрушение корпуса или даже записываемой поверхности — это еще не гарантия невозможности восстановления. Если вопрос о гарантированном уничтожении информации достаточно острый, то следует проанализировать возможность приобретения специальной аппаратуры. При этом необходимо различать аппаратуру с полным разрушением внешнего носителя и с возможностью его последующего использования.

Другой важной особенностью оборудования информационных технологий является наличие большого количества мелких деталей, которые обладают рядом неприятных для информационной безопасности качеств:

- ☐ они самоценны, т. е. имеют отдельную стоимость по отношению к остальному оборудованию;
- ☐ соотношение цена/масса/размер некоторых деталей в определенные периоды развития рынка превышает аналогичное соотношение, например, даже для золота;

- ❑ они могут быть заменены на похожие детали худшего качества и меньшей стоимости, при этом замена не будет очевидна для конечного пользователя;
- ❑ они пользуются устойчивым спросом на рынках, где не требуются документы, подтверждающие легальное происхождение данных деталей.

Наиболее ярким примером в данном случае могут быть платы оперативной памяти. Только квалифицированный пользователь сможет заметить отличие в работе компьютера с 128 или с 96 Мбайтами оперативной памяти. Таким образом, пропажа или подмена может быть не выявлена достаточно долго, до очередной подробной инвентаризации (при отсутствии автоматизированных средств инвентаризации). Приемлемым средством снижения данного риска будет использование кожухов для сокрытия мелких деталей (практически у всего оборудования такие кожухи присутствуют) и опечатывание кожухов печатью уполномоченного подразделения — скажем, информационной безопасности или технической поддержки. При этом важно, чтобы целостность печати регулярно проверялась самим пользователем оборудования, в противном случае эта мера будет носить лишь фиктивный характер.



Часть VII

Обеспечение нормативно-правового пространства

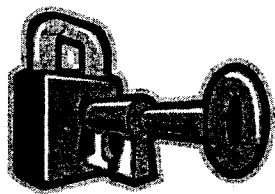
Глава 27. Создание политики безопасности

**Глава 28. Работа с пользователями,
администраторами и разработчиками**

Глава 29. Законы и нормативы

**Глава 30. Тестирование процедур
и механизмов безопасности**

Глава 27



Создание политики безопасности

Создание политики безопасности — одно из первых требований к организации информационной безопасности предприятия. Еще могут быть не приобретены автоматизированные средства безопасности, не сконфигурировано оборудование и не обучены пользователи, но то, как все это будет происходить, организация и ее руководство должны знать и одобрять. В данной книге рассмотрение вопросов создания политики безопасности вынесен ближе к концу изложения. Сделано это только потому, что перед тем, как браться за разработку политики специалист или группа, которая будет работать над созданием документов, должны получить представление о принципах, механизмах и средствах безопасности. В противном случае в дальнейшем работа по безопасности будет расходиться с нормативными документами и служба информационной безопасности окажется в правовом беспределе, что не сможет не сказаться на уровне доверия к службе и состоянию информационной безопасности в целом.

Структурирование документа или пакета документов, называемого "политикой", и его детализация могут быть различными. Мы предлагаем следующую опробованную на практике схему. Политика должна быть представлена пакетом из следующих документов.

- ❑ *Концепция*, описывающая общие принципы и подходы в организации информационной безопасности. Именно этот документ должен быть понят и одобрен руководством предприятия, так как он составляет основу для построения всей информационной безопасности организации.
- ❑ *Стандарты* информационной безопасности предприятия. В документе будут даны основные понятия — определения, термины и построены общие модели, необходимые для обеспечения информационной безопасности предприятия.
- ❑ *Процедуры информационной безопасности*. В этом документе указывается, что должно быть сделано и делаться в данной организации для того, чтобы уровень информационной безопасности соответствовал Стандартам информационной безопасности.

- ❑ **Методы информационной безопасности.** Это документ конечного пользователя с точки зрения информационной безопасности. В нем указывается, как будут выполняться процедуры информационной безопасности конечными пользователями.
- ❑ **Аварийный план.** Документ, посвященный тому, что делать, если произойдет нарушение информационной безопасности в том или ином виде, повлекшее некий существенный ущерб предприятию и процесс восстановления нормальной деятельности.

Рассмотрим содержание каждого из документов подробнее.

Концепция



Структура документа

В общей части или введении документа можно указать назначение его и всего пакета документов по информационной безопасности с указанием его структурированности. Дополнительно можно указать структуру службы безопасности как основного инструмента информационной безопасности предприятия, ее иерархию и подчиненность.

Необходимо указать правовую основу политики и всех работ по информационной безопасности, т. е. дать ссылку на государственные и отраслевые нормативные акты (законы, указы, постановления, положения и порядки), а также на уже существующие в организации документы, которые будут использованы или затронуты настоящим пакетом по безопасности. Причем здесь должны быть упомянуты не только документы, связанные с безопасностью напрямую, но и с информатизацией вообще, а также другие полезные нормативы, возможно, до Уголовного и Уголовно-процессуального кодексов.

В документе должны быть определены цели и задачи информационной безопасности. Иногда понятия целей и задач смешивают. На наш взгляд, в данном случае разделение должно быть следующим. Цели — это описание идеальной ситуации, к которой следует стремиться организации для наилучшего состояния информационной безопасности, а задачи — это набор общих мероприятий, которые необходимо произвести или регулярно производить для того, чтобы максимально приблизиться к намеченной цели.

Целесообразно в таком документе указать отношения между данной конкретной организацией как крупным субъектом глобального информационного пространства с другими внешними и внутренними субъектами:

- ❑ с клиентами организации, если организация работает с клиентами;
- ❑ с вышестоящими организациями;

- ☐ с другими организациями отрасли или посторонними организациями;
- ☐ с собственными подразделениями — отделениями, филиалами и т. д.;
- ☐ с собственными сотрудниками.

В заключение можно указать перспективы развития информационной безопасности вообще или данного документа в частности.

Примеры: выдержки из концепции предприятия

Рассмотрим предприятие под условным названием АБВ.

Цели и задачи

Целями информационной безопасности АБВ являются:

- ☐ создание безопасного информационного пространства для осуществления АБВ своей деятельности как самостоятельно, так и в сотрудничестве с другими субъектами и системами экономической и другой деятельности;
- ☐ соблюдение интересов клиентов, вышестоящих организаций и партнеров в вопросах обеспечения информационной безопасности;
- ☐ совершенствование информационного пространства в сфере действия информационной безопасности.

Основными задачами информационной безопасности АБВ являются:

- ☐ определение и классификация информационных объектов, на основе которых АБВ осуществляет свою деятельность, изменение и дополнение списка данных объектов с учетом изменяющихся условий;
- ☐ учет программного, аппаратного и другого информационного обеспечения объектов, которое создает, хранит, обрабатывает, пересылает или уничтожает информационные объекты;
- ☐ разработка способов формирования защищенной информационной среды, совершенствование данных способов в соответствии с изменением состояния мировой информационного пространства;
- ☐ разработка механизмов оценки и сокращения ущерба от результатов нарушений информационной безопасности во всех аспектах.

Взаимоотношения с другими субъектами

В соответствии с принятым определением АБВ как информационная система принимает участие во взаимодействиях со следующими субъектами и системами:

- ☐ клиент АБВ как субъект, формирующий основу для создания информационных объектов и являющийся их конечным владельцем;

- ❑ отдельный работник АБВ как субъект, формирующий информационные объекты и работающий с ними;
- ❑ подразделение АБВ (управление, отдел, рабочая группа и т. п.), как структура, составленная из отдельных субъектов;
- ❑ вышестоящие организации как информационная система, получающая от АБВ и направляющая в АБВ различные информационные объекты;
- ❑ другие организации как субъекты и системы, обменивающиеся с АБВ информационными объектами.

АБВ — клиент

АБВ ставит своей задачей соблюдение интересов клиентов в вопросах их безопасности при выполнении работ.

АБВ сохраняет в конфиденциальности все данные по своим клиентам и их деятельности, как это указано в следующих законах (следует перечень законов и нормативных актов).

АБВ предоставляет сведения о своих клиентах только тем организациям, которые имеют право на получение таких данных.

АБВ соблюдает интересы своих клиентов в вопросах безопасности, если это не вредит интересам самого АБВ.

В свою очередь АБВ вправе требовать от клиентов предоставления достоверной и своевременной информации, необходимой для корректного и безопасного осуществления деятельности АБВ, а также соответствия деятельности клиента требованиям безопасности, которыми руководствуется АБВ, а также сохранения в тайне тех сведений о АБВ, которые закреплены в договорных обязательствах с клиентом.

АБВ — вышестоящие организации

С одной стороны, АБВ в своей деятельности руководствуется требованиями законов и других нормативных актов, нормативными актами вышестоящих организаций, в частности инструкциями по правилам информационной безопасности главной организации отрасли.

С другой стороны, АБВ разрабатывает самостоятельно методы и процедуры информационной безопасности в соответствии с мировыми (государственными) стандартами.

АБВ — другие организации

Вступая в любые виды экономических отношений с другими организациями, АБВ, с одной стороны, следует их требованиям безопасности, если они не понижают уровень требований к соблюдению мер безопасности самого АБВ.

С другой стороны, АБВ вправе требовать увеличения уровня безопасности со стороны своего партнера, если данный уровень не соответствует требованиям АБВ.

АБВ готов оказать помощь своим партнерам в организации должного уровня безопасности при защите финансово-экономических интересов.

АБВ — подразделения АБВ

АБВ требует от своих подразделений поддержания уровня безопасности, соответствующего общему стандарту АБВ.

Специалисты по различным видам безопасности центрального офиса АБВ оказывают консультационную помощь и другие виды поддержки сотрудникам подразделений АБВ в вопросах безопасности.

Клиент, обращаясь в любое подразделение АБВ, вправе ожидать, что его интересы будут защищены на одинаково высоком уровне безопасности.

АБВ — сотрудники АБВ

АБВ требует от своих сотрудников, как рядовых работников, так и руководителей различных уровней, соблюдения всех мер безопасности в интересах АБВ, клиентов и экономики государства.

Сотрудники АБВ регулярно проходят и будут проходить обучение различным аспектам правил безопасности, необходимым для поддержания общего уровня безопасности АБВ на должном уровне.

Стандарты

Структура документа

В общей части или введении стоит указать структуру документа для упрощения его понимания, т. е. кратко описать его назначение и содержание.

Во избежание разночтений при использовании формулировок и терминов начало документа необходимо посвятить определениям, таким как — объект, субъект; информационная система и пространство; информационное, аппаратное, программное обеспечение; несанкционированное действие, конфиденциальность, целостность, доступность и все другие термины, которые будут использованы в пакете документов.

В данном документе необходимо представить результаты инвентаризации и классификации информационных объектов. Если Инвентаризатор и Классификатор — это отдельные документы, то следует дать на них ссылку, а в текущем документе указать общую схему инвентаризации и классификации.

Далее следует указать стандарты работы в информационном пространстве организации, т. е. что считается нормальной работой субъектов с объектами, не нарушающей безопасность. Именно здесь следует подчеркнуть, что информационные системы предприятия со всем аппаратным и программным обеспечением — это собственность предприятия и предмет примене-

ния правил использования, в том числе и правил безопасности. Здесь следует распределить порядок ответственности за информационные системы и объекты.

Кроме того, в данном документе следует определить информационные угрозы, расставить приоритеты и составить модель злоумышленника. Такие специфические, на первый взгляд, только для информационной безопасности задачи, могут тем не менее оказать существенное влияние на развитие информационных систем предприятия. Например, определение в качестве основной, угрозы целостности может позволить пренебречь (в определенной мере) угрозами конфиденциальности и доступности. Или определение злоумышленника строго как внешнего по отношению к организации субъекта позволит не использовать усиленную систему аутентификации или аудита для внутренних субъектов.

Примеры: выдержки из стандартов предприятия

В качестве примера рассмотрим возможные критерии классификации объектов и связанную с процессом классификации терминологию.

Объекты и подобъекты защиты

Существование объектов

Каждый информационный объект имеет свой жизненный цикл, который состоит из следующих стадий:

- ☐ создание объекта субъектом;
- ☐ хранение объекта (в качестве хранителя или хранилища, вообще говоря, может выступать некий субъект);
- ☐ передача объекта от хранилища к месту обработки и обратно;
- ☐ обработка объекта (модификация, уничтожение, создание нового объекта из данного).

В соответствии с данным жизненным циклом определим основные действия с объектом:

- ☐ как хранение;
- ☐ передачу;
- ☐ обработку (создание, модификация, уничтожение).

Классификация объектов и субъектов

Классификацию субъектов информационной деятельности произведем следующим образом:

- ☐ субъект, создающий объекты;
- ☐ субъект, использующий объекты;

- ❑ субъект, администрирующий объекты, т. е. обеспечивающий среду работы с объектами;
- ❑ субъект, контролирующий использование объектов.

Каждый конкретный субъект может совмещать в себе несколько или все указанные классификационные сущности.

Классификацию важности информационного объекта определим исходя из триединой сущности способов обработки информации с точки зрения информационной безопасности, а именно:

- ❑ объект должен быть доступен только тем субъектам, кто имеет на это право (данную сущность назовем *конфиденциальность*);
- ❑ вносить изменения в состояние объекта могут только те субъекты, которые имеют на это права (*целостность* или *достоверность*);
- ❑ для субъектов, обладающих необходимым уровнем прав, объект должен быть доступен в течении необходимого для работы времени (*доступность*).

Стандарты использования средств вычислительной техники

Принадлежность и ответственность за средства вычислительной техники

Средства вычислительной техники (СВТ), объекты информационного пространства АБВ и сама информационная сеть, включая аппаратное, программное, сетевое, организационное, нормативное обеспечение, являются собственностью АБВ и переданы сотрудникам АБВ во временное пользование для выполнения служебных обязанностей.

По этой причине сотрудники АБВ, получившие доступ к работе на СВТ, несут ответственность за работоспособное состояние самого СВТ, а также за результаты своей работы в информационной сети. Для обеспечения нормального функционирования СВТ и всей сети работник должен строго следовать правилам, порядкам и другим нормативным документам, регламентирующим соответствующие части и разделы работы сотрудников.

Перед получением доступа к работе на СВТ в информационную сеть или к какому-либо ресурсу сети, работник АБВ должен пройти инструктаж по правилам информационной безопасности, а также заполнить обязательство. В обязательстве работника должно быть указано, что он ознакомлен с правилами, готов им следовать, а также осознает возложенную на него ответственность.

В случаях, когда сторонние субъекты (специалисты, консультанты, подрядчики и т. п.) в соответствии с документальными соглашениями с АБВ получают доступ в информационную сеть АБВ, к конкретному сетевому ресурсу

или СВТ, они (субъекты) должны быть ознакомлены с правилами информационной безопасности АБВ и письменно подтвердить готовность им следовать.

В случаях, когда в целях обучения, демонстрации и т. п. доступ в информационную сеть АБВ к конкретному сетевому ресурсу или СВТ получают субъекты, не имеющие документального соглашения с АБВ (стажеры, представители других организаций и т. п.), такой доступ должен происходить только в присутствии и под наблюдением сотрудника АБВ, который назначен для этих целей руководителем подразделения, взаимодействующего с данным субъектом. В этом случае все действия субъекта должны выполнять только заранее оговоренным функциям и строго соответствовать правилам информационной безопасности. При любых нарушениях ответственность ложится на руководителя подразделения, допустившего указанную работу субъекта.

Определение ответственного за средства вычислительной техники

Стандартом АБВ является определение одного ответственного за каждую единицу информационного обеспечения или, иначе говоря, за средство вычислительной техники, т. е. даже если данное средство вычислительной техники используют в работе несколько человек, полную ответственность за сохранение ее работоспособности и соблюдение мер безопасности на ней несет один человек — *ответственный сотрудник*.

Если СВТ снабжено периферийными устройствами, то ответственный сотрудник за данное СВТ является ответственным и за все периферийные устройства, соединенные с данным СВТ.

При увольнении ответственного сотрудника, переходе его в другое подразделение, перераспределении ответственности за СВТ внутри подразделения, смене сетевого имени СВТ, получении нового СВТ или перемещении СВТ в другое помещение, не позднее чем в недельный срок начальник отдела должен произвести переназначение ответственного сотрудника.

При временном отсутствии (болезнь, командировка и т. д.) сотрудника, ответственного за СВТ, ответственность за данное СВТ автоматически принимает на себя начальник подразделения (в его отсутствие — исполняющий его обязанности), если только начальником подразделения не назначен другой ответственный за данное СВТ. Таким образом, в подразделении не должно быть ситуаций, когда за каким-либо СВТ не закреплен ответственный. Если начальник подразделения не назначил ответственных сотрудников — он отвечает за все СВТ своего подразделения.

Каждому начальнику подразделения следует назначить ответственного за "СВТ начальника подразделения", в отсутствие начальника подразделения, либо принять меры к тому, чтобы в его отсутствие работа на данном СВТ была невозможна.

Ответственный сотрудник в случае работ на закрепленном за ним СВТ других сотрудников обязан осуществлять контроль за сохранением данного СВТ в состоянии, отвечающем требованиям правил, после завершения на нем работ других сотрудников.

Обязанности ответственного

Ответственный сотрудник должен обладать знаниями, достаточными для того, чтобы поддерживать СВТ в работоспособном состоянии. Решение о компетентности конкретного сотрудника принимает начальник его подразделения, при необходимости консультируясь со специалистами по информатизации.

Ответственный сотрудник должен следить за работоспособностью закрепленного за ним СВТ. При сбоях в функционировании СВТ он должен немедленно обратиться к начальнику своего подразделения для консультаций со специалистами по информатизации или оформления заявки на ремонт/проверку данного СВТ.

Ответственный сотрудник несет ответственность за принятие решения об установке нового программного обеспечения, за удаление или изменение какой-либо информации на данном СВТ. Соответственно он несет ответственность за наличие на СВТ программного обеспечения иного, чем установленное специалистами информатизации, уничтожение или изменение системных файлов и настроек.

Ответственный сотрудник обязан производить необходимые профилактические работы на вверенном ему СВТ: периодический и целевой запуск антивирусных программ, удаление ненужной информации и запуск процедуры проверки целостности и чистки жесткого диска и др. При необходимости он должен обращаться за консультациями и обновлением антивирусного программного обеспечения к специалистам информатизации.

Ответственный сотрудник должен присутствовать при любых работах, производимых на его СВТ специалистами информатизации, сторонними специалистами или другими сотрудниками АБВ. При временном отсутствии ответственного согласно настоящему документу при выполнении работ должен присутствовать начальник подразделения.

Учет информационного обеспечения

Стандартом АБВ является строгий учет каждой единицы информационного обеспечения и основных его параметров и настроек.

Подразделению автоматизации необходимо обеспечить ведение учетного архива всех единиц СВТ, размещенных в распоряжении данного отделения или филиала. При изменении настроек СВТ, перемещении, смене ответственного за СВТ необходимо оперативно обновить данные учетного архива.

Процедуры



Структура документа

В зависимости от подробности охвата документ может разделяться на две, три или более частей, а именно — процедуры для обычных пользователей, процедуры для администраторов, процедуры для специалистов по безопасности, процедуры для разработчиков и т. д. Причем, с одной стороны, следует проанализировать — все ли эти части могут быть доступны для всех пользователей. Возможно, рядовым пользователям не стоит знать, каким образом производится администрирование систем или контроль их использования. С другой стороны, возможно, организация считает, что работа всех субъектов системы должна быть прозрачна.

Рекомендуется разделять процедуры на *первоначальные процедуры* (получение доступа в информационное пространство организации или к конкретной информационной системе); *общие процедуры*, которые характерны для всех пользователей сети (работа с аппаратным обеспечением, с операционной системой); *специфические рабочие процедуры* (работы в отдельных информационных системах, администрирование, контроль и т. д.), *завершающие процедуры*, связанные с временным или постоянным прекращением работы пользователя с конкретной информационной системой по специальным должностным функциям или в организации вообще.

Важным акцентом процедур должно быть разделение полномочий и ответственности между различными службами. В крупных организациях возможно появление конфликтов интересов, в том числе и рабочих, например, между пользователями и администраторами или между администраторами и контролерами (специалистами по безопасности) в следующих вопросах.

- ❑ Для обеспечения оперативности и удобства администрирования, особенно удаленных рабочих станций, администраторы имеют повышенные права доступа к информационным объектам пользователей. Часто пользователи неодобрительно относятся к наличию таких возможностей по причинам чисто психологическим (кто-то без их ведома может что-то делать на их компьютере), но кроме того существует еще и опасность того, что администратор получит доступ к бизнес-информации, которая для него не предназначена.
- ❑ Администраторы довольно неохотно делятся своими правами в информационных системах с контролерами (специалистами по безопасности), которые в свою очередь стремятся получить максимально допустимые права для контроля деятельности администраторов. Претензии администраторов обоснованы, так как именно они несут обычно ответственность за работоспособность систем, а посторонние несогласованные вмешательства в настройки, даже для усиления безопасности системы,

могут быть вредны, особенно если настройка системы была достаточно специфична.

Подобные конфликты должны быть учтены и процедуры взаимного согласования, оповещения и использования ресурсов прописаны особенно тщательно.

Примеры: выдержки из процедур предприятия

Процедуры получения доступа к работе в информационной сети

Решение о необходимости получения тем или иным работником АБВ доступа к информационной сети принимает непосредственный начальник данного работника, основываясь на квалификации и уровне подготовки работника. Он готовит рапорт на имя начальника подразделения информатизации за подписью начальника подразделения, к которому относится данный работник.

Начальник подразделения информатизации принимает решение о допуске данного работника к работе в информационной сети, основываясь на возможностях имеющихся в данный момент информационных мощностей. В случае одобрения указанный работник проходит процедуры регистрации, получения идентификатора и пароля и становится пользователем информационной сети.

Получение статуса пользователя означает, что данный работник имеет право работать на средствах вычислительной техники, получать доступ к сетевым ресурсам и будет нести ответственность за все действия в сети, совершенные с его идентификатором.

Решение о прекращении пользователем работы с данным ресурсом или прекращении доступа к информационной сети производится в следующих случаях.

- ❑ Пользователь увольняется из АБВ. В этом случае из информационной сети удаляются все сетевые и ресурсные счета, ассоциированные с данным пользователем.
- ❑ Пользователь меняет свои функциональные обязанности (переходит в другое подразделение). В этом случае доступ к сети и ресурсам может быть приостановлен до получения рапорта от начальника нового подразделения пользователя о необходимости работ в сети и/или с ресурсами на новом месте работы.
- ❑ Пользователь длительное время отсутствует на рабочем месте (отпуск, болезнь, командировка). В этом случае доступ к сети и ресурсам может быть приостановлен до прибытия пользователя на рабочее место, о чем он должен проинформировать подразделение автоматизации.

Во всех этих случаях подразделение по работе с персоналом не позднее чем в недельный срок должно проинформировать подразделение информатизации.

- ☐ Пользователь нарушил требование правил пользования сетью, требование информационной безопасности и/или нанес ущерб (подверг угрозе нанесения ущерба) информационным ресурсам АБВ. В этом случае доступ пользователя к сети и/или ресурсам закрывается до окончания проведения расследования и принятия решения начальником подразделения информатизации.

При длительном единовременном отсутствии пользователя на рабочем месте, его функции могут быть переданы другому работнику подразделения по усмотрению руководителя подразделения. В этом случае готовится рапорт на имя начальника подразделения информатизации за подписью начальника подразделения, к которому относится данный работник, с кратким описанием причин необходимости передачи функций, списка ресурсов, к которым необходим доступ, а также срока, на который доступ предоставляется. Запрещается проводить передачу функций одного работника другому путем передачи пароля.

Аппаратно-технические меры безопасности

Для обеспечения информационной безопасности на вверенной ему единице средства вычислительной техники пользователь должен в первую очередь обеспечить надежную работу данного СВТ доступными ему способами. Это необходимо для того, чтобы избежать сбоев аппаратного и программного обеспечения в результате механических воздействий на СВТ, неправильного включения в сеть и т. п.

Перед включением компьютера и началом работы необходимо произвести внешний осмотр места его установки. Категорически запрещается загораживать специальные вентиляционные отверстия на панелях системного блока и монитора. Необходимо убедиться, что СВТ включено в сеть гарантированного электропитания.

Включение осуществляется в следующем порядке:

- ☐ системный блок (кнопка включения питания расположена на лицевой панели);
- ☐ монитор (кнопка включения питания расположена на лицевой панели);
- ☐ принтер и другое периферийное оборудование.

Выключение должно осуществляться в обратном порядке, т.е. принтер (или другое периферийное устройство), монитор, системный блок.

Запрещается производить включение кондиционеров, обогревателей и другие посторонние электротехнические устройства в электрическую сеть гарантийного питания, предназначенную для компьютеров, а также размещать их поблизости от СВТ.

Запрещается изгибать или ломать кабель локальной вычислительной сети (ЛВС), проходящий в каждом помещении, и другие кабели, соединяющие различные части СВТ. Нельзя отключать кабель, соединяющий СВТ и розетку ЛВС. Нельзя допускать повреждения розеток ЛВС и любых кабелей СВТ, например, при перемещении стульев и другой мебели.

Запрещается ставить посторонние предметы (скрепки, чашки, сумки и т. п.) на корпус компьютера, монитор и клавиатуру, другие периферийные устройства. Нельзя самовольно подключать или отключать СВТ, его блоки или узлы к источникам/от источника питания и тем более производить самостоятельный ремонт СВТ. Запрещается самостоятельно перемещать компьютер и отдельные его блоки. При необходимости перемещения техники вызывать специалистов подразделения информатизации. Нельзя размещать рядом с компьютером любое другое электронное оборудование, кроме установленного специалистами отдела технического обслуживания. Категорически запрещается курить в помещениях, где находятся компьютеры. При долговременном отсутствии на рабочем месте необходимо выключать монитор.

Системно-технические меры безопасности

Важным фактором информационной безопасности СВТ является обеспечение такого состояния СВТ, которое препятствовало бы возможности посторонних лиц получать доступ к данным, находящимся на СВТ в выключенном состоянии или во время отсутствия пользователя, в том числе и путем физического изъятия носителей информации (например, жесткого диска или дискет) или других частей СВТ.

Ключ от процессорного блока компьютера должен храниться в подразделении информатизации. Процессор (системный блок) следует опечатывать печатью подразделения безопасности. Пользователь СВТ должен ежедневно перед началом работы проверять целостность оттисков печати, о его повреждениях докладывать начальнику подразделения и в подразделение безопасности.

На компьютере должен стоять пароль на включение компьютера (загрузку). Процедуру установки пароля выполняет специалист подразделения информатизации, пароль вводится пользователем самостоятельно, индивидуально и секретно.

На компьютере должен стоять пароль на режим ожидания работы (Screen Saver), который включается сотрудником при временном отсутствии на рабочем месте или автоматически через 3—5 минут. В качестве режима ожидания необходимо выбирать Screen Saver с затемнением экрана, как наиболее ресурсосберегающий для монитора. Следует избегать программ Screen Saver с меняющимися или движущимися картинками, как перегружающие монитор и процессор лишними функциями. Способ установки режима ожидания описан в документе "Методы информационной безопасности предприятия".

Для получения доступа к различным информационным ресурсам и к самой сети используется система паролей для подтверждения правомочности доступа. Для избежания разглашения (компрометации) паролей необходимо придерживаться следующих правил:

- ☐ не передавать и не сообщать никому свой пароль, при попытках других лиц узнать пароль или при прямом принуждении к этому (в том числе со стороны руководителей подразделения) необходимо обращаться в службу информационной безопасности;
- ☐ не использовать примитивные, легко разгадываемые пароли;
- ☐ для доступа к различным информационным ресурсам иметь различные пароли, не вычисляемые один из другого, тем более не использовать одни и те же пароли для доступа к внутренним данным сети АБВ и при использовании внешних информационных систем (например, Интернет);
- ☐ при начале работы с информационный объектом (ресурсом), при запросе на ввод пароля необходимо прочитать в верхней строке имя ресурса, который запрашивает пароль, если наименование не совпадает с именем ресурса, следует выяснить, не является ли это попыткой перехвата пароля;
- ☐ не записывать пароли там, куда может получить доступ посторонний;
- ☐ не использовать возможность запоминания пароля системой;
- ☐ по возможности не использовать свои пароли для доступа к ресурсам сети с компьютеров других пользователей — на чужих компьютерах могут быть установлены программы перехвата паролей;
- ☐ периодически менять пароли доступа самостоятельно, если позволяет данный сервис, либо обращаться в подразделение информатизации для смены пароля.

Если пользователь на данном СВТ производит работы с электронными документами (файлами), которые содержат закрытую информацию, ему следует принимать меры к более строгому ограничению доступа к данной информации. Такой мерой может служить установка на СВТ дополнительного программного обеспечения для шифрования. Решение о необходимости использования шифрования принимает начальник подразделения. В этом случае следует обращаться в подразделение информационной безопасности для установки указанного программного обеспечения и получения инструкций для работы с ним.

В случае, когда возникает необходимость проверки того, какую зашифрованную информацию пользователь сохраняет, пересылает, получает или использует иным образом, создается комиссия, в которую входят: руководитель подразделения данного пользователя, представитель службы информационной безопасности, службы информатизации. Указанная комиссия прибывает на

рабочее место пользователя (место хранения зашифрованной информации), после чего выполняются следующие действия:

- ☐ производится включение СВТ пользователя;
- ☐ пользователь демонстрирует комиссии все свои зашифрованные файлы, при необходимости производится сканирование всего СВТ для обнаружения всех файлов;
- ☐ по выбору комиссии пользователь демонстрирует содержимое отдельных файлов или всех файлов, производя их дешифровку, при необходимости запуск на исполнение программных файлов;
- ☐ в случае обнаружения данных, которые определяются комиссией как не соответствующие функциональным обязанностям пользователя, производится их полное уничтожение на данном СВТ, результаты заносятся в акт;
- ☐ в случае, когда пользователь отказывается продемонстрировать содержимое файлов (произвести запуск на исполнение), он отстраняется от выполнения работ на данном СВТ до завершения расследования, само СВТ блокируется отдельным паролем или изымается в подразделение информационной безопасности;
- ☐ в случаях, когда пользователь не может самостоятельно продемонстрировать содержимое, запустить файл, ссылаясь на их повреждение или другие причины и нет возможности произвести их восстановление на месте, производится изъятие данных файлов (программного обеспечения) в подразделение информационной безопасности на внешних носителях или по сети с одновременным полным уничтожением указанных данных на СВТ пользователя. В этом случае подразделение информационной безопасности прилагает усилия к восстановлению данных. В случае успешного восстановления данных и определения их соответствия рабочим функциям пользователя данные возвращаются на место изъятия в восстановленном состоянии;
- ☐ в случае, когда пользователь высказывает сопротивление удалению файлов на своем СВТ, он отстраняется от выполнения работ на данном СВТ до завершения расследования, само СВТ блокируется отдельным паролем или изымается в подразделение информационной безопасности.

Для того, чтобы быть уверенным, что программное обеспечение СВТ выполняет только те действия, которые необходимы пользователю, любое программное обеспечение должно проходить тестирование у соответствующих специалистов по информатизации. Поэтому пользователям запрещается самостоятельно устанавливать любое программное обеспечение, которое не проверено и зарегистрировано в подразделении информатизации.

Специалисты подразделения информатизации производят настройку каждого СВТ таким образом, чтобы она выполняла необходимые работы максимально производительно, а также была защищена с точки зрения безопасности. Поэтому пользователям запрещается самостоятельно вносить изменения в системные настройки СВТ. Настройки системы, которые поль-

зователь может производить самостоятельно, описаны в документе "Методы информационной безопасности предприятия".

Меры безопасности при работе в отдельных сервисах

Помимо соблюдения общесистемных мер безопасности при работе на СВТ, существуют отдельные, специальные меры безопасности для работы в прикладных информационных сервисах.

Сервис Internet

Интернет предназначен для получения сотрудниками АБВ оперативной информации по вопросам, связанным с рабочей деятельностью. Установка Интернета для каждого конкретного сотрудника производится по заявке начальника соответствующего подразделения после рассмотрения заявки руководством.

Для обеспечения нормального функционирования сервиса пользователь должен:

- ☐ использовать Интернет только для служебных целей;
- ☐ не заходить на Веб-сайты, предоставляющие платные услуги, без согласования с начальником подразделения;
- ☐ не осуществлять подписку на получение интернет-информации и услуг, без согласования с начальником подразделения;
- ☐ не распространять в Интернете информацию, касающуюся работы АБВ или его сотрудников, в том числе о себе;
- ☐ не создавать и не использовать собственных почтовых ящиков, Веб-страниц и других информационных активов в Интернете;
- ☐ запрещается заходить на Веб-сайты, содержащие эротико-порнографические материалы и игровые программы;
- ☐ запрещается переносить (копировать) на свой компьютер картинки, художественную литературу, музыкальные и видеоклипы, архивированные файлы и любое программное обеспечение.

Методы



Структура документа

Методы представляют собой подробнейшие инструкции того, как производить ту или другую процедуру и, по этой причине, могут представлять постоянно пополняемый документ, который приобретает все новые и новые части по мере добавления и обновления информационных систем в организации. Кроме того, сюда могут быть включены другие инструкции и рекомендации, например, по выбору пароля.

Данный документ обычно предназначен для рядовых конечных пользователей, не имеющих сложных специальных знаний в области информатизации и информационной безопасности, и должен быть написан доступным для них языком. Для администраторов и квалифицированных специалистов по безопасности документ может представлять перепечатку руководств по работе в отдельных информационных системах. Тем не менее, если это необходимо, возможно сделать ссылку из данного документа на указанные руководства.

Примеры: выдержки из методов предприятия

Установка режима ожидания компьютера

Режим ожидания (Screen saver) для Windows 2000 (русская версия).

- ☐ На панели задач нажать кнопку **Пуск**, затем выбрать подменю **Настройка**.
- ☐ Выбрать пункт **Панель управления** (откроется окно со списком разделов настройки).
- ☐ В окне **Панель управления** выбрать пиктограмму с названием **Экран** и дважды щелкнуть (левой кнопкой мыши) (откроется окно с параметрами настройки дисплея).
- ☐ Выбрать вкладку **Заставка**.
- ☐ Из раскрывающегося списка выбрать произвольный тип экранной заставки.
- ☐ Установить флажок **Защита паролем**.
- ☐ Нажать кнопки **Применить** и **ОК**.

Для временного блокирования работы компьютера необходимо одновременно нажать три клавиши <Ctrl>+<Alt>+<Delete>, после чего следует на появившейся панели выбрать кнопку **Блокировка** или нажать клавишу <Enter>.

Открытие и закрытие доступа к каталогам

Для Windows 2000.

- ☐ Нажать двойным щелчком (левой кнопки мыши) на значке **Мой компьютер**.
- ☐ В открывшемся окне **Мой компьютер** выбрать диск "C:".
- ☐ В появившемся окне (C:) будет представлен список всех каталогов (директорий) компьютера. Для того чтобы открыть требуемый каталог, установить на нем курсор и выполнить двойной щелчок левой кнопкой мыши. Для выбора каталога, свойства которого необходимо изменить, установить на нем курсор и щелкнуть один раз левой кнопкой мыши.

Примечание

Каталоги, к которым ранее уже был открыт доступ, изображены в виде руки, протягивающей папку.

- ☐ В окне (**C:**) открыть меню **Файл** (в строке меню) и выбрать пункт **Свойства**.
- ☐ В появившемся диалоговом окне **Свойства** выбрать вкладку **Доступ**.
- ☐ Для закрытия доступа к каталогу установить переключатель напротив опции **Отменить общий доступ к этой папке**, для открытия доступа — **Открыть общий доступ к этой папке**.

При открытии доступа к каталогу установить в строках: сетевое имя каталога (по умолчанию такое же как на диске), количество пользователей, имеющих возможность одновременно получать доступ к данному каталогу, затем нажать кнопку **Разрешения**.

- ☐ Удалить группу **Все** (отметив ее, нажать кнопку **Удалить**).
- ☐ Нажать кнопку **Добавить**.
- ☐ Выбрать пользователя или группу пользователей, для которых данный каталог будет доступен.
- ☐ По завершении выбора нажать кнопку **Добавить** и, установив необходимый уровень доступа в разделе **Разрешения**, нажать кнопку **ОК**.
- ☐ Вернувшись в окно **Свойства**, выбрать закладку **Безопасность** в верхней части окна.
- ☐ Если необходимо, нажать кнопку **Добавить** и установить доступ к каталогу необходимому количеству пользователей аналогично тому, как это было сделано в закладке **Доступ**.
- ☐ Нажать кнопку **ОК**.
- ☐ Нажать кнопки **Apply** и **ОК**.
- ☐ Закрыть все окна к обратной последовательности.

Выбор наименования пользователя и пароля: выбор пароля

Поскольку комбинация наименования пользователя и пароля, связанного с этим наименованием однозначно воспринимается любой информационной системой как указанный пользователь, то и все действия, совершенные в системе будут зарегистрированы как действия данного пользователя. Если пользователь легкомысленно отнесся к выбору пароля и посторонний человек смог получить пароль, то данный пользователь несет всю полноту ответственности за все действия, совершенные посторонним лицом под видом указанного пользователя.

Неудачно выбранным можно считать пароль, при выборе которого использовалось следующее:

- ☐ любое собственное имя;
- ☐ любое слово, которое есть в словаре;
- ☐ идентификатор пользователя;
- ☐ любая дата рождения;
- ☐ один и тот же символ, повторенный несколько раз (например, "aaaaaa");
- ☐ любая строка, длина которой не превышает восьми символов;
- ☐ строка символов, расположенных последовательно на клавиатуре (например, "qwerty" или "asdfgh");
- ☐ серийный номер конкретного компьютера или стоящего рядом с ним;
- ☐ первый попавшийся конкретный пароль из любого руководства по выбору хорошего пароля;
- ☐ пароль, установленный на компьютере кем-либо другим.

Надежный пароль должен включать в себя достаточное количество случайных символов, однако при этом он должен легко запоминаться. Приведем примеры хороших паролей:

- ☐ несвязанная пара слов, разделенная какими-либо знаками или цифрами (например, "5dog86mis");
- ☐ произносимый набор букв или бессмысленная фраза длиной не менее восьми символов (например, "roluminos");
- ☐ случайная строка символов в формате автомобильных знаков некоторых государств (например, "598LS461");
- ☐ инициалы и телефонный номер человека, с которым вы никогда не встречались, взятый из справочника (например, "IV330115").

Последние рекомендации могут показаться недостаточными при выборе надежного пароля для специалистов по безопасности. Однако, во-первых, это пример, легко расширяемый до повышенных требований. Во-вторых, следует помнить, что требование для рядового пользователя запоминания длинных и сложных, со спецсимволами, паролей может привести к записыванию паролей на мониторе или клавиатуре компьютера и свести на нет всю построенную парольную политику.

Аварийный план



Аварийному плану будет посвящена *глава 31*. Здесь отметим только, что он должен состоять из трех частей.

- ☐ *Предварительные мероприятия*, которые проводятся в виде подготовки к возможным инцидентам нарушения безопасности.

- ❑ *Оперативные мероприятия*, которые производятся в момент, когда нарушение безопасности находится в стадии реализации или сразу после него, когда производится расследование инцидента.
- ❑ *Возвращение к обычной деятельности* с погашением, сокращением или возмещением полученного ущерба, разбором ситуации и причин ее возникновения и т. п.

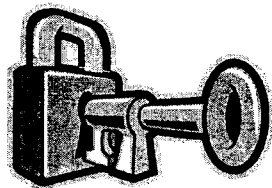
Политики по отдельным направлениям



Можно встретить другой подход к организации политик безопасности, когда концепция и стандарты, скажем, объединены в одну общую политику информационной безопасности предприятия, а процедуры и методы воплощены в виде отдельных политик по работе с информационными системами. Часто такой подход можно увидеть в западных стандартах, особенно для автоматизированных средств безопасности — антивирусная политика, политика использования межсетевого экрана (другое название — политика доступа из локальной сети/в локальную сеть предприятия), политика использования системы обнаружения атак (куда могут входить элементы аварийного плана — анализ и прекращение атак во время совершения, расследования инцидентов и т. д.).

В любом случае это должны быть живые документы, состоящие не из общих фраз, а отражающие специфику работы данного предприятия, которая не может быть отражена в обычной инструкции к отдельному средству безопасности или информационной системе.

Глава 28



Работа с пользователями, администраторами и разработчиками

В зависимости от того, как служба информационной безопасности позиционирует себя по отношению к остальным субъектам информационного пространства, ее работа может быть более или менее эффективна. Она может стать карательным органом для остальных сотрудников предприятия либо защитником их интересов.

Вопрос этот напрямую не связан с техническими особенностями функционирования информационных систем, однако, может оказать определенное, иногда существенное влияние на обеспечение информационной безопасности. Он также связан с построением модели злоумышленника, о которой шла речь в *главе 9*. Статистика показывает, что значительная часть нарушений информационной безопасности производится либо самими сотрудниками организации, либо при их участии. По этой причине очевидно, что "внутренний враг" должен в той или иной степени войти составляющей частью в модель злоумышленника.

В общих чертах специфика проблемы сводится к тому, рассматривается ли каждый работник как злоумышленник или нет. Здесь очень важно не перегнуть палку и не превратить всех пользователей, администраторов и т. д. в потенциальных злоумышленников, очень многое зависит при этом от персонала службы безопасности и ее руководителя. Всегда следует помнить, что бизнес первичен, а безопасность по отношению к нему вторична (речь идет о коммерческих учреждениях, а не о армейских и специальных учреждениях). В первую очередь организация должна работать, а затем уже заботиться о безопасности. В этом смысле необходимо построить работу по безопасности таким образом, чтобы она не мешала ведению бизнеса.

Рекомендации по взаимодействию с персоналом

Работа с пользователями



Известно, что самый безопасный корабль — тот, который стоит на берегу, а, с другой стороны, проще всего работать, когда у сотрудников нет никаких ограничений и сторонних обязательств, не связанных напрямую с бизнес-деятельностью (необходимость запоминать пароли, выполнять сложные процедуры входа в информационную систему, производить периодические мероприятия по повышению уровня безопасности и т. д.). Идеальная ситуация для службы безопасности — когда все компьютеры выключены, сеть отрезана от внешнего мира, помещения пусты, заперты и опечатаны. Для остальных работников идеальная ситуация представляется прямо противоположно — в виде полного отсутствия требований безопасности. С такими противоречивыми целями построение безопасности — сложный процесс, почти искусство.

На наш взгляд, служба безопасности должна выступать не как цербер, принуждающий работников выполнять сложные, непонятные мероприятия, а как помощник, берущий на себя значительный груз по обеспечению безопасности, в том числе данного конкретного работника, и ожидающего, что работник будет сотрудничать с ней, а не противодействовать ее требованиям.

Очень важно наладить прямую и обратную связь с пользователем, приучить его в случае любых подозрительных или просто непонятных событий консультироваться с представителями службы безопасности. Служба безопасности в ответ должна демонстрировать, что обращения пользователя не пропадают в "черной дыре", а приносят свои, пусть небольшие результаты. Очевидно, что при таком подходе появляется большая вероятность ложных срабатываний. Пример тому — обращение пользователя в антивирусную службу в случае появления сообщения Doctor Watson при сбоях приложений в среде Windows. Однако нельзя, основываясь на низкой квалификации пользователя в области информационных технологий, игнорировать или принимать за значимость подобных сообщений. Лучше иметь полную картину информационного пространства, с лишними или даже второстепенными деталями, чем не иметь ее вообще.

Хороший контакт с пользователями можно наладить во время семинаров или занятий по основам информационной безопасности, которые следует проводить регулярно, а также при беседах или интервью при приеме на работу.

Работа с администраторами



Администраторы информационных систем, как правило, специалисты высокой квалификации, с неудовольствием относящиеся к вторжению в сферу их работы, особенно когда целью вторжения является контроль работы администрато-

ров с вероятностью последующего наказания за какие-либо нарушения. Грубое вмешательство в их работу, особенно со стороны специалиста по безопасности с более низкой квалификацией в какой-либо конкретной области, чем у администратора (а такое встречается достаточно часто, так как администратор обычно проводит гораздо больше времени за изучением возможностей конкретной информационной системы), может привести к сопротивлению со стороны данной категории работников управления. Полбеда, если сопротивление будет пассивным в виде нежелания делиться знаниями и навыками по работе системы. Намного хуже, когда обиженный администратор, для того чтобы выставить специалиста по безопасности в глупом виде специально начинает создавать уязвимости и понижать безопасность данной системы.

Разумным средством в таком случае будет привлечение администратора к разработке модели безопасности данной информационной системы как эксперта и консультанта. Необходимо предложить ему самому определить наиболее тонкие и уязвимые места системы и выслушать его предложения по усилению их безопасности. Действия специалиста по безопасности (обычное название контролера системы безопасности информационной системы) данной системы необходимо построить по возможности так, чтобы не только он сам контролировал действия администратора, но и администратор мог при желании проверить действия специалиста по безопасности, например, по регистрационному журналу системы. Возможно, наилучшим решением будет обратный способ, когда администратор имеет возможность любых действий (согласно его функциональным обязанностям) в системе, однако все его действия есть предмет для анализа специалистами службы безопасности (с помощью, скажем, регистрационных журналов). В общем случае это может быть выражено формулой: "Администратор — полный доступ, специалист по безопасности — доступ на чтение". Естественно, необходимо также обеспечить независимость журнала аудита, но это уже детали.

В случаях когда администратор был вовлечен в процесс проработки модели безопасности, он сам становится участником процесса контроля работы системы и обычно следует рассчитывать на его сотрудничество. Если же в процессе дальнейшего изучения системы будет обнаружено, что администратор целенаправленно утаил или даже создал уязвимость, то это достаточная характеристика для постановки вопроса о целесообразности его работы в должности администратора.

Дополнением к сказанному будет то, что все мероприятия по контролю администратора (или взаимному контролю администратора и специалиста по безопасности) должны быть сопровождаемы соответствующими нормативными документами. Все это снизит риск возможного конфликта и обеспечит качественную обстановку для функционирования информационной безопасности.

Работа с разработчиками



Разработчики отличаются от администраторов тем, что вторые работают в системах, разработанных третьей стороной; о механизмах и схемах функционирования таких систем можно узнать из документации, курсов, у производителя и т. д., а первые сами являются создателями механизмов функционирования систем и в смысле информационной безопасности имеют больше возможностей для манипулирования ресурсами системы.

В *главе 25* рассматривались способы обеспечения безопасности разработки приложений. В данном разделе мы только заострим внимание на необходимость также вовлекать разработчиков в процесс построения моделей безопасности. Вернее наоборот — специалист по информационной безопасности должен стать составной частью рабочей группы или проекта по разработке нового программного обеспечения. При этом он должен обладать достаточной квалификацией для чтения кода, навыком работы с отладчиком (дисассемблером), поскольку, возможно, функциональность некоторых, наиболее важных частей кода, таких как стадия аутентификации, авторизации, аудита, придется проверять построчно. Также необходимо совместно с разработчиками проработать процедуры тестирования механизмов безопасности, причем учесть возможность перспективного проведения независимой сертификации продукта на безопасность третьей стороной.

Вопросы подчинения и взаимодействия служб



Как видим, интеграция службы безопасности с остальными субъектами направлена на осуществление контроля со стороны службы безопасности, причем администраторы и разработчики должны сами принимать участие в изобретении механизмов их контроля. Этот процесс должен быть двусторонним. Известно, что бесконтрольность даже по отношению к самым доверенным лицам, провоцирует неустойчивые личности к возможным злоупотреблениям. Кроме того, создавать лестницу "проверяющих над проверяющими" также не всегда целесообразно, поскольку в этом случае верхняя служба все же находится вне контроля. На наш взгляд, наилучшим способом будет организовать возможность перекрестной проверки, скажем, контролеров (специалистов по безопасности) и администраторов, причем проверка контролерами администраторов носит характер обязанности, а проверка администраторами контролеров носит характер возможности. То есть она не входит в обязанности администраторов, но тем не менее может проводиться по его усмотрению регулярно или выборочно. В данном случае важную роль играет не сама проверка, а именно ее возможность, что дисциплинирует контролирующего.

Примером может быть контроль корпоративного использования Интернета. Сотрудник, который проверяет, чтобы работники организации не использовали корпоративный канал для посещения развлекательных и других, не связанных с работой, Веб-сайтов, может также иметь доступ к Интернету. Таким образом, необходимо фиксировать его интернет-активность в регистрационном журнале, который должен быть доступен на чтение, например, администратору домена доступа к Интернету или другому администратору. Соответственно, администратор с правами контролера в любой момент сможет проконтролировать то, как сам контролер Интернета использует вверенный ему ресурс. Если для анализа регистрационных журналов сервиса используется какое-либо дополнительное программное обеспечение, оно также должно быть в распоряжении администратора. Очевидно, что и для проверок администратором контролера также необходимо установить определенный регламент.

Еще одним важным моментом является размещение службы информационной безопасности в структуре организации. Конкретное местоположение службы в иерархии коммерческой структуры может быть различным и зависеть от множества факторов — особенностей самого бизнеса и способов его ведения, традиций организации, местности и страны, персональных предпочтений и качеств руководителя организации, руководителя службы и многих других факторов. Однако есть вопросы, на которые стоит обратить внимание при планировании структурного размещения.

- ❑ Службы информационных технологий и информационной безопасности не должны иметь принципиально различных целей, в положении о подразделениях необходимо указать взаимоподдерживающие, согласовывающие и дополняющие черты.
- ❑ Подчиненность служб руководителям, находящимся в персональном конфликте, есть отрицательный фактор.
- ❑ Целесообразно (при существующих возможностях) размещать ключевых специалистов безопасности и администраторов в одном здании, на одном этаже, в соседних помещениях, обеспечивать совместное их участие в семинарах, обучении и т. п.
- ❑ Уполномоченные специалисты службы информационной безопасности должны иметь возможность прямого обращения к руководителю организации или его ближайшему заместителю.

Для обеспечения эффективной работы возможны более причудливые комбинации, когда специалист по информационной безопасности находится в штате подразделения информационных технологий и по обычной деятельности (трудовой распорядок, командировки и т. д.) подчинен начальнику ИТ, но по инцидентам безопасности напрямую подотчетен только руководителю организации.

В любом случае тот или иной формат работы службы информационной безопасности должен быть строго регламентирован и нацелен на общие со всей организацией цели в бизнесе.

Все работники организации должны быть уверены, что служба безопасности проводит те или иные мероприятия (особенно, связанные с вторжением в персональное рабочее пространство и, тем более, репрессивные) не по своей прихоти, а в соответствии с правилами и инструкциями, которые одобрены руководством и доступны для ознакомления остальным работникам.

В завершение, хотелось бы обратить внимание на важность фиксации и регистрации взаимодействия между различными службами (субъектами) организации. Это целесообразно, как для отслеживания тенденций в использовании информационных систем, так и для анализа в рамках информационной безопасности. В качестве примера, приведем выдержки из документа — регламента оформления заявок на те или иные работы подразделения информатизации. Известно, что наиболее частый заказ на работы ИТ-подразделения идет от рядовых пользователей. Формализация этого процесса (вне зависимости — в бумажном или электронном виде) позволит оптимизировать процесс работы ИТ-подразделения и даст основу (в том числе правовую) для ряда работ службы информационной безопасности.

Выдержки из типового регламента заявок в службу информационных технологий



Рассмотрим примеры регламента оформления заявок на предприятии под условным названием АБВ.

Создание заявки на выполнение работ

Решение о необходимости проведения работ, связанных с информационными системами (ИС), в подразделении АБВ принимает начальник подразделения. При этом в зависимости от характера работ, которые необходимо произвести, он заполняет заявку согласно одной из прилагаемых форм.

Начальник подразделения должен сознавать, что направление подобной заявки в службу информационных технологий (СИТ) приведет к расходу рабочего времени соответствующих специалистов, а потому требуется правильно оценить необходимость проведения указанных работ, а также способность работников его подразделения адекватно использовать новые возможности, предоставляемые ИС. По этой причине при заполнении заявки начальник подразделения должен максимально подробно отразить обоснование проведения работ и другие параметры документа.

Заполнение заявки производится в форме, полученной следующим способом: с помощью текстового процессора Microsoft Word в электронной копии настоящего документа выбирается соответствующая страница приложения; заполнение необходимых полей производится в шрифте исходного документа, с соблюдением форматов исходного документа и без внесения в него дополнительных модификаций.

Если заполняющий затрудняется правильно определить параметр для заполнения, ему необходимо проконсультироваться с соответствующими подразделениями СИТ. Поля заявки нельзя оставлять незаполненными. В случае отсутствия необходимости выполнения полям действий в рамках данной заявки, необходимо поставить прочерк или написать слово "нет".

Служба информационных технологий имеет право оставлять заявки, заполненные не по форме или неполностью заполненные, без рассмотрения или требовать повторного, корректного заполнения заявки.

Право подписи заявки

Вне зависимости от типа заявки право ее подписи имеет только начальник управления или отдела.

Прохождение заявки

Подготовленные согласно настоящему документу заявки регистрируются в службе делопроизводства подразделения (выставляется отметка в соответствующем поле), затем в установленном порядке направляются в СИТ.

Начальник СИТ дает поручение соответствующим службам согласно виду заявки и вносит отметки в соответствующие поля документа.

После выполнения работ службой, стоящей последней в списке заявки, документ возвращается в подразделение делопроизводства СИТ, на заявке выставляется отметка о завершении работ, затем она подшивается в архив СИТ.

В случае когда запрос, указанный в заявке, подразумевает параллельное выполнение работ различными службами, заявка копируется в необходимом количестве экземпляров и направляется службам, указанным в списке на заявке. После выполнения указанных в заявке работ руководитель службы делает соответствующую отметку в соответствующих полях и направляет свою копию заявки в подразделение делопроизводства СИТ, на ней выставляется отметка о завершении работ, а заявка направляется в архив СИТ.

Классификация видов работ по заявкам

Новый компьютер

При получении подразделением нового компьютера необходимо определить работника или группу работников подразделения, которые будут его использовать и заполнить заявку 01¹.

¹ Нумерация данной и последующих заявок указаны в соответствии с *приложением 14*.

Перерегистрация компьютера

При увольнении, длительном отсутствии сотрудника или при принятии руководителем решения о перераспределении функциональных задач в подразделении необходимо перераспределить ответственность за персональный компьютер с одного на другого сотрудника. При этом заполняется заявка 01.

Перенос компьютера

При возникновении объективной потребности или при принятии руководителем решения о переносе персонального компьютера на другое место расположения необходимо перераспределить ответственность за персональный компьютер с одного на другого сотрудника. При этом заполняется заявка 02 (те поля, которые указаны для переноса компьютера).

Ремонт аппаратной части компьютера

При появлении поломок в аппаратной части компьютера (обрыв кабеля, перегорание монитора и т. п.) восстановление работоспособности персонального компьютера производится на основании устного обращения в соответствующее подразделение СИТ. Однако, если причины сбоя носят серьезный характер, специалист СИТ, ответственный за восстановление работоспособности ПК, может принять решение о необходимости оформления заявки на ремонт, о чем он сообщает ответственному за ПК. В этом случае руководителю подразделения необходимо заполнить заявку 02, те поля, которые указаны для ремонта компьютера.

Сбой в работе программной части компьютера

При появлении сбоев в системе ("зависание компьютера", потеря файлов и т. п.) восстановление работоспособности персонального компьютера производится на основании устного обращения в соответствующую службу СИТ. Однако, если причины сбоя носят серьезный характер, специалист СИТ, ответственный за восстановление работоспособности ПК, может принять решение о необходимости оформления заявки на ремонт, о чем он сообщает ответственному за ПК. В этом случае руководителю подразделения необходимо заполнить заявку 03.

Регистрация нового пользователя в основной информационной системе

При принятии руководителем подразделения решения о необходимости включения в список функциональных обязанностей для данного сотрудника работу с основной информационной системой (ОИС), заполняется заявка 04 (те поля, которые указаны для регистрации нового пользователя).

Постоянные изменения статуса пользователя

При изменении функциональных обязанностей данного сотрудника необходимо изменить его права в ОИС. Для этого заполняется заявка 04 (те поля, которые указаны для постоянного изменения статуса пользователя).

Временные изменения в статусе пользователя

В случае, когда необходимо временно назначить те или иные функции пользователю ОИС, заполняется заявка 04 (те поля, которые указаны для временного изменения статуса пользователя).

Работа с Интернетом

При принятии руководителем подразделения решения о необходимости включения в функциональные обязанности сотрудника работу в Интернете заполняется заявка 05.

Внешняя электронная почта

При принятии руководителем подразделения решения о необходимости включения в функциональные обязанности сотрудника переписку с партнерами, расположенными вне АБВ по электронной почте заполняется заявка 05.

Системные приложения

При необходимости установки на персональный компьютер дополнительных системных приложений или обновления версий имеющихся приложений (таких, как приложения Microsoft Office, сервисы внутренней электронной почты, электронные словари, файловые оболочки и т. п.), заполняется заявка 05.

Создание нового программного обеспечения

При принятии решения о необходимости автоматизации того или иного направления производственной деятельности необходимо оформить заявку 06 с обязательным заполнением приложений в зависимости от характера автоматизируемой задачи.

Модификация существующего программного обеспечения

При принятии решения о необходимости дополнительного изменения разработанного для подразделения прикладного программного обеспечения необходимо заполнить заявку 06 с обязательным заполнением приложений в зависимости от характера модифицируемой задачи.

Нестандартное программное обеспечение

При возникновении производственной необходимости, требующей установки дополнительного программного обеспечения, не входящего в список зарегистрированных и поддерживаемых СИТ, необходимо заполнить заявку 07.

Прочие виды работ

В случаях когда необходимо, чтобы СИТ выполнила работы, которые не попадают ни в один из указанных выше разделов, необходимо заполнить заявку 08.

Формы приложений к заявке 06

Создание или модификация отчета

В случае, когда руководитель подразделения принимает решение о необходимости создания программного обеспечения для формирования отчета или о внесении изменения в существующее программное обеспечение для модификации существующего отчета или добавления нового отчета, в качестве приложения к соответствующей заявке необходимо заполнить приложение PR-1¹.

Создание или модификация расчета

При принятии решения руководителем подразделения о необходимости создания программного обеспечения для автоматизации некоторого расчетного процесса или о внесении изменения в существующее программное обеспечение для модификации или добавления нового расчетного процесса, в качестве приложения к соответствующей заявке необходимо заполнить приложение PR-2.

Модификация справочников

Если руководитель подразделения принимает решение о внесении изменения в существующее программное обеспечение для его модификации или добавления нового справочника, в качестве приложения к заявке необходимо заполнить приложение PR-3.

Модификация логического контроля

В случае когда руководитель подразделения принимает решение о внесении изменения в существующее программное обеспечение для его модификации или добавления нового параметра, который необходимо контролировать

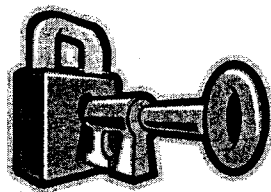
¹ Нумерация данного и последующих приложений к заявкам указаны в соответствии с *приложением 14*.

автоматически при работе данного программного обеспечения, в качестве приложения к заявке необходимо заполнить приложение PR-4.

Прочие задания по модификации программного обеспечения

Если руководитель подразделения принимает решение о необходимости создания программного обеспечения или о внесении изменения в существующее программное обеспечение, а характер задачи, предназначенной для автоматизации, не подпадает под указанные пункты, в качестве приложения к соответствующей заявке необходимо заполнить приложение PR-5.

Глава 29



Законы и нормативы

Обзор законодательных актов

В этой главе речь пойдет об общегосударственных законах и ведомственных нормативных актах, их влиянии на работу службы информационной безопасности предприятия.

Поскольку книга написана на русском языке, логично предположить, что большинство читателей — россияне. Кроме того, на пост-советском пространстве Россия является одним из лидеров в формировании правовой базы информационной безопасности. По этой причине приведенные законы и нормативы — российские.

Почему речь идет о, казалось бы, внешнем по отношению к организации нормативном пространстве? Потому что на самом деле никакие приказы и распоряжения (а значит и правила, инструкции, предписания и т. д.), которые будут выпущены внутри организации, не могут превышать общегосударственные законы либо приуменьшать общегражданские права сотрудников предприятия. То же самое относится и к договорным взаимоотношениям между данной организацией и ее партнерами.

Это означает, что при работе служба информационной безопасности должна учитывать положения Конституции, Уголовного, Процессуального, Гражданского и Трудового кодексов и других законов. Кроме того, для специалистов важно знать и законодательство и нормативы, касающиеся именно той области, в которой они работают.

Некоторые из приведенных в данной главе документов полезны тем, что создают единое терминологическое пространство [17]¹, позволяя специалистам разговаривать на одном языке, а кроме того, обеспечивая отсутствие разночтений, например, при оформлении договоров и контрактов. Другие

¹ Ссылки на документы в виде сквозной нумерации, заключенной в квадратные скобки, соотносятся только с материалами данной главы.

предлагают конкретные методики по оценке эффективности систем [14, 15], позволяя анализировать качество того или иного продукта (автоматизированного средства) по информационной безопасности. Третьи придают правовой статус механизмам информационной безопасности [4], перемещая ряд правовых моделей из обычного мира в электронный.

Если же по каким-либо причинам необходимые законы в стране отсутствуют, либо не покрывают многообразия имеющихся возможностей использования информационных систем, работающих на предприятии, специалистам по безопасности придется самостоятельно разрабатывать необходимые документы или их части. Основной принцип здесь таков, что если государство не регламентирует какой-либо из видов информационного взаимодействия между субъектами, то он должен быть регламентирован двух- (или более) сторонними договорными взаимоотношениями.

Необходимо не только не допускать неоднозначного толкования используемых понятий, но и не превышать договорными отношениями прав и обязанностей, накладываемых общегосударственным законодательством.

Знание государственного законодательства, ведомственных нормативов и всех документов, создающих правовое пространство необходимо и при разработке политики, и при собеседованиях с кандидатами на работу, и при проведении расследования.

Список полных наименований рассматриваемых законов и нормативных актов приведен в *перечне документов в конце данной главы*.

Избранные статьи и положения



Закон Российской Федерации "О государственной тайне"

Данный закон [1] интересен для организаций, работающих с информацией, которая уже отнесена или может быть отнесена в будущем к государственной тайне. Кроме того, ряд положений данного закона можно использовать для формирования нормативных документов по работе с информацией, представляющей коммерческую тайну данного предприятия.

Статья 2. Основные понятия, используемые в настоящем законе:

- ❑ *государственная тайна — защищаемые государством сведения в области его военной, внешнеполитической, экономической, разведывательной, контрразведывательной и оперативно-розыскной деятельности, распространение которых может нанести ущерб безопасности Российской Федерации;*

□ *носители сведений, составляющих государственную тайну, — материальные объекты, в том числе физические поля, в которых сведения, составляющие государственную тайну, находят свое отображение в виде символов, образов, сигналов, технических решений и процессов.*

В виду большого объема материала, который может быть полезен читателю, чтобы не перегружать книгу цитатами, укажем лишь названия статей, наиболее интересных с точки зрения организации работ, связанных с охраной государственной тайны.

Статья 5. Перечень сведений, составляющих государственную тайну.

Статья 7. Сведения, не подлежащие отнесению к государственной тайне и засекречиванию.

Статья 8. Степени секретности сведений и грифы секретности носителей этих сведений.

Статья 9. Порядок отнесения сведений к государственной тайне.

Статья 10. Ограничение прав собственности предприятий, учреждений, организаций и граждан Российской Федерации на информацию в связи с ее засекречиванием.

Статья 11. Порядок засекречивания сведений и их носителей.

Статья 16. Взаимная передача сведений, составляющих государственную тайну, органами государственной власти, предприятиями, учреждениями и организациями.

Статья 17. Передача сведений, составляющих государственную тайну, в связи с выполнением совместных и других работ.

Статья 22. Основания для отказа должностному лицу или гражданину в допуске к государственной тайне.

Основаниями для отказа должностному лицу или гражданину в допуске к государственной тайне могут являться:

- *признание его судом недееспособным, ограниченно дееспособным или рецидивистом, нахождение его под судом или следствием за государственные и иные тяжкие преступления, наличие у него неснятой судимости за эти преступления;*
- *наличие у него медицинских противопоказаний для работы с использованием сведений, составляющих государственную тайну, согласно перечню, утверждаемому Министерством здравоохранения Российской Федерации;*
- *постоянное проживание его самого и (или) его близких родственников за границей и (или) оформление указанными лицами документов для выезда на постоянное жительство в другие государства;*
- *выявление в результате проверочных мероприятий действий оформляемого лица, создающих угрозу безопасности Российской Федерации;*

- ☐ уклонение его от проверочных мероприятий и (или) сообщение им заведомо ложных анкетных данных.

Последняя статья приведена более развернуто, так как может быть использована в формировании требований по информационной безопасности при приеме на работу нового сотрудника.

Федеральный закон Российской Федерации "Об информации, информатизации и защите информации"

Данный Закон [2] является базовым документом для любой службы информационной безопасности российского предприятия. Достаточно взглянуть на первую статью.

Статья 1. Сфера действия настоящего Федерального закона.

Настоящий Федеральный закон регулирует отношения, возникающие при:

- ☐ *формировании и использовании информационных ресурсов на основе создания, сбора, обработки, накопления, хранения, поиска, распространения и предоставления потребителю документированной информации;*
- ☐ *создании и использовании информационных технологий и средств их обеспечения;*
- ☐ *защите информации, прав субъектов, участвующих в информационных процессах и информатизации.*

В Законе указаны важные параметры функционирования информационного пространства, в частности условия приобретения информацией того или иного статуса.

Статья 5. Документирование информации.

Документирование информации является обязательным условием включения информации в информационные ресурсы. Документирование информации осуществляется в порядке, устанавливаемом органами государственной власти, ответственными за организацию делопроизводства, стандартизацию документов и их массивов, безопасность Российской Федерации.

Документ, полученный из автоматизированной информационной системы, приобретает юридическую силу после его подписания должностным лицом в порядке, установленном законодательством Российской Федерации.

Юридическая сила документа, хранимого, обрабатываемого и передаваемого с помощью автоматизированных информационных и телекоммуникационных систем, может подтверждаться электронной цифровой подписью.

Юридическая сила электронной цифровой подписи признается при наличии в автоматизированной информационной системе программно-технических средств, обеспечивающих идентификацию подписи, и соблюдении установленного режима их использования.

Право удостоверять идентичность электронной цифровой подписи осуществляется на основании лицензии. Порядок выдачи лицензий определяется законодательством Российской Федерации.

Важным моментом для работы службы безопасности организации является работа с различными данными о сотрудниках организации. Следующий раздел закона накладывает ограничения и определяет условия, невыполнение которых может быть критическим для успешной работы.

Статья 11. Информация о гражданах (персональные данные).

Перечни персональных данных, включаемых в состав федеральных информационных ресурсов, информационных ресурсов совместного ведения, информационных ресурсов субъектов Российской Федерации, информационных ресурсов органов местного самоуправления, а также получаемых и собираемых негосударственными организациями, должны быть закреплены на уровне федерального закона. Персональные данные относятся к категории конфиденциальной информации.

Не допускаются сбор, хранение, использование и распространение информации о частной жизни, а равно информации, нарушающей личную тайну, семейную тайну, тайну переписки, телефонных переговоров, почтовых, телеграфных и иных сообщений физического лица без его согласия, кроме как на основании судебного решения.

Статья 14. Доступ граждан и организаций к информации о них.

Граждане и организации имеют право на доступ к документированной информации о них, на уточнение этой информации в целях обеспечения ее полноты и достоверности, имеют право знать, кто и в каких целях использует или использовал эту информацию. Ограничение доступа граждан и организаций к информации о них допустимо лишь на основаниях, предусмотренных федеральными законами.

Статья 21 определяет собственно необходимость существования службы безопасности, как таковой.

Статья 21. Защита информации.

Защите подлежит любая документированная информация, неправомерное обращение с которой может нанести ущерб ее собственнику, владельцу, пользователю и иному лицу.

Закон Российской Федерации "О правовой охране программ для электронных вычислительных машин и баз данных"

Данный Закон [3], с одной стороны, применим в информационной безопасности с точки зрения обеспечения защиты интеллектуальной собственности предприятия, а с другой стороны, создает основу для защиты от обвинений в использовании пиратских программ.

Статья 2. Отношения, регулируемые настоящим законом.

Программы для ЭВМ и базы данных относятся настоящим Законом к объектам авторского права. Программам для ЭВМ предоставляется правовая охрана как произведениям литературы, а базам данных — как сборникам.

Статья 3. Объект правовой охраны.

Предоставляемая настоящим Законом правовая охрана не распространяется на идеи и принципы, лежащие в основе программы для ЭВМ или базы данных или какого-либо их элемента, в том числе на идеи и принципы организации интерфейса и алгоритма, а также языки программирования.

Авторское право на программы для ЭВМ и базы данных не связано с правом собственности на их материальный носитель. Любая передача прав на материальный носитель не влечет за собой передачи каких-либо прав на программы для ЭВМ и базы данных.

Статья 14. Использование программы для ЭВМ или базы данных по договору с правообладателем.

Использование программы для ЭВМ или базы данных третьими лицами (пользователями) осуществляется на основании договора с правообладателем, за исключением случаев, указанных в статье 16 настоящего Закона.

Договор на использование программы для ЭВМ или базы данных заключается в письменной форме.

При продаже и предоставлении массовым пользователям доступа к программам для ЭВМ и базам данных допускается применение особого порядка заключения договоров, например, путем изложения типовых условий договора на передаваемых экземплярах программ для ЭВМ и баз данных.

Статья 15. Свободное воспроизведение и адаптация программы для ЭВМ или базы данных.

Лицо, правомерно владеющее экземпляром программы для ЭВМ, вправе без согласия правообладателя и без выплаты дополнительного вознаграждения декомпи-

лировать или поручать декомпилирование программы для ЭВМ с тем, чтобы изучать кодирование и структуру этой программы при следующих условиях:

- ❑ информация, необходимая для взаимодействия независимо разработанной данным лицом программы для ЭВМ с другими программами, не доступна из других источников;
- ❑ информация, полученная в результате этого декомпилирования, может использоваться лишь для организации взаимодействия независимо разработанной данным лицом программы для ЭВМ с другими программами, а не для составления новой программы для ЭВМ, по своему виду существенно схожей с декомпилируемой программой для ЭВМ или для осуществления любого другого действия, нарушающего авторское право;
- ❑ декомпилирование осуществляется в отношении только тех частей программы для ЭВМ, которые необходимы для организации такого взаимодействия.

Федеральный закон Российской Федерации "Об электронной цифровой подписи"

Для предприятий, использующих этот криптографический механизм, важен практически весь Закон [4], поэтому выдержки из него здесь не приведены. Специалистам понятно значение этого закона, который придает электронным документам юридическую значимость документам на бумаге, с соответствующими подписями и печатями.

Положение по аттестации объектов информатизации по требованиям безопасности информации

Данное Положение [9] интересно тем, что в нем представлены схемы, которые можно использовать специалистам по безопасности предприятия. Например, раздел 3.7.1 данного Положения полезен для составления плана по внедрению новой информационной системы, поэтому представим его более подробно.

На этапе аттестационных испытаний объекта информатизации:

- ❑ осуществляется анализ организационной структуры объекта информатизации, информационных потоков, состава и структуры комплекса технических средств и программного обеспечения, системы защиты информации на объекте, разработанной документации и ее соответствия требованиям нормативной документации по защите информации;

- *определяется правильность категорирования объектов ЭВТ и классификации АС (при аттестации автоматизированных систем), выбора и применения сертифицированных и несертифицированных средств и систем защиты информации;*
- *проводятся испытания несертифицированных средств и систем защиты информации на аттестуемом объекте или анализ результатов их испытаний в испытательных центрах (лабораториях) по сертификации;*
- *проверяется уровень подготовки кадров и распределение ответственности персонала за обеспечение выполнения требований по безопасности информации;*
- *проводятся комплексные аттестационные испытания объекта информатизации в реальных условиях эксплуатации путем проверки фактического выполнения установленных требований на различных этапах технологического процесса обработки защищаемой информации;*
- *оформляются протоколы испытаний и заключение по результатам аттестации с конкретными рекомендациями по устранению допущенных нарушений, приведению системы защиты объекта информатизации в соответствие с установленными требованиями и совершенствованию этой системы, а также рекомендациями по контролю за функционированием объекта информатизации.*

Приложение к форме заявки документа может быть использовано для организации работ по инвентаризации информационных систем.

Исходные данные по аттестуемому объекту информатизации готовятся на основе следующего перечня вопросов.

- 1. Полное и точное наименование объекта информатизации и его назначение.*
- 2. Характер (научно-техническая, экономическая, производственная, финансовая, военная, политическая) и уровень секретности (конфиденциальности) обрабатываемой информации определены в соответствии с перечнями — государственным, отраслевым, ведомственным — предприятия.*
- 3. Организационная структура объекта информатизации.*
- 4. Перечень помещений, состав комплекса технических средств (основных и вспомогательных), входящих в объект информатизации, в которых (на которых) обрабатывается указанная информация (расположенных в помещениях, где она циркулирует).*
- 5. Особенности и схема расположения объекта информатизации с указанием границ контролируемой зоны.*

6. Структура программного обеспечения (общесистемного и прикладного), используемого на аттестуемом объекте информатизации и предназначенного для обработки защищаемой информации, используемые протоколы обмена информацией.
7. Общая функциональная схема объекта информатизации, включая схему информационных потоков и режимы обработки защищаемой информации.
8. Наличие и характер взаимодействия с другими объектами информатизации.
9. Состав и структура системы защиты информации на аттестуемом объекте информатизации.
10. Перечень технических и программных средств в защищенном исполнении, средств защиты и контроля, используемых на аттестуемом объекте информатизации и имеющих соответствующий сертификат, предписание на эксплуатацию.
11. Сведения о разработчиках системы защиты информации, наличие у сторонних разработчиков (по отношению к предприятию, на котором расположен аттестуемый объект информатизации) лицензий на проведение подобных работ.
12. Наличие на объекте информатизации (на предприятии, на котором расположен объект информатизации) службы безопасности информации, службы администратора (автоматизированной системы, сети, баз данных).
13. Наличие и основные характеристики физической защиты объекта информатизации (помещений, где обрабатывается защищаемая информация и хранятся информационные носители).
14. Наличие и готовность проектной и эксплуатационной документации на объект информатизации и другие исходные данные по аттестуемому объекту информатизации, влияющие на безопасность информации.

Положение о сертификации средств защиты информации по требованиям безопасности информации

Данное Положение [10] интересно возможностью признания иностранных сертификатов, а также формой самого сертификата.

Для признания зарубежного сертификата заявитель направляет его копию и заявку на признание сертификата в Гостехкомиссию России, которая извещает заявителя о признании или необходимости проведения сертификационных испытаний не позднее двух месяцев после их получения. В случае признания — заявителю выдается сертификат установленного образца.

Государственная техническая комиссия
при Президенте Российской Федерации
Система сертификации средств защиты информации
по требованиям безопасности информации
№ РОСС RU. 0001. 01БИОО

СЕРТИФИКАТ Выдан “___” _____ 200_ г.

№ _____ Действителен до “___” _____ 200_ г.

Настоящий сертификат удостоверяет, что:

(наименование вида продукции, код, № ТУ)

соответствует требованиям _____

(перечисление конкретных стандартов или нормативных документов,
на соответствие которым проведены сертификационные испытания)

Сертификат выдан на основании экспертного заключения

(наименование органа по сертификации)

и результатов испытаний указанной продукции

(наименование испытательного центра (лаборатории))

Заявитель

(наименование организации-заявителя, адрес)

Место

гербовой

печати

(подпись)

(дата)

Положение о государственном лицензировании деятельности в области защиты информации

Положение [11] интересно рядом определений, которые возможно использовать при создании собственных нормативных документов предприятия, а также при официальной переписке с другими организациями. Обратим внимание на следующие определения.

Защита информации — комплекс мероприятий, проводимых с целью предотвращения утечки, хищения, утраты, несанкционированного уничтожения, искажения, модификации (подделки), несанкционированного копирования, блокирования информации и т. п.

Эффективность защиты информации — степень соответствия достигнутых результатов действий по защите информации поставленной цели защиты.

Безопасность информации — состояние информации, информационных ресурсов и информационных систем, при котором с требуемой вероятностью обеспечивается защита информации (данных) от утечки, хищения, утраты, несанкционированного уничтожения, искажения, модификации (подделки), несанкционированного копирования, блокирования и т. п.

Особенно интересно определение безопасности информации, которое можно сравнить с определением информационной безопасности, данным в *главе 1* этой книги. Для обнаруживших различия поясним, что для авторов ряд понятий (таких, скажем, как "утечка и утрата", "искажение и модификация") являются в смысле информационной безопасности синонимичными.

Сам документ полезен тем, что дает инструмент оценки потенциального поставщика услуг по информационной безопасности. Например, пункт 1.5 данного Положения [11] указывает максимальный срок действия лицензии такого поставщика (три года). А пункт 1.6 Положения [11] показывает, какими ресурсами он должен обладать.

Лицензия выдается подавшему заявку на ее получение предприятию-заявителю, располагающему производственной и испытательной базой, нормативной и методической документацией, научным и инженерно-техническим персоналом, при условии их соответствия требованиям государственного органа по лицензированию на основании результатов экспертизы деятельности предприятия по заявленному направлению работ.

Кроме того, Положение в разделах 2.4.2. и 2.5. указывает возможность защиты предприятия от некачественной работы поставщика.

Лицензиаты несут юридическую и финансовую ответственность за полноту и качество выполнения работ и обеспечение сохранности государственных и коммерческих секретов, доверенных им в ходе практической деятельности.

Предприятия-потребители имеют право обращаться:

- ☐ *в государственный орган по лицензированию или в лицензионный центр с рекламациями на некачественно выполненные лицензиатами работы по защите информации;*
- ☐ *в судебные органы в установленном порядке.*

Также указано, для каких сфер информационной безопасности поставщику требуется лицензия (список сокращен).

- ☐ *Сертификация, сертификационные испытания технических средств защиты информации.*

- ☐ *Аттестация средств приема, передачи и обработки подлежащей защите информации.*
- ☐ *Разработка технических средств защиты информации.*
- ☐ *Проведение специсследований на побочные электромагнитные излучения и наводки.*
- ☐ *Проектирование объектов в защищенном исполнении.*
- ☐ *Подготовка кадров в области защиты информации по видам деятельности, перечисленным в данном перечне.*

Классификация автоматизированных систем и требования по защите информации

Этот документ [12] описывает схемы, которые могут быть использованы предприятием при проведении классификации собственных автоматизированных систем (АС).

Основными этапами классификации АС являются:

- ☐ *разработка и анализ исходных данных;*
- ☐ *выявление основных признаков АС, необходимых для классификации;*
- ☐ *сравнение выявленных признаков АС с классифицируемыми;*
- ☐ *присвоение АС соответствующего класса защиты информации от НСД.*

Необходимыми исходными данными для проведения классификации конкретной АС являются:

- ☐ *перечень защищаемых информационных ресурсов АС и их уровень конфиденциальности;*
- ☐ *перечень лиц, имеющих доступ к штатным средствам АС, с указанием их уровня полномочий;*
- ☐ *матрица доступа или полномочий субъектов доступа по отношению к защищаемым информационным ресурсам АС;*
- ☐ *режим обработки данных в АС.*

К числу определяющих признаков, по которым производится группировка АС в различные классы, относятся:

- ☐ *наличие в АС информации различного уровня конфиденциальности;*
- ☐ *уровень полномочий субъектов доступа АС на доступ к конфиденциальной информации;*
- ☐ *режим обработки данных в АС — коллективный или индивидуальный.*

Концепция защиты средств вычислительной техники и автоматизированных систем от несанкционированного доступа к информации

Концепция [13] интересна как основа, на которой можно строить собственные нормативные документы. Например, модель злоумышленника, представленную в *разделе 4* данной Концепции, можно целиком перенести в Стандарты безопасности предприятия.

МОДЕЛЬ НАРУШИТЕЛЯ В АС

Классификация является иерархической, т.е. каждый следующий уровень включает в себя функциональные возможности предыдущего.

Первый уровень определяет самый низкий уровень возможностей ведения диалога в АС — запуск задач (программ) из фиксированного набора, реализующих заранее предусмотренные функции по обработке информации.

Второй уровень определяется возможностью создания и запуска собственных программ с новыми функциями по обработке информации.

Третий уровень определяется возможностью управления функционированием АС, т.е. воздействием на базовое программное обеспечение системы и на состав и конфигурацию ее оборудования.

Четвертый уровень определяется всем объемом возможностей лиц, осуществляющих проектирование, реализацию и ремонт технических средств АС, вплоть до включения в состав средств вычислительной техники (СВТ) собственных технических средств с новыми функциями по обработке информации.

В своем уровне нарушитель является специалистом высшей квалификации, знает все об АС и, в частности, о системе и средствах ее защиты.

Описание способов организации несанкционированного доступа (НСД) к объектам можно использовать при анализе защищенности или написания политики безопасности для конкретной информационной системы.

ОСНОВНЫЕ СПОСОБЫ НСД

К основным способам НСД относятся:

- ☐ *непосредственное обращение к объектам доступа;*
- ☐ *создание программных и технических средств, выполняющих обращение к объектам доступа в обход средств защиты;*
- ☐ *модификация средств защиты, позволяющая осуществить НСД;*
- ☐ *внедрение в технические средства СВТ или АС программных или технических механизмов, нарушающих предполагаемую структуру и функции СВТ или АС и позволяющих осуществить НСД.*

Раздел 6 Концепции [13] могут использовать разработчики программного обеспечения или технологи при проектировании механизмов безопасности информационных систем, поэтому приведем основные его положения.

ОСНОВНЫЕ НАПРАВЛЕНИЯ ОБЕСПЕЧЕНИЯ ЗАЩИТЫ ОТ НСД

Обеспечение защиты СВТ и АС осуществляется:

- ☐ системой разграничения доступа (СРД) субъектов к объектам доступа;
- ☐ обеспечивающими средствами для СРД.

Основными функциями СРД являются:

- ☐ реализация правил разграничения доступа (ПРД) субъектов и их процессов к данным;
- ☐ реализация ПРД субъектов и их процессов к устройствам создания твердых копий;
- ☐ изоляция программ процесса, выполняемого в интересах субъекта, от других субъектов;
- ☐ управление потоками данных в целях предотвращения записи данных на носители несоответствующего грифа;
- ☐ реализация правил обмена данными между субъектами для АС и СВТ, построенных по сетевым принципам.

Обеспечивающие средства для СРД выполняют следующие функции:

- ☐ идентификацию и опознание (аутентификацию) субъектов и поддержание привязки субъекта к процессу, выполняемому для субъекта;
- ☐ регистрацию действий субъекта и его процесса;
- ☐ предоставление возможностей исключения и включения новых субъектов и объектов доступа, а также изменение полномочий субъектов;
- ☐ реакцию на попытки НСД, например, сигнализацию, блокировку, восстановление после НСД;
- ☐ тестирование;
- ☐ очистку оперативной памяти и рабочих областей на магнитных носителях после завершения работы пользователя с защищаемыми данными;
- ☐ учет выходных печатных и графических форм и твердых копий в АС;
- ☐ контроль целостности программной и информационной части как СРД, так и обеспечивающих ее средств.

Межсетевые экраны. Показатели защищенности от несанкционированного доступа к информации

Данный руководящий документ [14] позволяет произвести оценку межсетевого экрана (МСЭ), предназначенного для разработки или приобретения предприятием. Кроме того, документ позволяет выбрать желаемый уровень

защищенности при проектировании системы в зависимости от классификации защищаемой информации.

Термины и определения (в алфавитном порядке)

Администратор МСЭ — лицо, ответственное за сопровождение МСЭ.

Критерии фильтрации — параметры, атрибуты, характеристики, на основе которых осуществляется разрешение или запрещение дальнейшей передачи пакета (данных) в соответствии с заданными правилами разграничения доступа (правилами фильтрации).

Межсетевой экран (МСЭ) — это локальное (однокомпонентное) или функционально-распределенное программное (программно-аппаратное) средство (комплекс), реализующее контроль за информацией, поступающей в АС и/или выходящей из АС.

Правила фильтрации — перечень условий, по которым с использованием заданных критериев фильтрации осуществляется разрешение или запрещение дальнейшей передачи пакетов (данных) и перечень действий, производимых МСЭ по регистрации и/или осуществлению дополнительных защитных функций.

Сетевые адреса — адресные данные, идентифицирующие субъекты и объекты и используемые протоколом сетевого уровня модели Международной организации по стандартизации взаимодействия открытых систем (OSI).

Трансляция адреса — функция МСЭ, скрывающая внутренние адреса объектов (субъектов) от внешних субъектов.

Транспортные адреса — адресные данные, идентифицирующие субъекты и объекты и используемые протоколом транспортного уровня модели OSI.

Экранирование — функция МСЭ, позволяющая поддерживать безопасность объектов внутренней области, игнорируя несанкционированные запросы из внешней области.

Документ накладывает следующие требования на МСЭ, используемые в механизмах защиты информации (классификация АС приводится по документу [12], классификация МСЭ производится ранее в самом документе [14]).

При включении МСЭ в АС определенного класса защищенности, класс защищенности совокупной АС, полученной из исходной путем добавления в нее МСЭ, не должен понижаться.

Для АС класса 3Б, 2Б должны применяться МСЭ не ниже 5 класса.

Кроме того, можно воспользоваться терминами, используемыми в документе для включения в собственную нормативную документацию, например в ту же политику для межсетевого экрана (список сокращен).

Показатели защищенности от несанкционированного доступа к информации

Данный документ [15] — российский аналог следующих широко известных в области информационной безопасности документов: "Оранжевой" [DOD1] и "Красной" [NCSC] книг, Общих критериев [CC] (их краткий обзор был представлен в главе 6).

Устанавливается семь классов защищенности СВТ от НСД к информации. Самый низкий класс — седьмой, самый высокий — первый.

Классы подразделяются на четыре группы, отличающиеся качественным уровнем защиты:

- ☐ *первая группа содержит только один седьмой класс;*
- ☐ *вторая группа характеризуется дискреционной защитой и содержит шестой и пятый классы;*
- ☐ *третья группа характеризуется мандатной защитой и содержит четвертый, третий и второй классы;*
- ☐ *четвертая группа характеризуется верифицированной защитой и содержит только первый класс.*

Помимо целей, которые были рассмотрены в главе 6, документ [15] может быть использован разработчиками программного обеспечения или технологами при проектировании комплексов средств защиты (КСЗ) — как справочник по следующим подсистемам безопасности конкретной информационной системы:

- ☐ дискреционный принцип контроля доступа;
- ☐ мандатный принцип контроля доступа;
- ☐ очистка памяти;
- ☐ изоляция модулей;
- ☐ маркировка документов;
- ☐ защита ввода и вывода на отчуждаемый физический носитель информации;
- ☐ сопоставление пользователя с устройством;
- ☐ идентификация и аутентификация;
- ☐ гарантии проектирования;
- ☐ регистрация;
- ☐ взаимодействие пользователя с КСЗ;
- ☐ надежное восстановление;
- ☐ целостность КСЗ;

- ☐ контроль модификации;
- ☐ контроль дистрибуции;
- ☐ гарантии архитектуры;
- ☐ тестирование;
- ☐ руководство для пользователя;
- ☐ руководство по КСЗ;
- ☐ тестовая документация;
- ☐ конструкторская (проектная) документация.

Документ "Защита от несанкционированного доступа к информации. Термины и определения"

Данный документ [17] интересен возможностью использования в работе службы информационной безопасности устоявшейся терминологии. Ряд определений из документа возможно напрямую перенести в нормативы предприятия, другие — модифицировать. Представим сокращенный список определений.

Доступ к информации — ознакомление с информацией, ее обработка, в частности, копирование, модификация или уничтожение информации.

Несанкционированный доступ к информации (НСД) — доступ к информации, нарушающий правила разграничения доступа с использованием штатных средств, предоставляемых средствами вычислительной техники или автоматизированными системами.

Субъект доступа — лицо или процесс, действия которого регламентируются правилами разграничения доступа.

Объект доступа — единица информационного ресурса автоматизированной системы, доступ к которой регламентируется правилами разграничения доступа.

Матрица доступа — таблица, отображающая правила разграничения доступа.

Идентификатор доступа — уникальный признак субъекта или объекта доступа.

Идентификация — присвоение субъектам и объектам доступа идентификатора и (или) сравнение предъявляемого идентификатора с перечнем присвоенных идентификаторов.

Пароль — идентификатор субъекта доступа, который является его (субъекта) секретом.

Аутентификация — проверка принадлежности субъекту доступа предъявленного им идентификатора; подтверждение подлинности.

Модель защиты — абстрактное (формализованное или неформализованное) описание комплекса программно-технических средств и (или) организационных мер защиты от несанкционированного доступа.

Безопасность информации — состояние защищенности информации, обрабатываемой средствами вычислительной техники или автоматизированной системы, от внутренних или внешних угроз.

Дискреционное управление доступом — разграничение доступа между поименованными субъектами и поименованными объектами. Субъект с определенным правом доступа может передать это право любому другому субъекту.

Мандатное управление доступом — разграничение доступа субъектов к объектам, основанное на характеризующей меткой конфиденциальности информации, содержащейся в объектах, и официальном разрешении (допуске) субъектов обращаться к информации такого уровня конфиденциальности.

Администратор защиты — субъект доступа, ответственный за защиту автоматизированной системы от несанкционированного доступа к информации.

Верификация — процесс сравнения двух уровней спецификации средств вычислительной техники или автоматизированных систем на надлежащее соответствие.

Система защиты информации от несанкционированного доступа — комплекс организационных мер и программно-технических (в том числе криптографических) средств защиты от несанкционированного доступа к информации в автоматизированных системах.

Перечень документов

[1]. О ГОСУДАРСТВЕННОЙ ТАЙНЕ. Закон Российской Федерации от 21 июля 1993 года N 5485-1 (в ред. Федерального закона от 6 октября 1997 года N131-ФЗ).

[2]. ОБ ИНФОРМАЦИИ, ИНФОРМАТИЗАЦИИ И ЗАЩИТЕ ИНФОРМАЦИИ. Федеральный закон Российской Федерации от 20 февраля 1995 года N 24-ФЗ. Принят Государственной Думой 25 января 1995 года.

[3]. О ПРАВОВОЙ ОХРАНЕ ПРОГРАММ ДЛЯ ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН И БАЗ ДАННЫХ. Закон Российской Федерации от 23 сентября 1992 года N 3524-1.

[4]. ОБ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ. Федеральный закон Российской Федерации от 10 января 2002 года N 1-ФЗ.

[5]. ОБ АВТОРСКОМ ПРАВЕ И СМЕЖНЫХ ПРАВАХ. Закон Российской Федерации от 9 июля 1993 года N 5351-1.

[6]. О ФЕДЕРАЛЬНЫХ ОРГАНАХ ПРАВИТЕЛЬСТВЕННОЙ СВЯЗИ И ИНФОРМАЦИИ. Закон Российской Федерации (в ред. Указа Президента РФ от 24.12.1993 N 2288; Федерального закона от 07.11.2000 N 135-ФЗ).

[7]. Положение об аккредитации испытательных лабораторий и органов по сертификации средств защиты информации по требованиям безопасности информации / Государственная техническая комиссия при Президенте Российской Федерации.

[8]. Инструкция о порядке маркирования сертификатов соответствия, их копий и сертифицированных средств защиты информации / Государственная техническая комиссия при Президенте Российской Федерации.

[9]. Положение по аттестации объектов информатизации по требованиям безопасности информации / Государственная техническая комиссия при Президенте Российской Федерации.

[10]. Положение о сертификации средств защиты информации по требованиям безопасности информации: с дополнениями в соответствии с Постановлением Правительства Российской Федерации от 26 июня 1995 года № 608 "О сертификации средств защиты информации" / Государственная техническая комиссия при Президенте Российской Федерации.

[11]. Положение о государственном лицензировании деятельности в области защиты информации / Государственная техническая комиссия при Президенте Российской Федерации.

[12]. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации: Руководящий документ / Государственная техническая комиссия при Президенте Российской Федерации.

[13]. Концепция защиты средств вычислительной техники и автоматизированных систем от несанкционированного доступа к информации: Руководящий документ / Государственная техническая комиссия при Президенте Российской Федерации.

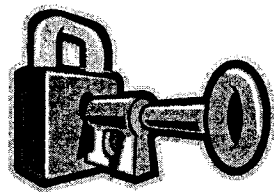
[14]. Средства вычислительной техники. Межсетевые экраны. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации: Руководящий документ / Государственная техническая комиссия при Президенте Российской Федерации.

[15]. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации: Руководящий документ / Государственная техническая комиссия при Президенте Российской Федерации.

[16]. Защита информации. Специальные защитные знаки. Классификация и общие требования: Руководящий документ / Государственная техническая комиссия при Президенте Российской Федерации.

[17]. Защита от несанкционированного доступа к информации. Термины и определения: Руководящий документ / Государственная техническая комиссия при Президенте Российской Федерации.

Глава 30



Тестирование процедур и механизмов безопасности

Как единовременное, так и периодическое тестирование всех внедренных механизмов безопасности является очень важной обязанностью специалистов службы информационной безопасности. В одних случаях тестирование может производиться автоматическими средствами, в других — при непосредственном участии специалиста службы ИБ. Кроме того, существует целый подкласс задач тестирования, который адекватно выполнить могут только внешние по отношению к предприятию специалисты — аудиторы.

Общие вопросы

Необходимость тестирования



Известно, что в конце каждого этапа жизненного цикла большинства продуктов информационных технологий (планирование, разработка, внедрение, тестирование, планирование изменений), поворотным шагом к началу нового этапа изменений продукта является тестирование. Причем этот термин можно в данном случае понимать достаточно широко — реальное использование продукта также может считаться определенным образом тестированием.

Создаваемые нами продукты по информационной безопасности (механизмы и процедуры) без сомнения являются продуктами информационных технологий, а значит, и для них тестирование является существенной частью жизненного цикла.

Классификация



Само по себе тестирование также разделяется на несколько видов (этапов).

- ☐ *Теоретическое тестирование.* Оно может начаться уже на этапе планирования или разработки продукта, когда разработчик предполагает, как будет проходить тот или иной процесс, связанный с функционированием продукта. Этап теоретического тестирования не прекращается и далее

на этапе внедрения продукта, по той простой причине, что провести практическое тестирование всех режимов работы продукта не всегда возможно, либо возможно в ограниченном режиме. Например, если на этапе внедрения продукта по безопасности или после внедрения появилась информация о новом виде атаки, следует сначала теоретически проанализировать особенности функционирования продукта в условиях новой атаки, возможно с последующей его доработкой. Кроме того, ряд атак достаточно трудно или накладно смоделировать практически, в частности, распределенную атаку на отказ в обслуживании (англ. *Distributed Deny of Service — DDoS*) производят со множества различных хостов (сетевых адресов), поэтому моделировать такие атаки также придется теоретически.

- ❑ *Практическое тестирование.* Элементы этого тестирования могут появляться на этапе разработки, когда может выясниться, что те или иные способы функционирования, допустимые теоретически, не могут быть реализованы по ряду причин. Также элементы практического тестирования могут использоваться разработчиком для первичного тестирования работы того или другого элемента продукта.
- ❑ *Полномасштабное тестирование.* Наиболее широко этот этап разворачивается обычно между разработкой и внедрением, либо между первым и масштабным внедрением. В это время группа пользователей, знакомых с функциональностью продукта, возможно, при участии самого разработчика, производит тестирование продукта в режиме, близком к реальному. Здесь необходимо учесть следующие моменты. Во-первых, очень желательно провести тестирование при участии сторонних пользователей, которые, как и будущие реальные пользователи, знакомы с продуктом только по документации, т. е. будут тестировать продукт по тому, как он будет работать, а не по тому, как думает разработчик о работе продукта. Во-вторых, необходимо хотя бы однократно провести т. н. стресс-тест, т. е. работу продукта в режиме, приближенном к максимальной загрузке или даже превышая ее. Порой увеличение количественного параметра приводит к существенным качественным изменениям в работе. Простой пример — ведение регистрационного журнала некоего механизма безопасности. Требование слишком подробного формата регистрации события при возникновении большого количества событий может привести к тому, что ряд событий будет просто пропущен.
- ❑ *Жизненная корректировка.* После внедрения продукта могут выявиться детали, порой весьма существенные, которые работают не так, как это было запланировано, причем это может быть не выявлено даже на этапе тестирования. Кроме того, изменения внешних обстоятельств (внедрение других продуктов или информационных систем, изменения нормативов и т. д.) может повлечь за собой необходимость изменений.

Дружеский взлом



Рассматривая тестирование именно продуктов по информационной безопасности, необходимо упомянуть такой специфичный метод, как *дружеский взлом*. Под этим термином понимается целенаправленное нарушение схемы работы механизма или процедуры безопасности виртуальным злоумышленником, роль которого может выполнять один из сотрудников службы безопасности предприятия или сторонний привлеченный эксперт. Естественно, что результаты такого "взлома" будут не столь плачевными, как могло бы быть при проникновении реального злоумышленника.

Поскольку такой вид тестирования имеет ряд специфических особенностей, на нем следует остановиться подробнее. Во-первых, в его результате, особенно если продукт еще достаточно сырой, псевдо-злоумышленник может, при успешной реализации взлома, получить доступ к данным или привилегиям, использование которых может нанести ущерб предприятию. Это означает, что вне зависимости от доверия лицу или группе лиц, производящих такое тестирование, оно должно быть оформлено как отдельный проект, т. е. достаточно структурировано, с отчетом по результатам выполнения каждого этапа, с согласованием начала следующего этапа, с контролем при выполнении очередного этапа.

Во-вторых, необходимо определиться с тем, кто конкретно будет реализовывать такое тестирование — собственные специалисты или привлеченные эксперты. Оба подхода имеют свои плюсы и минусы:

- ❑ Свои работники, без сомнения, лучше управляемы и контролируемы, но они могут нарушить чистоту эксперимента своими знаниями о внутреннем распорядке, способах работы и взаимосвязи систем и т. д. Кроме того, они обычно находятся под влиянием некоторого шаблона работы и могут не увидеть, скажем, уязвимости, которая очевидна для постороннего человека. Если работники привыкли каждый день проходить через охраняемую проходную, им может и не прийти в голову мысль, что с противоположной стороны забора уже давно есть большая удобная дыра.
- ❑ Привлеченные эксперты гораздо лучше подойдут на роль полностью посторонних злоумышленников. Однако это чужие люди и следует очень внимательно подготовить контракт на проект такого тестирования с тщательным распределением ответственности за последующее использование информации, полученной в результате такого теста. Лучше всего обратиться к компаниям, которые достаточно долго находятся в подобном бизнесе и не станут рисковать своим авторитетом за сомнительную выгоду использования полученных данных клиента. Но в этом случае может потребоваться и адекватная оплата труда привлеченных экспертов.

Тестирование аварийного плана



В обоих случаях потребуется соответствующая регламентация процедур тестирования. Пример "дружеского взлома" это наиболее яркий пример необходимости регламентации. Другим вариантом такой необходимости будет тестирование аварийного плана. О самом аварийном плане речь пойдет в *части VIII*, а в данном разделе рассмотрим его как одну из процедур безопасности, имеющую ярко выраженные особенности. Они заключаются в том, что ряд процедур аварийного плана затрагивает критические, жизненно важные информационные системы, а также в том, что некоторые процедуры аварийного плана требуют массового участия работников организации.

Например, можно многократно проверить процедуру восстановления систем и данных из резервных копий на тестовом оборудовании, но необходимо минимум однократно для данной версии плана или регулярно, скажем, раз в год, производить тестовое восстановление копий на рабочей системе, поскольку тестирование в реальной рабочей среде может открыть дополнительные особенности их функционирования. Очевидно, что для этого необходимо выбрать время, когда тестирование нанесет наименьший вред работе с клиентами организации, привлечь ряд сотрудников предприятия на дополнительную работу, получить согласие руководства и т. д. Другой пример — тестирование системы, связанной с аварийным выходом из здания. Необходимо участие всех работников предприятия, внесение помех в рабочий цикл функционирования предприятия, согласие руководства и множество менее существенных деталей, таких как охрана рабочих мест и имущества сотрудников на период их эвакуации.

Обратная связь от пользователей



Если говорить о тестировании процедур, то помимо собственно функциональной значимости, должно быть проверено также удобство использования этих процедур в жизни. Как бы не были надежны процедуры безопасности с точки зрения защиты корпоративных ресурсов, но, если они создают чрезмерные помехи в работе рядовым пользователям, они будут нарушаться. Задача тестирования выявить такие несоответствия.

Рассмотрим стандартный пример, возникающий на любом предприятии. Для входа в новую информационную систему пользователю потребуется отдельный пароль. Для специалиста, профессионально занимающегося информационной безопасностью, не составит труда запомнить еще один пароль к тем 2—3—4 качественным паролям, которые он уже использует. При обычном тестировании, когда тестирующий пользователь оторван от остальных систем, это также не составит проблем — использовать качественный пароль. Однако, для рядового пользователя, который и так, возможно, считает требования безопасности излишними, запоминание нового пароля

(да еще и требование менять его раз в месяц) может оказаться избыточным, и он начнет записывать пароль на мониторе или придумывать другие способы облегчения жизни. Возможно, в этом случае разработчикам следует задуматься о способе аутентификации *single-sign-on (SSO)* — *один вход*, когда для использования различных информационных систем требуется только однократный ввод пароля или однократная аутентификация с помощью системы токенов или биометрии.

Средства обратной связи с пользователями для обеспечения такого тестирования жизненными обстоятельствами можно придумывать различными. Например, это могут быть беседы с наиболее квалифицированными пользователями или опросы с помощью анкет.

Резюме данного раздела таково. Только тестирование порой может быть наиболее существенным критерием качества механизма или процедуры информационной безопасности.

Порядок и методика тестирования



Вопросы создания методики

Перед тем как привести выдержки из конкретных документов и практические рекомендации, необходимо сказать, что детальный документ по тестированию исключительно специфичен для каждой информационной системы. Поэтому любые общие рекомендации необходимо перерабатывать для конкретного случая.

Второе важное замечание касается организаций, существующих уже несколько (минимум 4—5) лет. Скорее всего такие предприятия уже прошли через процесс подготовки к проблеме 2000 года. Возможно, проблема оказалась несколько преувеличенной средствами массовой информации, но из самого процесса подготовки был извлечен важный урок — специалисты получили опыт в инвентаризации и тестировании различных информационных систем. Нарботанные за этот период документы можно использовать как основу для разработки нормативов по тестированию с точки зрения информационной безопасности.

Пример методики тестирования: "Проблема 2000"

В данном разделе приводится выдержка именно из методики тестирования программного обеспечения на соответствие 2000 году по параметру ДАТА.

ОПРЕДЕЛЕНИЯ

Дата_до — дата за один день до критической даты N.

Дата — критическая дата N.

Дата_после — дата, следующая за критической датой N.

Время_до — время за одну минуту до перехода Дата_до/Дата.

Время — время перехода Дата_до/Дата.

Время_после — время после перехода Дата_до/Дата.

ТПО — тестируемое программное обеспечение.

Форвардная проверка

Для тестирования ТПО выставить на аппаратных средствах (BIOS, system и т. п.) Дата_до. Затем проверить все места ТПО, где в сроке окончания расчета или процесса указывается будущая дата, путем проставления в поле Дата и Дата_после. Нужно также проконтролировать промежуточные результаты конкретного процесса или расчета, а также финальные, связанные с окончанием цикла, работы ТПО (концом дня, подсчетом общей суммы, расчетом проекта и т. п.).

Обратная проверка

Выставить на аппаратных средствах, используемых для тестирования ТПО (BIOS, system и т. п.), дату Дата_до, после чего проверить все места, где в сроке начала расчета или процесса указана прошлая дата, путем проставления в ее поле Дата и Дата_после, фиксируя промежуточные результаты конкретного процесса или расчета. Затем нужно проконтролировать финальные результаты по окончании цикла работы ТПО (в конце дня, при подсчете общей суммы, расчете проекта и т. п.).

Визуальная проверка

1. Выставить на аппаратных средствах (BIOS, system и т. п.) Дата_до и Время_до. Запустить модуль ТПО на исполнение и визуально отследить все поля, в которых дата отображается для пользователя. При этом необходимо убедиться, что дата не допускает неоднозначного толкования. Не прерывая сеанса работы ТПО, нужно дождаться наступления момента Время. После этого в течение Времени_после повторно рассмотреть все поля, где для пользователя отображается дата, и убедиться, что ее нельзя толковать неоднозначно. Если необходимо, следует вернуть аппаратное время к моменту Время_до и проверить поля, которые не успели проконтролировать.
2. Снова выставить на аппаратных средствах (BIOS, system и т. п.) Дата_до и Время_до и запустить модуль ТПО на исполнение. Затем выйти из сеанса ТПО, дождаться наступления Время и снова запустить ТПО. В течение Времени_после повторно рассмотреть все поля, в которых отображена дата для пользователя. Следует убедиться, что ее нельзя толковать неоднозначно. Если необходимо, следует вернуть аппаратное время на Время_до и проверить поля, которое не успели проконтролировать.

Проверка неквалифицированного пользователя

Провести прямую и обратную проверки, вставляя в поля результатов расчетов (процессов) неверные даты (вроде 3001) или те, которые можно неоднозначно толковать (например, 02).

Проверка ввода даты

Выставить на аппаратных средствах (BIOS, system и т. п.) Дата_до и Время_до и запустить модуль ТПО на исполнение. Далее нужно рассмотреть режим ввода, относя его к текущей дате в параметре ТПО, очередном по инвентаризационному списку. Вводя данные до истечения Время_после, проверить их из текущего сеанса работы ТПО. После возобновления этого сеанса прекратить текущую сессию и запустить новую внешними средствами без участия ТПО, если это возможно.

Внешний обмен данными

1. Выставить на аппаратных средствах (BIOS, system и т. п.) Дата_до и Время_до и запустить модуль ТПО на исполнение. Войти в режим экспорта данных с участием текущей даты в соответствующем параметре ТПО. Далее импорт данных выполнить как минимум три раза в периоды Время_до, Время_после, а также при условии, чтобы процесс экспорта данных включал в себя Время. Завершить сессию работы с ТПО и проверить полученные в результате этого данные.
2. Подготовить данные для импорта таким образом, чтобы они включали Дату_до, Дату и Дату_после. Выставить на аппаратных средствах (BIOS, system и т. п.) дату Дата_до и Время_до, а затем запустить модуль ТПО на исполнение. Войти в режим импорта данных в ТПО, соответствующий параметру. Выполнять импорт данных как минимум три раза в периоды Время_до, Время_после, а также при условии того, что процесс импорта включает в себя Время. Завершить сессию работы с ТПО, проверить полученные импортированные данные.

Архивирование

1. Выставить на аппаратных средствах (BIOS, system и т. п.) дату Дата_до и Время_до. Запустить модуль ТПО на исполнение, войти в режим архивирования данных в соответствующем месте ТПО. Выполнять архивирование данных, как минимум, три раза в периоды Время_до, Время_после, а также при условии, чтобы процесс охватывал данные с Время. Завершить сессию работы с ТПО, проверить полученные архивированные данные внешними средствами, если это возможно.
2. Выставить на аппаратных средствах (BIOS, system и т. п.) Дата_до и Время_до. Запустить модуль ТПО на исполнение, войти в нужное место

в режиме разархивирования данных, которые были ранее архивированы (при проверках по предыдущему пункту). Выполнять операцию как минимум три раза в периоды `Время_до`, `Время_после`, а также при условии, что процесс разархивирования данных включает в себя `Время`. Далее проверить полученные разархивированные данные средствами ТПО, а также после завершения сессии ТПО с помощью внешних средств, если это возможно.

Исправление даты

Выставить на аппаратных средствах (BIOS, system и т. п.) `Дата_до` и `Время_до`. Запустить модуль ТПО на исполнение. Войти в режим корректировки данных типа ДАТА в программном обеспечении. Выполнять корректировку дат как минимум три раза в периоды `Время_до`, `Время_после`, а также тогда, когда процесс корректировки дат включает в себя `Время`. Проверить откорректированные данные средствами ТПО, а также по завершении сессии работы ТПО с помощью внешних средств, если это возможно.

Сортировка

Выставить на аппаратных средствах (BIOS, system и т. п.) `Дата_до` и `Время_до`. Запустить модуль ТПО на исполнение. Войти в режим сортировки данных по дате в ТПО, по соответствующему параметру. Выполнять сортировку по дате как минимум три раза в периоды `Время_до`, `Время_после`, а также тогда, когда процесс сортировки включает в себя `Время`. После каждого раза проверить полученные отсортированные данные средствами ТПО, а также по завершении сессии ТПО с помощью внешних средств, если это возможно.

Если в ТПО включена обработка процесса индексации данных с участием даты, то необходимо дополнительно проверить все перечисленные этапы с учетом индексирования.

Вычисление

Выставить на аппаратных средствах (BIOS, system и т. п.) дату `Дата_до` и `Время_до`. Запустить модуль ТПО на исполнение. Войти в режим вычисления данных с участием даты в ТПО, по соответствующему параметру. Выполнять вычисления с участием даты как минимум три раза в периоды `Время_до`, `Время_после`, а также тогда, когда процесс вычисления включает в себя `Время`. Проверить вычисленные данные средствами ТПО, а также после ее завершения с помощью внешних средств, если это возможно.

Пассивная проверка

Выставить на аппаратных средствах (BIOS, system и т. п.) `Дата_до` и `Время_до`. Запустить модуль ТПО на исполнение. Войти поочередно во все режимы ввода, обработки и/или другого использования текущей даты и вы-

ставить ее. Завершить сессию работы с ТПО. После наступления `Время_после` запустить модуль ТПО и проверить корректность введенной даты, а результаты внести в акт. Повторить эксперимент, выключая аппаратное обеспечение на время прохождения `Время`.

Анализ результатов

Для того, чтобы оценить результаты тестирования, необходимо знать, насколько успешно или неуспешно оно завершилось. Поскольку итогом тестирования должно быть выявление определенного количества ошибок, необходимо проклассифицировать предполагаемые ошибки и вывести предварительные прогнозы их появления.

Предполагаемые ошибки.

- ☐ Ошибка класса А. Система не может работать, либо работает неприемлемо, либо при работе создает серьезные помехи или вмешательство в работу других систем, искажает данные.
- ☐ Ошибка класса Б. Система работает с перебоями или задержками, либо не функционирует важный, но не критический модуль системы, либо полнофункциональная работа одного из модулей требует серьезных дополнительных ресурсов.
- ☐ Ошибка класса В. Не работает вспомогательный или несущественный модуль системы, либо работа одного или небольшой группы пользователей затруднена.

Предварительные прогнозы.

- ☐ При наличии хотя бы одной ошибки класса А система не может быть принята для дальнейшего тестирования, тем более в эксплуатацию.
- ☐ Система может быть принята для дальнейшего тестирования и обучения конечных пользователей при наличии не более одной ошибки класса Б.
- ☐ Система может быть принята в ограниченную эксплуатацию (на определенный срок) при наличии не более двух ошибок класса В.

Общая схема тестирования

Другой пример: каркас плана тестирования, опубликованного на сайте http://quality.boom.ru/articles/plan_ex.html; ознакомиться с его подробностями можно там же, хотя большая часть пунктов понятна и без пояснений.

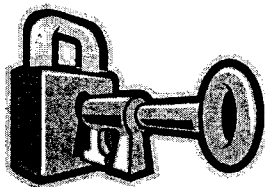
Основные пункты тестового плана

- ☐ Введение.
 - Цель.
 - Область применения.

- Ссылки.
- Допущения.
- Обзор тестирования.
 - Условия выполнения тестов.
 - Вопросы аппаратного обеспечения.
 - Вопросы программного обеспечения.
 - Критерии выполнения.
- Тестовые процедуры.
 - Блочное тестирование.
 - Интеграционное тестирование.
 - Тестирование контроля версий.
 - Функциональные тестовые процедуры.
 - Тестирование пользовательского интерфейса.
 - Отсутствующее тестирование.
 - Пользовательские сценарии.
 - Сценарий 1.
 - Сценарий 2.
 - Сценарий 3.
 - Сценарий 4.
 - Сценарий 5.
 - Системное тестирование.
 - Нагрузочное тестирование.
 - Эксплуатационное тестирование.
 - Тестирование надежности.
 - Тестирование установки.

В дополнение укажем еще один адрес. Это интересный сборник вопросов и ссылок на готовые решения по тестированию — <http://www.s-test.elistel.net/FAQ/faq.htm> .

Кроме того, необходимо упомянуть автоматизированные средства тестирования программных продуктов. Хотя их использование характерно в основном для сертификационных лабораторий и институтов, тем не менее ознакомиться с ними полезно и для специалистов по безопасности. Достаточно подробно автоматизированные средства поиска уязвимостей при проектировании и реализации программных продуктов описаны в книге [Лукацкий2].



Часть VIII

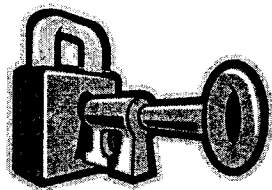
Если нарушение безопасности все же произошло

Глава 31. Аварийный план

**Глава 32. Реагирование на нарушения
информационной безопасности**

Глава 33. Проведение расследования

Глава 31



Аварийный план

Что такое аварийный план?



Говоря об информационной безопасности в околопрофессиональных кругах, об аварийном плане упоминают редко. Действительно, гораздо интереснее обсуждать построение эффективной эшелонированной защиты, применять термины *брандмауэр*, *взломщики*, *вирусная атака*, нежели обсуждать — что делать, если нарушение информационной безопасности все же произошло. Тем более, если оно произошло не по причине высокоинтеллектуального взлома, а по причине сбоя в электросети. Скучно и хочется переложить всю заботу о таком происшествии на электронщиков, пожарников и кого угодно, только не на специалистов службы информационной безопасности. Между тем, если пропали данные, то произошло именно нарушение *доступности*, как категории информационной безопасности, а значит, заниматься этим вопросом придется службе информационной безопасности.

Если проанализировать статистику наиболее часто встречающихся реализаций угроз информационной безопасности, то на первом месте стоят именно причины аварийного характера (сбои сети, природные катаклизмы и т. д.), следующими идут инциденты по причинам ошибок персонала и только затем атаки злоумышленников. Поэтому, вероятнее всего, чаще придется заниматься именно скучными вещами.

Так что же делать, если нарушение безопасности (не важно, скучное или увлекательное) все же произошло? Срочно собирать специалистов на совещание? А может звонить консультантам и поставщикам? Бегать и совершать непродуманные действия? Гораздо лучше достать документ, где подробно описана ситуация, в которой оказалась организация в данный момент, расписаны действия персонала, а также то, как найти этот персонал; документ, согласно которому выполнен ряд предварительных мероприятий, значительно сокративший возможный ущерб, и многое другое. Причем это руководство создано в спокойной обстановке, когда можно было не торопясь обдумать все варианты и выбрать из них оптимальный. Документ протестирован, причем на практике, и весь персонал чувствует себя в аварийной ситуации не в панике, а в обстановке, сходной с тренировочной.

Аварийный план служит ответом на ряд важных вопросов. Что делать, чтобы снизить возможные последствия аварии? Какие действия выполнять, когда аварийная ситуация находится в стадии развития? Что делать сразу после завершения аварии? Как вернуться к обычной деятельности?

Принципы создания аварийного плана



Общие соображения

Конкретный аварийный план для предприятия — это очень индивидуальный документ, который зависит не только от принадлежности организации к той или иной отрасли, особенностям бизнеса и т. д., но и от географических особенностей местности, других предприятий, расположенных поблизости, состава и профессиональных качеств сотрудников, входящих в аварийную команду, и многого другого.

Используемая профессиональная терминология *Disaster recovery* (восстановление после несчастных случаев), *Business contingency* (непредвиденные обстоятельства в бизнесе), *Fraud control* (борьба с мошенничеством) формально разделяет некоторые понятия аварийного плана, также предлагаются различные модели построения такого плана, вплоть до автоматизированных средств создания плана. В данной главе будут изложены рекомендации по созданию плана, наиболее близкого авторам, а также выдержки из примеров аварийных планов.

Составление аварийного плана следует начинать с определения — а что такое *авария*? Авария — это нештатная ситуация, которая требует действий, отличных от рутинных, определенных регулярными процедурами, т. е. ввод в действие того или иного элемента плана.

Классификация событий

При этом необходимо выбрать такую классификацию событий, которая бы наиболее эффективно работала бы при реализации плана. Возможно, не стоит разделять в классификации пожар, наводнение и землетрясение, если все эти катастрофы потребуют с точки зрения плана одинаковых действий. В таком случае следует принять иную классификацию. Например, разделить их на события природные и зависящие от человеческого фактора (такая классификация различает способы предварительной подготовки), на требующие эвакуации персонала и не требующие (здесь существенно разнятся действия, происходящие во время самого события) и т. д.

Большинство природных катастроф можно классифицировать, как временная или постоянная недоступность некоторого информационного обеспечения (оборудования, программ, данных), а в внутри этого класса подклассифицировать, как в рамках участка или части одного подразделения или

здания (кража или разрушение оборудования), затрагивающая все здание (пожар), затрагивающее все учреждения предприятия в данной местности (землетрясение, война). Можно производить классификацию и с точки зрения угроз информационной безопасности, однако при этом следует учесть, что на первое место всегда выходит угроза жизни и здоровью людей, а лишь затем — угрозы всему остальному (возможно, для ряда организаций на втором месте будут также не информационные угрозы, а угрозы материальным ресурсам — зданиям, ценностям и т. д.)

Классификация объектов по приоритетам

Следующим шагом будет определение приоритетов восстановления систем и объектов. В этом случае для упрощения процесса следует идти от верхнего уровня, например:

- ☐ основная цель предприятия — обслуживание клиентов;
- ☐ обслуживанием клиентов занимаются сотрудники;
- ☐ клиенты обслуживаются с помощью автоматизированной системы;
- ☐ автоматизированная система требует приложения и данных;
- ☐ приложения и СУБД требуют операционной системы и коммуникаций;
- ☐ операционная система требует аппаратного обеспечения;
- ☐ коммуникации требуют сетевого обеспечения;
- ☐ оборудование требует электропитания, возможно охлаждения;
- ☐ электропитание требует источника и средств доставки;
- ☐ сотрудники требуют определенных условий труда (помещение, освещение, возможно, питание).

Когда речь идет об информационных системах (в нашем списке — третий пункт), необходимо учитывать, что не все системы после катастрофы потребуются незамедлительно после перехода к восстановлению бизнеса, т. е. информационные системы следует проклассифицировать с точки зрения аварийного плана (классификация будет близка к той классификации информационных объектов и систем, которая была приведена в главе 6).

Системы могут быть:

- ☐ критически важными (без них невозможно продолжение бизнеса) с подразделением на постоянно и периодически требующиеся;
- ☐ для вторичных и вспомогательных бизнес-задач;
- ☐ для рутинной работы, в которых потребность возникнет, но не в первые часы/дни после катастрофы;
- ☐ необходимые для перспектив развития предприятия.

Шкала резервных объектов

Таким образом, в случае катастрофы для восстановления деятельности можно определить то, что необходимо иметь. Если сохранились здания и коммуникации, значит, необходим следующий этап — оборудование и др. Однако следует учесть самый негативный вариант — когда организация понесла максимальные потери. Допуская такой сценарий, можно подняться по представленному списку снизу вверх.

- ❑ Для продолжения бизнес-деятельности необходимы помещения. Возможно, в каких-то экзотических случаях допустимо начать работу под открытым небом или в палатках, но мы не будем ориентироваться на такие варианты.
- ❑ Помещения должны быть снабжены соответствующей инфраструктурой (электропитание, водопровод и т. д.) либо необходимо создать эту инфраструктуру в кратчайшие сроки.
- ❑ Должна быть установлена локальная сеть и организованы выходы во внешние информационные пространства, если этого требует бизнес. Возможно, сеть при катастрофе сохранится полностью или частично, но потребует замены части сетевого оборудования.
- ❑ Предприятию желательно позаботиться о запасе аппаратного обеспечения, либо договориться с поставщиком об оперативной поставке оборудования.
- ❑ У предприятия должны быть инсталляционные копии операционных систем, СУБД, приложений и других программ либо возможность оперативно получить эти копии у поставщиков. Немаловажным вопросом является наличие копий конфигурационных настроек программных комплексов, которые специфичны для данной организации (если такие есть).
- ❑ У предприятия должны быть резервные копии данных, причем максимально приближенные по актуальности к моменту времени непосредственно перед катастрофой.
- ❑ Сотрудники предприятия, которые необходимы для обеспечения работы информационных систем (администраторы и т. п.) и работы с клиентами, должны быть готовы исполнять свои функции, либо быть оперативно заменены сотрудниками другого учреждения данной организации, либо оперативно наняты со стороны.
- ❑ Необходимо обеспечить доступ клиентов к сервису организации. Если это удаленный сервис — желательно обеспечить наличие соответствующих коммуникаций (Интернет, выделенные или коммутируемые каналы и т. д.). Если для обслуживания клиенту требуется прибывать в организацию, то следует обеспечить его доставку и т. д.



Структура аварийного плана

Структурированный аварийный план должен состоять из нескольких частей, возможно даже это будут отдельные документы, тем не менее, приведенные элементы должны присутствовать в нем обязательно.

Определения аварийных ситуаций и расстановка приоритетов по объектам и системам

О задачах, описываемых этим документом, шла речь в *разд. "Принципы создания аварийного плана"*, и работа по определению и расстановке приоритетов должна быть проделана до начала составления плана. Результаты этой работы необходимо формально закрепить в утвержденном руководством предприятия плане с тем, чтобы при наступлении аварийной ситуации не возникали бы дополнительные споры — какие мероприятия проводить и какие системы в какой очередности восстанавливать.

Предварительные меры

Предварительные меры содержат описание мероприятий и процедур, которые должны быть произведены или регулярно проводится для того, чтобы при возникновении аварийного события устранить или существенно снизить размер ущерба для предприятия. Поскольку речь все же идет об информационной безопасности, далее будут проанализированы в основном мероприятия, связанные именно с информационными объектами и системами, оставив в стороне другие, иногда не менее важные аспекты (связанные с жизнью и здоровьем людей). Также будем предполагать, что аварийный план составляется для аварийных событий, максимальным из которых будет полное уничтожение информационного пространства предприятия.

Основой восстановления должно быть создание нового места работы — существования восстановленного информационного пространства. Эту задачу можно не рассматривать только в том случае, если уничтожение текущего места размещения означает прекращение деятельности предприятия. В остальных случаях организации необходимо здание с имеющейся информационной и вспомогательной (электропитание, водоснабжение, телефонная связь и т. д.) инфраструктурой. Возможные варианты для этого случая могут быть в зависимости от финансовых возможностей компании.

❑ Содержание отдельного запасного здания, принадлежащего предприятию. Возможны различные конфигурации готовности резерва:

- "горячий" — место полностью копирует рабочую обстановку (возможно в несколько усеченном виде), только пользователям осталось разместиться за своими новыми местами и включить рабочие станции (серверы уже могут работать);

- степени "постепенного охлаждения" — например, локальная сеть присутствует только в виде кабелей и пассивного оборудования, требуется установка дополнительного (активного) оборудования, электропитание подведено, но не подключено и т. д.;
 - "холодный" — собственно только само здание и инфраструктура электропитания и санитарно-технического оборудования.
- Содержание выделенного здания совместно с другой организацией (организациями) с условием одновременного или поочередного использования при катастрофе.
- Содержание выделенного здания с возможным частичным использованием его в обычной жизни, например, как склад или место расположения зеркального рабочего сервера и т. д.
- Переезд в случае катастрофы в здание одного из подведомственных учреждений (отделение, филиал) или к организации-партнеру.

При наличии альтернативного запасного здания необходимо обеспечить его охрану и регулярно проверять его готовность к принятию рабочей нагрузки.

Если предположить, что у организации нет "горячего" резерва компьютерного оборудования, стоящего "наготове" в ожидании аварии, значит, надо обеспечить запас такого оборудования на складе предприятия (желательно сделать запас распределенным по различным складам с тем, чтобы запасное оборудование не погибло вместе с основным в случае масштабной катастрофы), либо обеспечить получение оборудования в кратчайшее время после катастрофы. Второй вариант предполагает предварительные договоренности с поставщиками или организацией-партнером, которые предоставят некий лимит оборудования во временное пользование. Это предполагает, что к аварийному плану прилагается предварительно согласованный договор на поставку оборудования. То же самое должно относиться и к другим объектам, в которых организация зависима от внешних поставщиков — электропитание, линии связи и т. д.

Следующий важный элемент — сохранение инсталляционных комплектов (дистрибутивов) информационных систем — системного и прикладного программного обеспечения. Необходимо учитывать, что восстанавливать, т. е. заново устанавливать, придется, возможно, в спешке и совсем не тем специалистам, которые проводили установку систем на обычном месте работы. Это не означает, что в аварийный план необходимо переписать руководства по инсталляции всех систем, однако, если при установке существуют специфические детали, требующие особого внимания, — о них нужно обязательно упомянуть. Особое внимание следует уделить специфическим настройкам — какие дополнительные сервисы должны быть включены и как настроены, порядок взаимодействия между ними и т. д.

Восстанавливая информационное пространство, нельзя забывать и об обеспечении его защиты и, чтобы не слышать в дальнейшем высказыва-

ний типа, сейчас аварийная ситуация, пару дней поработаем без антивирусов и межсетевых экранов, следует внести в аварийный план пункт о восстановлении систем безопасности с заранее выбранным приоритетом. Если данная информационная система позволяет сохранять свои настройки в отдельных конфигурационных файлах, значит, следует сохранять копии актуальных из них наряду с резервными копиями данных.

В зависимости от специфики работы предприятия данные могут составлять значительную или даже основную значимость организации. Иначе говоря, эти данные и могут представлять из себя организацию, а их отсутствие означает закрытие предприятия как такового. Вопросам сохранности данных посвящены специальные публикации, приведем только основные пункты из них.

Данные должны быть проклассифицированы с точки зрения аварийного плана (в этом поможет классификация информационных активов по разделу "доступность") на категории. В литературе по этому вопросу можно встретить термин *Управление структурированным хранением* (англ. *Hierarchical Store Management — HSM*), которое подразумевает разделение данных:

- ❑ на *первичные*, или *оперативные*, данные, т. е. которые должны быть постоянно доступны;
- ❑ *вторичные*, или *полуоперативные*, данные, т. е. которые могут стать доступными автоматически, при возникновении необходимости;
- ❑ *третичные*, или *автономные*, данные, т. е. которые станут доступными после выполнения ряда ручных операций.

Учитывая эту классификацию, становится понятным, почему резервное копирование данных, оставаясь важной составляющей, тем не менее не является единственным средством и не всегда панацея от аварий. Например, для организации, предоставляющей он-лайн услуги, крах сервера с данными, требующими восстановления с резервных копий в течении нескольких часов, может означать очень серьезные потери, превышающие расходы на приобретение дублирующего (зеркального) сервера.

Вопросам обеспечения бесперебойной работы дисков и дисковых массивов можно также посвятить отдельное издание. Всем заинтересовавшимся читателям рекомендуем обратиться к документам, в которых обсуждаются на соответствующем техническом уровне технологии *RAID (Redundant Arrays of Independent Disks)* — зеркалирования, дуплексирования и т. д. В разд. "*Классификация технологий RAID*" данной главы, приводится краткая классификация различных вариантов данной технологии. Кроме того, отметим, что среди современных технологий есть возможности использования территориально разнесенных дублирующих узлов (географически разнесенный кластер), а также возможность аренды дискового, процессорного и других ресурсов у стороннего провайдера хоть на другом конце света.

Поскольку, резервное копирование — это отдельный большой вопрос, некоторые его аспекты, которые должны быть упомянуты, перенесены в *разд. "Сохранение данных"* данной главы, чтобы не загромождать общий ход описания аварийного плана.

На предварительном этапе также необходимо определить уполномоченных лиц — комитет, рабочую группу или иное объединение, которые будут осуществлять управление действиями сотрудников во время наступления аварии и при восстановлении. Возможно, оперативной и восстановительной деятельностью будут заниматься различные лица.

Рассматривая возможность удаленного внешнего сохранения резервных копий, следует учесть, что они также должны быть предметом для применения мер безопасности. Одна из существующих распространенных ошибок — отсутствие защиты резервных копий, а между тем в них может присутствовать сверхкритическая информация, такая как база паролей пользователей и администраторов. Нормальным способом сохранения данных на внешних носителях вне защищенного и контролируемого размещения в центре является применение сильных криптографических механизмов защиты данных.

Заканчивая разговор о предварительных мероприятиях, необходимо сказать, что обсуждаемые на протяжении всей книги процедуры и механизмы безопасности также относятся к предварительным мероприятиям аварийного плана. Просто они носят характер защиты не от катастроф глобального или природного характера, а от катастроф, спровоцированных злоумышленниками, ошибками персонала и т. д.

Мероприятия во время катастрофы

А теперь рассмотрим активность, которая наступает во время аварийного события и сразу же после аварии до того момента, когда предприятие начинает продолжение производственного процесса. Один из авторов имел возможность практической оценки важности данного раздела при устранении последствий террористического акта в крупном финансовом учреждении.

Несмотря на то, что мы уже несколько раз повторили, что при описании аварийного плана мы ориентируемся на аспект информационных систем, тем не менее повторно укажем, что в первую очередь необходимо обеспечить отсутствие угрозы жизни и здоровью людей либо, если ущерб людям уже нанесен, постараться максимально сократить негативные последствия и обеспечить пострадавших соответствующей помощью — медицинской, транспортной и т. д. Этот пункт также подразумевает необходимые меры для оперативного информирования работников организации, клиентов и других лиц, находящихся в сфере действия мероприятий, о наступающем или наступившем аварийном событии, мерах по эвакуации людей в безопасное место, обеспечению порядка и предотвращения паники и др.

Если нет угрозы жизни людей, необходимо сразу же позаботиться о сохранности информационных ресурсов. Для этого сначала проанализируйте, в каких случаях, какие действия, для каких систем потребуются. Возможно, если потребуется эвакуация персонала без непосредственной немедленной угрозы жизни и здоровью, то целесообразно произвести корректное выключение рабочих станций. Может быть, требуется перевод серверов в сокращенный режим работы (при наступлении аварии типа "распределенная атака на отказ в обслуживании" — DDoS — одним из стандартных требований является отключение на маршрутизаторах и межсетевых экранах поддержки обработки трафика второстепенных сервисов). Таким образом, необходимо:

- ❑ на этапе подготовки определить и формализовать действия, требуемые для выполнения в момент аварии;
- ❑ при наступлении аварийного события идентифицировать, к какому виду событий оно относится с точки зрения необходимых мероприятий;
- ❑ проинформировать работников, которые будут задействованы в мероприятии, о наступлении данного вида события;
- ❑ обеспечить информированность работников о необходимых действиях и проверить готовность к их выполнению.

Последний пункт следует рассмотреть подробнее. Его особенность заключается в необходимости учитывать то, что описываемые мероприятия будут производиться в момент аварии или катастрофы, в условиях нервозности, стресса или даже шока, когда действия, кажущиеся простыми и понятными в обычных условиях, могут оказаться затруднительны к выполнению.

Примечание

Специалисты рекомендуют домохозяйкам держать список телефонов важных служб на видном месте возле телефонного аппарата, так как бывали случаи, когда в панике люди не могли вспомнить номер пожарной службы 01. А теперь представьте, что работнику придется в экстремальных условиях производить настройку сложной информационной системы.

Для обеспечения эффективной работы работник должен иметь под рукой подробную инструкцию:

- ❑ о том, кому из ответственных сотрудников предприятия позвонить для сообщения о возникновении аварийного события, для вызова на место происшествия и консультаций — со всеми возможными номерами телефонов (рабочий, домашний, мобильный), пейджеров и других средств связи;
- ❑ что конкретно делать для данной информационной системы — запуск исполняемых модулей или скриптов, дополнительное аварийное копирование, внесение изменений в конфигурацию и т. д.;
- ❑ где находятся дополнительные средства, необходимые для выполнения мероприятий, — средства резервного копирования, дополнительные инструкции, огнетушители и т. п.

Подобные инструкции должны быть написаны простыми доступными словами, без использования сложных технических терминов с "разжевыванием" всего, что казалось бы, и так всем известно. Примерами таких инструкций могут быть следующие.

- ❑ *Открыть большой серый шкаф справа от входа в помещение, если стоять лицом к двери. На второй полке сверху взять синюю коробку и вынуть из нее компакт диск и надпись "Экстренная резервная копия".*
- ❑ *В главном меню системы выбрать раздел Options, затем подраздел Services. В открывшемся окне выбрать закладку Emergency и на появившейся странице установить флажки 1 и 2 в положение включено, щелкнув по ним левой кнопкой мыши до появления в ячейке символа X.*

После выполнения всех требуемых действий уполномоченные лица должны проанализировать имеющуюся ситуацию. Возможно, при каких-либо нестандартных условиях потребуются дополнительные работы, о которых не было известно на этапе составления и тестирования плана.

После того, как все возможные действия, направленные на сокращение ущерба, выполнены, необходимо произвести оценку понесенного ущерба; не исключено, что для этого потребуются привлечение сторонних арбитражных организаций. Материал, полученный на этом этапе, будет нужен при обращении в страховые компании, для предъявления исков и т. п.

Возвращение к бизнесу

Как только воздействие аварии или катастрофы закончено, а иногда, возможно, и во время продолжения воздействия, организация принимает решение о возврате или продолжении своего бизнес-процесса. В зависимости от результатов аварии и составленного плана это может быть бизнес-процесс в полном объеме, такой же, каким он был до наступления аварии, либо в сокращенном объеме на определенный период времени.

В зависимости от созданного на подготовительном этапе плана необходимо начать восстановление функционирования систем по predetermined приоритетам от критических к менее важным. Уже были рассмотрены степени возможного поражения предприятия и степени готовности к подобным случаям, поэтому лишь напомним, что это может быть перемещение в новое здание, получение или приобретение нового оборудования, переустановка системного и прикладного программного обеспечения, восстановление данных с резервных копий. Таким образом, часть плана должна включать следующие данные.

- ❑ Адрес нового месторасположения предприятия и систему доступа на новое место (документы, пропуска, идентификаторы), схемы подходов и подъездов, график движения транспорта и т. д.

- ❑ Способы получения оборудования — адреса и телефоны поставщиков, гарантийных и ремонтных центров, копии или ссылки на договора поставок и поддержки, предварительно согласованные договора на поставку нового оборудования и т. д.
- ❑ Адреса и телефоны своих и привлеченных специалистов, которые будут заниматься установкой и приведением систем в рабочее состояние.
- ❑ Место хранения инсталляционных копий программного обеспечения, копии конфигураций систем, специфических для данного предприятия. Для формирования последних еще на предварительной фазе особенно актуально зафиксировать все изменения настроек, проводимые для работы (это бывает затруднительно, так как в отличие от резервного копирования данных может происходить нерегулярно).
- ❑ Место хранения резервных копий данных и порядок их получения.
- ❑ Правила восстановления данных из резервных копий. Они должны быть не менее простыми и понятными, чем инструкции по действиям во время самой аварии, так как восстановление работоспособности может происходить в экстремальных условиях, в сжатые сроки, при постоянном появлении дополнительных задач и т. д.
- ❑ Возможно, процедуры по дополнительному обучению или тренировкам пользователей для работы в новых условиях.

Если работа первоначально восстанавливается в сокращенном объеме, значит, план должен предусматривать переход к полнообъемному функционированию с указанием этапов (контрольных точек), после которых выполняется следующая фаза перехода.

Методология работы с аварийным планом



Извлечение уроков

Когда высвобождаются ресурсы, направленные на возвращение бизнеса в обычный режим, а для ряда организаций или видов аварий одновременно с этим или даже предшествуя нормализации бизнеса, должно быть произведено расследование. Расследованию нарушений информационной безопасности будет посвящена *глава 32*, а здесь поделимся общими соображениями.

Не следует думать, что расследование необходимо производить только в случае подозрения на злой умысел, как на источник аварии. К сожалению, и при природных катаклизмах, когда большинство сотрудников предприятия

будет занято сокращением ущерба и восстановлением бизнеса, могут найтись люди, которые постараются использовать нестандартную ситуацию в собственных интересах. Если служба безопасности не может сразу обеспечить полный контроль всех действий сотрудников в момент наступления аварии и при восстановлении бизнеса, то сразу же при появлении возможности, высвобождении ресурсов должна быть произведена своего рода инвентаризация использования корпоративных ресурсов за критический период, с возможным охватом стадии до начала аварии и после восстановления бизнеса.

Дополнительно к этому действия персонала предприятия в период реализации аварийного плана являются приобретением опыта, возможно, и негативного. Следовательно, он становится источником для анализа уполномоченными специалистами с целью последующего улучшения плана. Таким образом, само наступление аварии можно рассматривать с определенными "натяжками" как вариант тестирования аварийного плана.

Тестирование плана

Тестирование — один из самых существенных вариантов проверки правильности подготовки аварийного плана. Проблема заключается только в практической невозможности полного его тестирования, ведь для этого пришлось бы создать аварийную ситуацию со всеми вытекающими последствиями. Поэтому предлагаем тестировать:

- ❑ аварийный план по составляющим, отдельным системам в выделенной тестовой среде (тестовом оборудовании, тестовом сегменте сети, с тестовыми данными) в режиме, максимально приближенном к реальной аварии;
- ❑ реальные отдельные системы — в шадящем режиме с минимальным влиянием на производительность и работу;
- ❑ весь комплекс составляющих, объектов и систем — по предопределенным сценариям, обычно при временном прекращении или приостановлении рабочей деятельности.

Возможно также последовательно пройти по всем вариантам, на каждом этапе внося в аварийный план необходимые изменения и дополнения. Следует только учесть, что если в первом варианте организация тестирования возможна в рабочем порядке, то уже во втором потребуются содействие администраторов и пользователей систем, а третий вариант возможен только при привлечении руководства предприятия. Очевидно, что такие полномасштабные тестирования не могут проводиться часто, чтобы не оказать негативного влияния на работу организации. Компетентные специалисты по безопасности называют сроки, в зависимости от размеров предприятия, от одного раза в полгода до одного раза в два года. Более редкое тестирование будет нецелесообразно, так как нетестированный

план отстанет от жизни и перед очередным тестированием его придется предварительно перерабатывать.

В западных источниках можно встретить следующие рекомендованные стадии тестирования аварийного плана (в данном случае: Disaster recovery plan — Плана восстановления после несчастного случая).

1. *Контрольная проверка* (англ. *checklist*). Копии плана распределяются между различными подразделениями, которым предлагается просмотреть его и убедиться, что в плане отражено все, что необходимо.
2. *Структурированный обзор* (англ. *structured walk-through*). Руководители подразделений собираются вместе, чтобы в ходе совещания обсудить каждый из пунктов. Требуется убедиться, что разработанные мероприятия позволят эффективно противостоять аварии.
3. *Симуляция* (англ. *simulation*). Весь выделенный операционный и поддерживающий персонал выполняет тренировочные мероприятия по predetermined сценарию, в том числе перемещение на альтернативное место работы, используя только средства предназначенные для аварийного использования. При этом, однако, не выполняются собственно процедуры восстановления или альтернативной работы.
4. *Параллельная проверка* (англ. *parallel*). Весь персонал выполняет полное тестирование аварийного плана. Единственное отличие от реального аварийного события в том, что не производится остановка основного производственного процесса (в нем задействуется минимум требуемого персонала). При этом сравнивается эффективность производительности продолжающейся реальной деятельности и тестовой деятельности, выполненной на альтернативном месте расположения.
5. *Полное прекращение деятельности* (англ. *full-interruption*). Все происходит как при реальном наступлении аварии с остановкой основного производственного процесса. Вся деятельность перемещается на альтернативное место работы. При этом важно отметить необходимость кооперации предприятия с внешними органами управления (полиция, пожарная служба, транспорт и т. д.).

Обновление плана

Поскольку и в промежутках между этапами тестирования как основными источниками внесения изменений и дополнений жизнь не стоит на месте, план придется дорабатывать и изменять. Наиболее существенной причиной изменения будет внедрение новой информационной системы — в аварийном плане появится целый новый раздел, посвященный обеспечению рабо-

ты этой системы, а также изменения в тех частях плана, где новая система сопрягается с другими объектами и системами.

Другим минимальным примером изменения плана может служить увольнение с предприятия одного ключевого работника и замена его другим. Поскольку в плане присутствуют персональные контактные данные они должны быть обновлены со сменой в должности конкретного лица. Кроме того, возможно, новый человек не будет обладать достаточной квалификацией, и часть его обязанностей в рамках плана придется перераспределить между другими участниками, либо, наоборот, из-за высокой квалификации на него можно будет возложить дополнительные обязанности.

В любом случае необходимо, чтобы уполномоченный человек или группа контролировали изменения в информационном пространстве предприятия, затрагивающие аварийный план, и инициировали обновление плана при возникновении соответствующих условий. Особенно это может быть важно при переконфигурировании систем. Восстановление систем после сбоя подразумевает и тонкую настройку системы для данного конкретного предприятия. Если настройки, произведенные в системе, не были закреплены в одном из документов или приложений аварийного плана, то при восстановлении система может оказаться в конфигурации, предшествующей последним изменениям, и не отвечать текущим условиям работы.

Заканчивая разговор об аварийном плане, следует напомнить, что, кроме качества составления плана (подробности, ясности изложения и т. д.), он должен быть доступен соответствующему персоналу в критический момент. Это означает, что копии плана требуется подготовить не только в электронном виде, но и в печатных копиях.

Сохранение данных



Следует различать два способа сохранения данных.

- ❑ *Архивирование.* Сохранение копий данных, которые не требуются в постоянном оперативном использовании. Чаще всего это данные приложений за истекший период времени, которые могут потребоваться при необходимости исторической справки.
- ❑ *Резервное копирование.* Сохранение избыточных копий данных, используемых оперативно либо за ближайший период. Основное их предназначение — восстановление данных в случае утери или повреждения текущей копии, используемой системой.

Архивирование — это вопрос бизнес-процесса и безопасности, касается в основном обеспечения защиты архивных копий и разграничения доступа к ним. С точки зрения аварийного плана более важно резервное копирование, поэтому остановимся на нем подробнее.

Следует помнить, что при планировании резервного копирования необходимо найти оптимальный баланс между актуальностью данных и загруженностью системы. С одной стороны, данные, сохраненные в резервной копии, должны быть по времени максимально приближены к текущим данным, используемым системой (возможны случаи, когда наличие данных недельной давности равнозначно, полному их отсутствию). С другой стороны, ежечасный запуск процедуры резервного копирования может послужить серьезной помехой работоспособности системы.

Процедуры регулярного резервного копирования по видам разделяются следующим образом.

- ❑ *Полное.* Заданный набор данных (файлы, каталоги и т. д.) регулярно полностью переносится в альтернативное место сохранения. Резервная копия занимает большой объем и процесс растянут по времени, однако этот метод наиболее надежен (никакие данные не пропущены) и прост в аварийной ситуации — восстанавливается все сразу.
- ❑ *Наращиваемое, или инкрементальное.* После первой полной копии данных в каждую последующую помещаются только те данные, которые были модифицированы со времени выполнения предыдущей копии. Периодически, по завершению выбранного цикла, снова выполняется полное копирование, и наращивание начинается заново. Резервная копия занимает мало места, процесс копирования проходит быстро, однако восстановление растянуто во времени — сначала восстанавливается первая полная копия, затем последовательно все остальные. Кроме того, при неверно выбранном критерии модифицированности данных возможен пропуск части данных (которые были изменены, но по каким-либо причинам не попали в разряд модифицированных).
- ❑ *Метод различий, или дифференцированный.* После первой полной копии данных в каждую последующую помещаются только те данные, которые были модифицированы со времени выполнения первой полной копии. По своим характеристикам этот способ является промежуточным между первыми двумя. Время копирования и объем архива меньше чем у полной копии, но больше, чем у наращиваемой. Однако для восстановления требуются только первая и последняя копии.

Резервное копирование имеет ряд тонкостей, на которые следует обратить внимание. Но перед рассмотрением этих вопросов и проблем, необходимо провести еще одно разделение способов резервного копирования — *ручное и автоматическое*. Для небольшого предприятия, где все данные сохраняются на одном сервере, возможно поручить процедуру резервного копирования системному администратору, оставив на его усмотрение способ, время, критерии и т. д. Но для предприятия с несколькими десятками серверов, возможно, географически разнесенных, с сотнями гигабайт или терабайтами постоянно изменяющихся данных, открытых для использования 24 часа в

сутки, необходима отдельная система резервного копирования. Возможно, отдельная *сеть сохранения данных* (англ. *Storage Area Network — SAN*).

Для выбора и настройки системы резервного копирования данных требуется учитывать ряд особенностей ее функционирования.

- ❑ Выбор критериев того, была ли конкретная часть набора данных изменена со времени последней копии. По каким параметрам система будет это определять (размер, дата модификации, специальные флаги, совокупность различных параметров)?
- ❑ Какие действия необходимо предпринять, если в момент резервного копирования данный файл, подлежащий копированию, находится в процессе модификации? Подождать, пропустить, вернуться к попытке через некоторое время?
- ❑ Должна ли система при восстановлении перезаписывать те данные, которые уже имеются на месте восстановления? Возможно, эти данные более актуальны, чем восстанавливаемые? А если не перезаписать, то не возникнет ли рассогласование между актуальными данными и восстановленными? Восстанавливать ли данные, которые были санкционированно удалены в ходе бизнес-процесса?
- ❑ Если данные распределены между различными операционными или файловыми системами, — должна ли система резервного копирования быть одной для всех или для каждого отдельной? Сохранять ли все резервные копии единообразно (например, все на стримерные ленты) или для различных видов данных различно (что-то на жесткий диск, что-то на ленту, что-то на магнитооптику)?
- ❑ Возможно, одновременно с копированием данные необходимо шифровать?

Кроме перечисленных вопросов, касающихся собственно системы копирования, возникнут еще и вопросы о сохранении самих резервных копий. Если хранить их рядом с текущими данными, то серьезная катастрофа может уничтожить и основные данные, и резервные копии. Географическое разнесение копий возможно для многофилиального предприятия, а если все учреждения организации сконцентрированы в одном районе? Может быть, необходимо передать копии (естественно, позаботившись об их защищенности с точки зрения информационной безопасности) организации-партнеру или отдельной компании, предоставляющей услуги хранения, например, в банковский депозитный ящик?

Всех, заинтересовавшихся вопросами резервного копирования в объеме, большем чем материал этой книги, авторы предлагают для поиска информации еще один термин. Это протокол обеспечения систем резервного копирования механизмами контроля создания копий и восстановления — *Протокол сетевого перемещения данных* (англ. *Network Data Moving Protocol — NDMP*).

Пример аварийного плана

Выдержка *"Состав структурных единиц"* из опубликованного варианта аварийного плана Массачусетского технологического института.

Комитет по регламенту административного расчета

- ❑ Старший Вице-президент института — председатель комитета. Управляет и руководит работами по восстановлению. Обеспечивает взаимодействие с высшим руководством научного института для отчетности о ходе выполнения восстановительных работ.
- ❑ Вице-президент по финансовым операциям. Обеспечивает взаимодействие с комитетом для поддержки критических бизнес-функций, пораженных катастрофой.
- ❑ Вице-президент по информационным системам. Координирует всю обработку данных и восстановление телекоммуникационных систем, включая восстанавливаемые здания и операции в выделенном резервном расположении.
- ❑ Вице-президент по исследованиям. Обеспечивает взаимодействие с комитетом для поддержки критических бизнес-функций, пораженных катастрофой.
- ❑ Вице-президент по развитию ресурсов. Обеспечивает взаимодействие с комитетом для поддержки критических бизнес-функций, пораженных катастрофой.
- ❑ Вице-президент по ассоциации выпускников. Обеспечивает взаимодействие с комитетом для поддержки критических бизнес-функций, пораженных катастрофой.
- ❑ Проректор. Обеспечивает взаимодействие с комитетом для поддержки критических бизнес-функций, пораженных катастрофой.

Команда управления продолжением бизнес-деятельности

- ❑ Специалист по информационной безопасности. Обеспечивает взаимодействие между командами операций и управления института и внешними командами в областях, пораженных катастрофой. Также отвечает за постоянное поддержание аварийного плана (АП), обучение навыкам АП и тестирование аварийного плана АП института. Координирует институтскую команду поддержки систем по вопросам, касающимся команды управления продолжением бизнес-деятельности (КУПБ).

- ❑ Директор по операциям и системам. Координирует поддержку для ресурсов обработки данных в главном центре данных и на выделенных местах временного расположения.
- ❑ Директор по телекоммуникационным системам. Обеспечивает альтернативные каналы передачи голоса и данных в случае, если основные линии и оборудование повреждены катастрофой. Оценивает требования и выбирает соответствующие средства для резервирования сети научного института.
- ❑ Начальник полиции академгородка. Обеспечивает физическую безопасность и дополнительную поддержку для пораженных областей и механизм оповещения для проблем, которые являются или могут быть катастрофами. Распространяет периметр безопасности вокруг областей, пораженных катастрофой.
- ❑ Директор коммунальных служб. Координирует все службы для восстановления водопровода, электричества и других систем поддержки, а также структурную целостность. Оценивает повреждения и осуществляет прогноз для возможного использования структур, поврежденных катастрофой.
- ❑ Директор по страхованию и юридическим аспектам. Обеспечивает взаимодействие специалистов по страхованию и искам. Координирует программу страхования с программами планирования продолжения бизнеса.
- ❑ Директор службы новостей. Обеспечивает общение с прессой, персоналом, общественными организациями, не задействованными в восстановительных операциях.
- ❑ Департамент по работе с персоналом. Обеспечивает поддержку элементов восстановления, связанных с людскими ресурсами и оповещение персонала посредством аварийной радиосвязи.
- ❑ Директор распределенных вычислительных и сетевых служб. Обеспечивает сетевую поддержку для административных и научных работ и других распределенных служб и сетей.
- ❑ Помощник вице-президента по информационным системам. Представляет службу Президента, обеспечивает взаимодействие с внешними командами.
- ❑ Инспектор бухгалтерии. Представляет Вице-президента по финансовым операциям, осуществляет взаимодействие со службой финансовых операций внешних команд.
- ❑ Руководитель подразделения аудита. Обеспечивает аудит в процессе аварии и восстановления. Вырабатывает рекомендации по изменениям в обычных процедурах контроля, необходимые для процесса восстановления.

- ❑ Служба надежности. Координирует с КУПБ работы по снижению риска, эвакуации и аварийного реагирования.
- ❑ Команда аварийного реагирования. Возглавляется Управляющим инженерными коммунальными службами и обеспечивает первый этап реагирования на аварийные ситуации.

Классификация технологий RAID



В табл. 31.1 приводится краткое описание RAID-технологий по источникам [Kurtz] и [Harris].

Таблица 31.1 Уровни технологии RAID

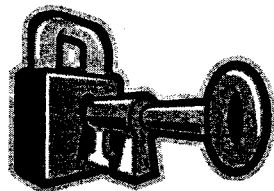
Уровень RAID	Описание	Краткое обозначение
0	Данные распределяются по нескольким дискам, функционирующим как один. Предназначен скорее для повышения производительности операций записи\чтения, так как реально не является отказоустойчивым. При сбое одного из дисков весь набор данных считается потерянным	Striping
1	Данные копируются с одного диска на другой диск (или несколько других дисков) в режиме реального времени. Таким образом, каждый из дисков является зеркальным отображением другого, при выходе из строя одного диска система начинает работать с данными, размещенными на другом диске. Считается одним из самых дорогих решений, так как требует дополнительного аппаратного обеспечения	Mirroring (duplexing)*
2	Представляет скорее теоретический уровень, потому что не используется на практике. Заключается в распределении данных по 32 дискам с использованием еще 7 дисков для данных с коррекций ошибок	Hamming code parity
3	Данные распределяются по нескольким дискам на уровне байтов данных. Один дополнительный диск используется для сохранения контрольных значений данных для корректировки ошибок	Byte-level parity
4	Данные распределяются по нескольким дискам на уровне дисковых секторов (блоков). Один дополнительный диск используется для сохранения контрольных значений данных для корректировки ошибок	Block-level parity

Таблица 31.1 (окончание)

Уровень RAID	Описание	Краткое обозначение
5	Данные распределяются по нескольким дискам на уровне дисковых секторов (блоков). Для сохранения контрольных значений данных для корректировки ошибок используется любой свободный в данный момент диск. Таким образом, удается избежать опасности сбоя контрольного диска, который может произойти в двух предыдущих вариантах Наиболее популярная в настоящее время технология, позволяется использовать метод <i>hot swap</i>, т. е. замену диска во время работы системы без перезагрузки	Interleave parity
6 или 15 (1+5)	Схож с предыдущим уровнем, однако добавлены дополнительные диски для дублирования (зеркалирования) контрольных данных	Second parity data
7	Схож с уровнем 5, но при этом добавлено специальное программное обеспечение, позволяющее работать с массивом дисков как с одним виртуальным диском	Single virtual disk
10 (1+0)	Распределение данных (по уровню 0) между парами зеркалированных дисков (по уровню 1)	1+0

* Иногда можно встретить дополнительное разделение этого уровня на два подвида. В этом случае *зеркалированием* (англ. *mirroring*) называется вариант использования двух дисков с одним контроллером (собственно записывающим устройством). Вариант использования двух разных контроллеров для двух дисков называется *дуплексированием* (англ. *duplexing*).

Глава 32



Реагирование на нарушения информационной безопасности

Вопросам реакции на нарушения безопасности посвящен отдельный документ информатория сообщества Интернет — RFC 2350 [RFC-2350]. На наш взгляд, документ ориентирован на реагирование на инциденты информационной безопасности, как на предмет предоставления (услуг возможно, коммерческих), хотя вопросы реагирования внутренних команд предприятия также рассмотрены. В данной главе мы постараемся рассказать о реакции на инциденты, ориентируясь на указанный документ, а также используя собственные идеи и опыт.

Основы функционирования группы реагирования



Каждая *группа реагирования* (ГР) должна принадлежать определенному сообществу (в нашем случае предприятию). Ей следует определить и сделать доступными для пользователей сферу своих действий, правила работы, а также проинформировать сообщество, что ГР ожидает от пользователей. На наш взгляд, ГР предприятия, которая может находиться в структуре службы информационной безопасности предприятия или быть тесно с ней связана, не должна ни по каким причинам отвергать сообщения пользователей по инцидентам, связанным с безопасностью. Например, если дежурная служба ГР информационной безопасности получила сообщение о нарушениях в области физической безопасности (скажем, охрана здания) или экономической безопасности (махинации в бухгалтерии), то она ни в коем случае не должна отказывать пользователю в рассмотрении этого вопроса или перенаправлять его в другую службу. Задача представителя ГР проинформировать службу, в чьей компетенции находится данный вопрос, предоставить имеющуюся по вопросу информацию, а также способы контакта с сотрудником, передавшим данную информацию.

Что касается требований к пользователям, особенно относительно формы доклада о нарушении, то они должны носить скорее рекомендательный характер. Безусловно, ГР будет гораздо легче работать, если на самом раннем

этапе пользователь соберет достаточное количество информации по инциденту, который затем предоставит ГР. Однако, если речь идет о внутренней работе предприятия, то следует учесть, что обычно пользователь обладает необходимой квалификацией лишь для того, чтобы выполнять свои служебные функции, а для вопросов безопасности как раз и создается соответствующая служба, куда и взяты на работу члены ГР. Поэтому чаще всего, в таком случае пользователь может, лишь инициировать работу по инциденту, а также помочь сотруднику ГР подготовить форму сообщения об инциденте, сообщая имеющиеся сведения. Тем не менее, если служба безопасности проводит регулярное обучение пользователей или семинары по информационной безопасности, необходимо подчеркивать важность участия пользователей в выявлении инцидентов, а также вырабатывать у них навыки соответствующего распознавания и идентификации нарушения информационной безопасности. Например, если пользователь сможет отличить приглашение к вводу пароля реальной системы от ее имитации программой-перехватчиком паролей (и проинформировать об этом), то это будет существенной заслугой службы безопасности. Как минимум пользователь должен знать, что есть служба, куда он может и должен сообщить об инциденте, а также знать способы контакта с этой службой.

Еще один вариант: внутренняя ГР предприятия планирует использовать некоторую внешнюю ГР для сотрудничества. В этом случае все формы доклада для внешней ГР необходимо подготовить в соответствии с требованиями.

Правила работы ГР должны быть известны пользователям, т. е. доступны для свободного прочтения. Если используемые на предприятии информационные системы функционируют таким образом, что методы (именно методы, а не данные, пароли, коды доступа и т. д.) обеспечения их безопасности должны содержаться в секрете, то это скорее всего не есть показатель хорошей защищенности. Тем не менее в таких случаях можно скрыть от пользователей технические детали реагирования на инциденты. Однако общие аспекты, особенно та часть, где могут быть затронуты права пользователей в рамках закона или морально-этические аспекты, должны быть опубликованы. В обязательном порядке должны быть опубликованы документы, касающиеся средств обнаружения, идентификации, снижения риска ущерба и т. п. по инцидентам в рамках средств, доступных самому пользователю, например, рекомендации по работе в Интернете, с электронной почтой и др.

Взаимоотношения всевозможных ГР, если речь идет о различных предприятиях, определяются взаимоотношениями между службами безопасности предприятий. Если в рамках одного предприятия существует несколько ГР, как в приведенном примере, скажем ГР информационной безопасности, ГР охраны здания, ГР экономической безопасности, то отношения между ними должны быть также определены. Следует обратить внимание на то, что между самими подразделениями различных сфер безопасности отношения могут быть достаточно формальными, требующими, например, письменного

обращения и согласования вопросов. В этом случае для ГР следует предусмотреть более ускоренный вариант логических коммуникаций.

Что касается коммуникаций в смысле обмена информацией между ГР, внутри ГР и ГР с пользователями, то все они должны быть защищены как любые каналы связи с информацией повышенной критичности и для них следует определить уровни конфиденциальности, целостности, апеллируемости и т. д. Если инцидент нарушения безопасности распределен по времени, то, с одной стороны, бывает важно не допустить несвоевременного разглашения деталей инцидента и реакции на него, значит, необходимо обеспечить конфиденциальность такой информации. С другой стороны, пользователь, получая информацию от ГР, должен быть уверен, что общается именно с ГР и информация поступает не в искаженном виде, следовательно, необходимо обеспечить целостность данных коммуникаций.

Современное программное обеспечение серверов электронной почты может обеспечить конфиденциальность, целостность и неотрекаемость обмена электронными письмами. Если ГР хочет использовать альтернативные каналы связи, то необходимо позаботиться об инфраструктуре управления криптографическими ключами — *PKI (Public Key Infrastructure)*.

Нормативные акты



Теперь рассмотрим смысловое наполнение документов — правил, инструкций, рекомендаций — на основе которых работает ГР. Основные рекомендуемые разделы этих документов следующие:

- ❑ **Информация о самом документе.** Поскольку развитие информационной безопасности — это динамический процесс, а документ может изменяться, дополняться и т. д., соответственно, для отслеживания текущей версии необходимо иметь дату последнего изменения, номер версии и другие данные, позволяющие использовать наиболее свежий документ. Если обновленные версии документа доставляются до пользователей после модификации, то должен быть указан список рассылки. Возможно, в рамках предприятия это не всегда целесообразно делать; скорее если какое-либо изменение носит глобальный характер и требует осведомленности пользователей, то следует производить разовую рассылку по всем пользователям информационного пространства предприятия информационных бюллетеней и т. п. При этом, если изменения затрагивают какие-либо документы, необходимо указывать ссылку на место хранения экземпляра полной версии самого документа.
- ❑ **Контактная информация ГР.** Здесь необходимо указать официальное название ГР как подразделения организации; адрес (для организации — место расположения, этаж, комната); часовой пояс — может быть актуален, пожалуй, для географически распределенного предприятия; номера теле-

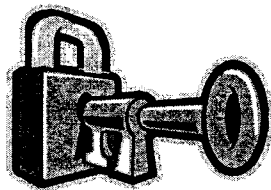
фонов, факса, адреса электронной почты и другие способы связи; алгоритмы или средства шифрования и открытые ключи; список членов группы, возможно, с указанием функциональной специфики должности; часы работы (для предприятия — рабочий день и время дежурных смен, если таковые есть); дополнительная контактная информация и специфические контакты (скажем, сообщения для отдельных категорий инцидентов принимаются в виде заполнения формы на Веб-сайте).

- ❑ *Информация о статусе группы*, т. н. *устав*. В него входит описание видов деятельности ГР, цели и задачи (для организации — работа с пользователями, оповещение и т. д.); клиенты или пользователи, обслуживаемые данной ГР; вышестоящие органы, которые наделяют ГР полномочиями и к которым можно апеллировать в случае недоразумений; полномочия ГР (например, очень важно определить, имеет ли право ГР прекратить временно деятельность данного пользователя, если его активность представляет собой угрозу информационной безопасности и рассматривается как инцидент).
- ❑ *Политики работы ГР*. Перечисляются типы инцидентов и формы реагирования на них. Уже упоминалось, что ГР предприятия реально необходимо реагировать на все сообщения пользователей и оказывать им поддержку. Данный раздел необходим в нашем случае скорее для защиты самой ГР от подobia "ложных вызовов", когда истеричный пользователь может требовать немедленного прибытия ГР по причине того, что сам пользователь уничтожил один из системный файлов. Необходимо указать политику взаимодействия с другими ГР и подразделениями (когда в случае инцидента потребуется оперативная поддержка). Указывается порядок раскрытия информации о деятельности группы и об инцидентах (это актуально, когда руководитель подразделения, где произошел инцидент, пытается вмешаться в ход работы ГР). В [RFC2350] приведены примеры различных видов информации, которая будет или не будет раскрыта в работе ГР, а если будет раскрыта — то кому. Информация представляет собой:
 - персональные данные пользователей;
 - информация о злоумышленнике;
 - техническая информация о конкретной информационной системе;
 - информация об уязвимостях системы;
 - информация, которая косвенно может повредить организации (например, то, что нарушение имело место, может характеризоваться как незащищенность организации);
 - статистическая информация о нарушениях;
 - контактная информация администраторов и ГР.

Кому может потребоваться информация:

- другим ГР;
 - поставщикам информационных систем;
 - правоохранительным органам;
 - прессе;
 - руководству;
 - исследовательским центрам и т. п.
- ❑ *Средства коммуникации.* Необходимо указать, какие способы коммуникации и для какой информации приемлемы при общении с ГР. Скажем, если сообщение об инциденте как о факте можно принять по открытому телефону, то детали инцидента (фрагмент регистрационного журнала) должны высылаться только при условии соблюдения конфиденциальности / целостности / апеллируемости.
- ❑ *Действия ГР.* Классификация нарушений (реальность и масштаб), реагирование (выяснение причины, информирование заинтересованных/затронутых субъектов), решение проблемы (устранение уязвимостей, ликвидация последствий, оценка дополнительных действий, сбор улик и доказательств), профилактика (контакты с внешними специалистами, хранение инструментов безопасности и программных заплат для систем, информационные сообщения и обучение, регулярный или выборочный аудит безопасности информационных ресурсов, ведение статистического учета нарушений).
- ❑ *Форма доклада об инциденте.* Структурированный набор информации, предназначенный для того, чтобы ГР начала квалифицированные действия по реагированию на инцидент.
- ❑ *Отводы.* Обычно здесь имеются ввиду сведения о снятии ответственности за то, что при использовании информации о работе ГР будет нанесен некий ущерб. Однако в случае предприятия группа реагирования скорее всего не может иметь столь отстраненный характер. Как составная часть организации она должна принимать участие в работе с инцидентом от начала до конца и нести ответственность за свои действия.

Глава 33



Проведение расследования

Проведение расследования может быть функцией группы реагирования или может выполняться другими специалистами — это зависит от политики предприятия и, вероятно, от вида конкретного инцидента. Для инцидентов, достаточно распределенных по времени, расследование возможно проводить уже в ходе осуществления атаки или другого злонамеренного воздействия. Основная цель расследования — собрать как можно больше данных, информации касательно данного инцидента, т. е. ответить на вопросы: кто, что, когда, где, как и почему. Далее мы постараемся указать основные направления и проблемы, с которыми может столкнуться расследование, а также предложим выдержки из методики расследования специалиста по безопасности *Thomas Welch* [HISM1].

Предварительные меры



На первом этапе следует определиться, что является *инцидентом, требующим расследования*. В *главе 32* основным ориентиром были инциденты, о которых сообщают пользователи, но ведь значительная часть нарушений может быть выявлена службой безопасности без помощи пользователей. Поэтому необходимо документально закрепить, наступление каких событий требует проведения расследования. Наиболее простой пример — обнаружение факта многократных попыток ввода пароля пользователем с отрицательным результатом. Что это — попытка подбора пароля злоумышленником или просто пользователь забыл свой пароль? Требуется в этом случае расследование или нет?

Вот список некоторых действий, которые потребуют от специалистов по безопасности определенного внимания по той причине, что могут являться преступлениями по общегосударственным законам, а значит, предприятие как субъект государства должно как-то отреагировать на такие события.

- ☐ Несанкционированный доступ.
- ☐ Превышение прав санкционированного доступа.
- ☐ Нарушение авторского права или интеллектуальной собственности.
- ☐ Злоупотребление полученной информацией.

- ☐ Использование порнографии, призывов к насилию, разжиганию национальной, религиозной и другой вражды.
- ☐ Несанкционированное использование ресурсов (сервисов, услуг и т. д.).
- ☐ Подделка данных, несанкционированная модификация.
- ☐ Кража собственности (в нашем случае информационного оборудования или комплектующих).
- ☐ Нарушение прав частной собственности, конфиденциальности информации.
- ☐ Вывод из строя или помеха в работе сервисов (отказ в обслуживании).
- ☐ Мошенничество с помощью компьютеров.
- ☐ Распространение вирусов.
- ☐ Злонамеренное искажение или уничтожение данных.
- ☐ Вымогательство или шантаж с использованием компьютеров.
- ☐ Присвоение чужого имущества.
- ☐ Шпионаж (в том числе промышленный).
- ☐ Терроризм (в нашем случае информационный или с помощью компьютеров).

Организация должна определить, какие признаки могут являться идентификаторами возможности наступления того или иного события, которое может быть отнесено к преступлению.

Другим важным моментом, на который необходимо обратить внимание, является осведомленность персонала предприятия о том, что их рабочая деятельность есть предмет возможного расследования. В противном случае, если в ходе расследования возникнет необходимость, например, анализировать личные файлы и данные работника, которые он хранит на своей рабочей станции или сервере, то это может послужить причиной иска со стороны работника к предприятию.

Кроме того, необходим ряд дополнительных предварительных мероприятий. Изучая западный опыт борьбы с компьютерными преступлениями, можно обнаружить, например, варианты отказа суда в рассмотрении исков по нарушениям информационной безопасности по следующим причинам:

- ☐ нарушитель не был предварительно оповещен, что неавторизованный доступ запрещен;
- ☐ персонал не был ознакомлен с правилами безопасности организации либо неадекватно понял их;
- ☐ организация не уделяла внимания информационной безопасности, и таким образом, спровоцировала нарушение.

Определение целей и задач расследования



Следующее, с чем должна определиться организация, — это для чего проводится данное расследование, каков будет его результат. В зависимости от конкретного инцидента расследование может быть основой:

- ☐ для внутреннего решения самого предприятия (штраф, увольнение);
- ☐ запроса на возмещение в страховую компанию;
- ☐ административной ответственности;
- ☐ уголовного преследования.

В зависимости от этого и должно различаться проведение расследования и особенно оформляться его результаты. Если для внутренних решений организации может быть достаточно рапорта начальника службы безопасности о результатах расследования, то для уголовного преследования необходимо будет представить должным образом оформленные улики, свидетельства и доказательства. Если заранее неизвестно о возможных решениях (как это чаще всего бывает), то проводить расследование следует изначально со всей возможной тщательностью, особенно в оформлении результатов.

Дело может быть дополнительно осложнено особенностями законодательства — не все виды преступлений могут быть отражены в законах, а также спецификой работы с компьютерами как с областью высоких технологий, требующих соответствующей подготовки и квалификации проверяющих лиц. Это касается обнаружения улик, их сбора, хранения, транспортировки и представления арбитражу. Поэтому перед началом расследования специалисту по информационной безопасности (который, будем считать, уже обладает требуемыми знаниями о технических особенностях работы с уликами) необходимо изучить особенности законодательства, чтобы обеспечить процессуальную корректность работы с уликами, для обеспечения их силы как доказательств.

Не вдаваясь в юридические аспекты предоставления доказательств, уточним, что для сторонней оценки доказательств (скажем, в суде) информационные улики (нули и единицы данных или в более общем случае намагниченные и размагниченные участки носителя) могут быть представлены в одном из следующих видов:

- ☐ как изображение на экране монитора;
- ☐ как распечатка;
- ☐ как видео- или фотосъемка;
- ☐ как устное или документальное заявление.

При проведении расследования следует учитывать эти способы представления доказательств. Так, например, если на месте преступления была зафик-

сирована некая улика в виде изображения на мониторе, которое затем (например, после случайной перезагрузки) было утеряно, то необходимо представить его в виде фотографии или документального описания свидетелей.

Кроме того, для информационных технологий не всегда возможно предоставить оригинал улики. Следует учитывать, что основой для предоставления копии могут быть случаи, когда:

- ❑ оригинал уничтожен природным явлением или другим случайным действием (ошибкой персонала);
- ❑ оригинал уничтожен в ходе обычного производственного процесса (одни данные записаны поверх других);
- ❑ оригинал находится в распоряжении третьей стороны, которая не может быть принуждена предоставить его арбитражу.

При сборе улик в виде физических предметов необходимо обеспечить их идентификацию (метку данного образца улики, дату и время изъятия, имя изымающего, место изъятия), сохранность (если на улике могут иметься отпечатки пальцев, изъятие следует производить в перчатках), неизменяемость (сохранять в опечатанном контейнере).

При хранении и транспортировке такие предметы должны быть защищены от воздействий жары, холода, воды, механических повреждений, вибрации, магнитных полей, статических разрядов и т. д. После использования физических предметов в качестве улик, следует предусмотреть процедуру их возврата (если это возможно) в бизнес-процесс.

Оперативные действия



Теперь, когда мы теоретически подготовлены к проведению расследования, возьмемся за сам процесс.

Сначала следует определиться, совершен ли уже инцидент или он находится в стадии совершения, и какие меры необходимо предпринять для возможного снижения ущерба от инцидента (быть может, это часть аварийного плана). Далее — каким образом был обнаружен инцидент — персоналом или автоматизированной системой, где источник события — на предприятии или удаленный. Необходимо предварительно определиться, на каком этапе будет оповещено руководство — сразу после поступления информации об инциденте (в этом случае возможно большое число ложных сообщений или сообщений о незначительных событиях) либо после некоторого предварительного расследования, оценки последствий и ущерба, если таковой имеется.

На следующем этапе желательно решить, будет ли информация об инциденте разглашена и, если да, то какая часть и кому. С одной стороны, разглашение информации может быть вредно, так как, во-первых, может инфор-

мировать злоумышленника, а, во-вторых, может бросить тень на компанию. С другой стороны, разглашение может осведомить пользователей о признаках атаки и предотвратить дальнейшее развитие инцидента, а также привлечь дополнительное число свидетелей. Для каждого инцидента необходим свой, возможно, уникальный подход.

Следующий шаг — решить, ограничиться ли результатами предварительного расследования (устранив соответствующие уязвимости систем, скорректировав аварийный план и т. д.) или проводить дальнейшее развернутое расследование (например, для сбора улик с целью уголовного преследования). Для этого необходимо ответить на следующие вопросы:

- ☐ не повлияет ли расследование на бизнес-процесс;
- ☐ есть ли необходимость отключать системы, которые затронул инцидент;
- ☐ будут ли компоненты систем изъяты как улики;
- ☐ будут ли раскрываться данные для внутреннего использования предприятия;
- ☐ не будут ли возложены на предприятие ряд обвинений со стороны суда;
- ☐ не повредит ли это репутации организации;
- ☐ приведет ли разглашение к привлечению новых злоумышленников или послужит предотвращению инцидентов;
- ☐ допустимы ли затраты времени: на опрос свидетелей, на привлечение специалистов по информационным технологиям, поиск и предоставление необходимых документов и т. д.

Необходимо также решить, кто будет осуществлять расследование, ведь специалисты различных источников имеют свои преимущества и недостатки. Собственные специалисты предприятия лучше других знают внутреннюю структуру, в том числе информационные системы, но могут быть малокомпетентны в юридических и криминальных аспектах. Представители правоохранительных органов придают расследованию большую законность, но обычно недостаточно квалифицированы в области информационных технологий. Коммерческие структуры, предоставляющие специалистов (частные детективы), предлагают дорогостоящие услуги. Ко всему прочему следует учесть, что привлечение внешних специалистов, особенно из правоохранительных органов, возможно, придется скрывать, чтобы не информировать злоумышленника.

При проведении расследования, кроме основного специалиста по расследованию, потребуются участие администратора информационной системы или экспертов по ней, юриста, понадобятся инструкции и руководства по работе с системой, особенно руководства по безопасности. Если привлеченным лицам необходимо посещение места физического расположения системы, то следует обеспечить их соответствующий допуск.

Наблюдение за злоумышленником и сбор улик



Удаленное наблюдение за злоумышленником

Очень важным и сложным этапом проведения расследования является сбор улик и возможное наблюдение за злоумышленником. В случае информационных систем, кроме обычного физического наблюдения (посредством видеокамер, слежки, а также контрол активности — например, банковских расходов, телефонных переговоров и т. д.), будет необходимо и компьютерное наблюдение. В него могут войти активные средства (мониторинг деятельности в режиме реального времени) или пассивные средства (аудит регистрационных журналов).

Если будет производиться копирование информации из информационной системы, следует обеспечить наличие необходимого числа носителей. Возможно, потребуются учесть юридические аспекты оформления процедуры, например, получение ордера на обыск. Если анализ технических компонентов системы будет производиться не в экспертной лаборатории, а на месте, то необходимо запастись соответствующим оборудованием. Если есть предположение, что злоумышленник продолжает работать в системе, то расследование (во всяком случае сбор улик) придется производить скрытно.

Дополнительными источниками информации для расследования будут остальные связанные с деятельностью субъекта факты — персональные данные и история работы, если это работник предприятия, биллинговые данные по прошлым телефонным переговорам, регистрационные журналы системы доступа в здание, информация о личных контактах, просто слухи.

Для наблюдения за деятельностью удаленного злоумышленника возможно использование технологии заманивания. В этом случае может быть создана *фиктивная информационная система* (англ. *Honey pot*), которая обеспечивается достаточными средствами безопасности, чтобы полностью контролировать активность субъекта в системе, и которой придается вид привлекательного для злоумышленника объекта (хранилище паролей, номеров кредитных карт, система денежных расчетов, сервер удаленного доступа и т. д.). Тогда все действия злоумышленника могут управляться и фиксироваться без причинения реального вреда предприятию, позволяя в то же время собирать материал для дальнейшего расследования или судебного преследования.

Принципы сбора улик о компьютерных инцидентах

Рассмотрим действия на месте инцидента или преступления, предназначенные для сбора улик и доказательств.

- ❑ Во-первых, необходимо обеспечить безопасность самого места работы в том смысле, что обеспечить бесперебойность электропитания, невмеша-

тельство в работу информационного обеспечения, телекоммуникаций и т. д. Возможно, все действия будут производиться после обычного рабочего времени, это также следует учесть, т. е. согласовать график работы служб эксплуатации здания/ сооружения.

- ❑ Если сам подозреваемый находится поблизости, необходимо удалить его либо обеспечить условия, когда он не может вмешаться в ход расследования. При этом надо учесть возможность удаленного доступа самого злоумышленника или его сообщников.
- ❑ При входе в помещение необходимо вести себя осторожно, чтобы не нарушить работу системы, которая может сама по себе являться уликой. Не следует трогать клавиатуры, кабели и т. д.
- ❑ Необходимо проанализировать и описать работающую технику, произвести фотографирование и/или видеосъемку (разместите на фотографиях указатель масштаба). Фотографирование следует производить без вспышки, а съемку изображения с мониторов производить (при наличии соответствующих средств) напрямую в камеру, а не с экрана дисплея.
- ❑ Если возникла необходимость отключить какое-либо работающее устройство, то целесообразно делать это не через стандартные процедуры отключения (которые могут быть подменены злоумышленником), а путем извлечения кабеля электропитания из розетки. Это впрочем не касается распределенных систем, скажем, мэйнфреймов, которые должны выключаться в установленном порядке.
- ❑ Перед отключением необходимо позаботиться о данных, которые хранятся в буферах обмена, оперативной памяти и могут быть потеряны при выключении. По вопросам извлечения информации из энергозависимой памяти следует обратиться к документации на оборудование или поставщикам.
- ❑ При работе с данными на системах, подверженных инциденту, не следует пользоваться средствами операционной системы или системных утилит, так как они могут быть подменены злоумышленником и привести к уничтожению данных, в том числе улик.
- ❑ Следует обратить внимание на все имеющиеся поблизости от места инцидента (и места расположения злоумышленника) альтернативные хранители информации. В этот список должны войти не только обычные носители — дискеты, компакт- и минидиски, ленты, распечатки, но и черновики, стандартная документация с пометками, ленты с матричных принтеров, ленты автоответчиков, электронные ежедневники и календари и т. д. Придется обратить внимание также и на мусорные ведра, при этом не стоит пренебрегать носителями, которые были подвергнуты попытке разрушения или уничтожения — опытные эксперты смогут восстановить данные во многих случаях, на первый взгляд кажущихся необратимыми.

Примечание

Известен случай, когда с дискеты (1,44 Мбайта) было восстановлено более 10 Мбайт информации с учетом перезаписей и удаления файлов.

Методика сохранения данных в затронутых системах

Теперь обратимся к рекомендациям по сохранению данных в системах, подверженных инциденту.

Создать копию образа диска системы

- ☐ Извлечь жесткий диск, при этом уточнить.
 - Какой диск снят (на шине — C, D).
 - Какой тип диска (IDE, SCSI).
 - Каков объем диска (цилиндры, головки и сектора).
- ☐ Поместить каждый диск на отдельную, специально подготовленную, исследовательскую машину следующим диском, учитывая, что на подозрительном диске могут содержаться вирусы (то есть необходимо на основном диске иметь минимум необходимого программного обеспечения и вести регистрацию работы приложений).
- ☐ Сделать копию образа диска на доступные носители. Потребуется минимум четыре копии.
 - Одну вместе с исходным диском запечатать и оформить как улику.
 - Одну вернуть в систему.
 - Две использовать при расследовании.
- ☐ Восстановить одну из рабочих копий на диск, равнозначный подозрительному, на контролируемом исследовательском оборудовании.
- ☐ Просмотреть все данные, имеющиеся на подозрительном диске.

Систематически проанализировать данные на диске

Каждый элемент системы должен быть проверен.

- ☐ Если файлы данных зашифрованы, необходимо попытаться получить ключ.
 - Выяснить используемый алгоритм и принцип формирования ключа.
 - Если ключ формируется на основе пароля — попытаться найти пароль (возможно, проанализировав другие улики).
 - Попытаться использовать средства взлома пароля.
 - Если ничего не помогло, попытаться получить пароль в ходе допроса или разговора с подозреваемым.

- ☐ Если диск отформатирован, следует использовать unformat — процедуры или другие средства восстановления данных.
- ☐ Проверить на наличие вирусов и, если возможно, устранить их.
- ☐ Составить схему размещения файлов на диске, при этом использовать утилиты только с исследовательского диска.
- ☐ При этом учесть скрытые и удаленные файлы.
- ☐ Снять атрибуты сокрытия и восстановить удаленные файлы.
- ☐ Проверить дисковое пространство от конца файла до конца кластера.
- ☐ Проинвентаризировать все файлы по категориям — doc, txt, xls, dbf и т. д., учитывая, что могут быть попытки сокрытия файлов путем изменения расширения.
- ☐ Проанализировать данные, хранимые в приложениях, особенно в коммуникационных программах.
 - Настройки программы.
 - Идентификаторы абонентов.
 - Регистрационные журналы сканеров телефонных номеров.
- ☐ Проанализировать оставшееся свободное пространство на диске.

Восстановить подозрительный диск на "родной" системе

- ☐ Переустановить копию диска на системе, откуда он был изъят.
- ☐ Проверить настройки BIOS — возможно загрузка системы начиналась с дискеты.
- ☐ Если загрузка защищена паролем, его возможно, обойти, удалив временно батарею и замкнув специальный ключ-контакт на материнской плате.
- ☐ Загрузиться с чистой копии операционной системы (с дискеты), обращая внимание на загрузочные процессы, которые могут быть также модифицированы (модификация BIOS, self-test и т. д.).
- ☐ Перед началом работы необходимо проверить системное время.
- ☐ Составить полный аналитический отчет по работе системы.
 - Базовая системная конфигурация.
 - Краткое описание диска.
 - Описание загрузки памяти со списком задач.
 - Описание дисплея.
 - Описание принтера.
 - Описание резидентных программ.

- Описание драйверов.
 - Системные прерывания.
 - Описание CMOS.
 - Описание системных переменных из конфигурационных файлов (для сложных систем, видимо, более актуальна копия конфигурации, скажем, реестра для Windows-систем).
- ☐ Проверить регистрационный журнал системы.
- Распечатать журнал, если возможно.
 - Проверить соответствие журнала политике аудита.
 - Проверить целостность журнала.
 - Убедиться, что журнал не был подменен, а если был — то когда.

Восстановить систему в исходной конфигурации

Использовать операционную систему подозрительного диска.

- ☐ Проверить все программы, запущенные системой.
- ☐ Проанализировать их влияние на систему.
- ☐ Проверить наличие троянских программ и других зловредных программ.
- ☐ Проверить системные прерывания и резидентные программы (скажем, перехватчики набора клавиатуры).

Восстановить данные и проанализировать их

Естественно, восстановление данных не должно повлечь за собой их изменение, которое могло бы сказаться на результатах расследования. Это означает, что место для восстановления данных должно быть проверено перед началом процесса восстановления. Кроме того, при анализе данных стоит фиксировать полученные выводы, а также предположения и догадки. В противном случае можно потерять или потратить много времени на восстановление аналитической информации, производной от анализируемых данных. Зафиксированный ход рассуждений при анализе также может служить основой для начала анализа при следующем расследовании.

Дополнительные замечания

- ☐ При работе с оборудованием и комплектующими использовать антистатические материалы и заземлители.
- ☐ Фиксировать каждое сделанное действие (восстановление файла, изменение конфигурации и т. д.).
- ☐ Учитывать и анализировать при работе вероятность как ошибки, так и злого умысла.

- ☐ Помнить последовательность записи байтов при выполнении копирования памяти (дампа) системы.
- ☐ Использовать дискеты с защитой от записи.
- ☐ При анализе баз данных учитывать структуру данных, сравнивать с исходной структурой, так как измененная, она может повредить данные.
- ☐ Учитывать, что на внешних носителях могут быть не только копии данных, но и те данные, которых нет на жестком диске.
- ☐ При подборе пароля:
 - искать его записанным на бумаге;
 - пытаться использовать слова, связанные с подозреваемым (имена, номера и т. д.);
 - обратиться к поставщикам (возможны мастер-ключи или пароли по умолчанию);
 - использовать программы взлома паролей;
 - в случае неудачи пытаться использовать атаку по словарю или прямым перебором.
- ☐ Следует помнить о возможностях *стеганографии* — технологии сокрытия данных в изображениях, музыкальных или видеоклипах и т. п.

Анализ результатов расследования



Результатом расследования, возможно, будут замечания к несовершенству каких-либо средств или мероприятий по безопасности. Возможные изменения могут коснуться:

- ☐ аварийного плана (раздел реагирования на инциденты);
- ☐ политики распространения информации;
- ☐ порядка сообщения и отчетов по инцидентам;
- ☐ политики электронного мониторинга и аудита (особенно в части предварительного оповещения и ознакомления персонала);
- ☐ предупреждений на входе в информационную систему (о запрете несанкционированного использования и о возможности мониторинга);
- ☐ политики предварительной проверки персонала и консультантов, принимаемых на работу.

Вместо заключения: идеальная служба информационной безопасности

В качестве некоторой шпаргалки, или checklist, суммирующей изложенную в данной книге информацию, авторы хотели бы предложить небольшой список того, что должно быть реализовано в организации при условии, что служба информационной безопасности работает в некоторых идеальных условиях (неограниченный бюджет, достаточное количество квалифицированных специалистов и т. п.). Естественно, что в жизни не удастся реализовать эту (равно как и любую другую идеальную) программу полностью. Однако, во-первых, она обозначает некоторую цель, к которой следует стремиться, а во-вторых, многие из обозначенных пунктов можно реализовать в усеченном варианте (специфика усечения определяется отдельно для каждой конкретной организации).

Следует также обратить внимание на то, что одни работы носят одноразовый характер по типу "сделали и забыли". Вторые — это разовое выполнение с последующим минимальным контролем соответствия текущего состояния некоторым условиям. Третьи — периодические работы по истечении промежутка времени или наступлению какого-либо события. И четвертые — рутинные постоянные работы типа мониторинга.

1. Вся работа службы информационной безопасности (как впрочем и всего предприятия) организована в строгом соответствии с законодательством государства.
2. Все кандидаты для приема на работу проходят собеседование и проверку службой информационной безопасности. Вновь принятые работники и сторонние специалисты по контракту проходят обучение основам безопасности, ознакомление с нормативными документами по безопасности, тестирование на уровень квалификации для работы с информационными системами и подтверждают подписью свое обязательство следовать нормативам организации (в том числе и по безопасности). Увольняющиеся работники проходят собеседование с представителем службы безопасности, увольнение работника сразу отражается в информационных системах (блокирование и удаление учетных записей, изъятие карт

- доступа и т. д.). Сотрудники организации проходят периодические семинары и обучение по информационной безопасности.
3. Ведется анализ моральной и психологической обстановки в организации, учет нарушений безопасности конкретными сотрудниками для выявления возможных тенденций. Поощряется тесный информационный контакт пользователей со службой безопасности. Выборочно или постоянно анализируется электронная почта сотрудников с соответствующим нормативным оформлением процедуры. Регулярно производится авторизованный мониторинг активности пользователя на рабочей станции.
 4. Все данные, объекты и информационные системы проклассифицированы. Субъекты распределены по ролям. Определена надежная структура и внедрен механизм доступа субъектов к объектам с учетом наименьших привилегий и разделения обязанностей. Каждый новый объект своевременно классифицируется и устанавливается в общую схему информационного пространства.
 5. Обеспечено нормативное пространство. Для всех информационных систем существуют политики, для функциональных обязанностей пользователей — правила, процедуры и методики. Изменения в работе организации и сотрудников адекватно отражаются в соответствующих документах.
 6. Способы аутентификации пользователей находятся под контролем службы информационной безопасности, т. е. производится периодический анализ отсутствия слабых паролей, фактов передачи паролей другим лицам, оставление токенов без надзора и т. п. Идентификация пользователей стандартизована, имеется четкая таблица соответствия пользователь — сетевые адреса, разрешенные для работы данного пользователя (username, IP-адрес, MAC-адрес и т. д.).
 7. Внешний удаленный доступ в информационную сеть предприятия (выход во внешнее информационное пространство) ограничен единственным центральным шлюзом плюс существует резервный канал связи, неактивный в штатном режиме. Оба канала защищены межсетевыми экранами, которые надежно функционируют, корректно настроены и регулярно подвергается проверке службой информационной безопасности. Все модемы и другие устройства удаленного доступа учтены и также сведены к указанной единой точке входа/выхода. Снаружи по отношению к точке входа установлен агент сетевой СОА.
 8. Агент сетевой системы обнаружения атак (СОА) присутствует на каждом сегменте в сети организации. Для СОА сигнатурного типа базы данных сигнатур регулярно обновляются, имеется специалист по созданию собственных сигнатур. Для СОА с выявлением аномалий определены все необходимые профили, которые регулярно пересматриваются. Кроме этого производится периодический анализ сетевой активности средствами сетевого мониторинга (перехватчиками сетевых пакетов — sniffерами).

9. Агенты СОА-хоста установлены на каждом узле сети, база данных атак (или профили) регулярно обновляется. Кроме этого производится периодический выборочный анализ регистрационных журналов (которые настроены на фиксацию всех событий), в том числе лично уполномоченным персоналом. СОА интегрирована с межсетевым экраном и другими устройствами защиты.
10. Настроено подробное ведение регистрационных журналов по всем информационным системам. Ведется регулярная автоматизированная обработка этих журналов, а также периодический выборочный ручной их анализ на предмет выявления подозрительных или злоумышленных событий. Система анализа информационной активности интегрирована с системой физического доступа сотрудников в здание и к рабочим местам. Все регистрационные журналы сохраняются наряду с резервными копиями и архивами бизнес-данных.
11. На каждый компьютер в сети установлен антивирусный пакет, кроме того антивирусы установлены на почтовый сервер, межсетевой экран и другие ключевые узлы. Режим работы антивируса — перехват на лету (autoprotect). Антивирусные базы обновляются ежедневно, а также при поступлении информации о новом вирусе.
12. Ведется строгий учет движения любой единицы аппаратного обеспечения или ее составляющей, а также строгий учет аппаратной или программной конфигурации каждого объекта или системы. При изменении конфигурации немедленно производится сохранение соответствующих настроек. Ведется регулярный мониторинг производительности работы аппаратного обеспечения, операционных систем, информационной системы в целом.
13. Произведена подписка на соответствующие бюллетени по информационной безопасности, изучаются сообщения о новых атаках, вирусах, уязвимостях и т. п., новых решениях и механизмах по безопасности. Налажена процедура регулярного получения и установки программных заплат (patches, hotfixes, updates и т. п.) и их тестирование. Производится аналитическая работа по определению тенденций развития атак, появлению уязвимостей, новых продуктов на рынке безопасности, новых технологий.
14. Все каналы обмена данными в информационной сети криптографически защищены, внутри локальной сети организованы виртуальные сети. Любой обмен данными регистрируется в электронных журналах и снабжен средством контроля целостности. Обеспечен контроль целостности данных, системных файлов и т. п. на серверах и рабочих станциях. Обмен данными между пользователями производится с использованием электронной цифровой подписи. Организовано надежное управление криптографическими ключами.

15. Обеспечен контроль входящей и исходящей информации на внешних носителях. Установлены надежные защищенные (возможно, виртуальные) шлюзы обмена информацией с внешними информационными системами (обеспечены соответствующие интерфейсы, установлена связь с центром сертификатов, получен корневой сертификат организации и т. д.).
16. Регулярно производится процедура тестирования всей сети или отдельных систем на взлом. В распоряжении службы информационной безопасности имеется команда программистов для создания собственных программ или утилит для анализа защищенности либо, наоборот, для защиты объектов и систем.
17. После изменения любой единицы данных обеспечивается ее резервное копирование либо организовано избыточное сохранение данных (RAID). Обеспечивается географически разнесенное защищенное сохранение архивов и резервных копий. Все сетевые устройства имеют возможность для быстрой замены (продублированы), все каналы передачи данных имеют альтернативные линии. Здание организации имеет горячий резерв (hot-site). Обеспечено бесперебойное электропитание (и контроль его работы) основного и резервного зданий. Ключевые работники организации могут выполнять функции друг друга либо имеют функциональных дублеров. Имеется регулярно обновляемый аварийный план.
18. Контроль за любой системой или объектом децентрализован. Служба информационной безопасности принимает участие в любом проекте организации (в том числе и в разработке программного обеспечения) с правом внесения серьезных изменений и запретов в рамках своих функций. Служба имеет полномочия к принятию и реализации решений, связанных с информационной безопасностью.
19. Установлена система отвлечения внимания злоумышленника (honeypot), сформирована собственная команда реагирования на инциденты, установлена связь с аналогичными командами других организаций, внешними экспертами, силовыми структурами.
20. Обеспечено единое согласованное непротиворечивое управление всеми автоматизированными средствами информационной безопасности.

Послесловие

Рассказывать об информационной безопасности, как о любом деле, в котором нужно использовать сочетание знаний и творческого подхода, можно еще очень долго, постепенно превращая книгу в многотомник под названием "Энциклопедия информационной безопасности".

Что же еще осталось не охваченным и почему?

- ❑ Безопасность телефонной связи: конфиденциальность в рамках офисных АТС (PBX), fax-security, IP-телефония — все это осталось за бортом, хотя является существенным фактором безопасности. Как минимум эти вопросы входят в экзамен на звание CISSP, о котором шла речь в начале книги.
- ❑ Технологии работы с данными. Data-Warehouse, data-mining могут являться вспомогательными технологиями для безопасности на больших предприятиях распределенного типа, когда для анализа набираются такие большие объемы данных, что количеству пора переходить в существенное качество. Построение статистических анализов, прогнозных моделей позволит службе информационной безопасности действовать проактивно, тем самым предупреждая возможные нарушения информационной безопасности.
- ❑ Конкретные методы программирования механизмов безопасности. Реализация механизмов защиты от копирования, использование стандартных библиотек по безопасности, GSSPI и многое другое.
- ❑ Модели и схемы безопасности отдельных систем, скажем, виртуальной Java-машины и вообще Java как языка программирования. Это именно тот случай, когда вопрос стоит отдельной книги (по счастью, такие книги уже есть).
- ❑ Психологические тесты, анализ, методы ведения интервью, а возможно и допросов. Все что необходимо для того, чтобы выявить предрасположенность потенциального или реального пользователя к компьютерным (или иным) преступлениям.

В отношении полностью пропущенных вопросов основной причиной является то, что изложение материала оказалось бы в данном объеме настолько поверхностным, что практически не дало бы никакой реальной пользы. Для более подробного их описания пришлось бы готовить отдельную книгу по размеру не меньше (а может и больше) этой.

В отношении глубины раскрытия освещенных вопросов авторы руководствовались следующим принципом: гарантированно создать необходимый минимум знаний о вопросе, превышая его лишь для наиболее часто встречающихся на практике средств и технологий информационной защиты.

Авторы искренне надеются, что представленный материал оказался полезным и нашел своего читателя.

Приложения

Приложение 1

Некоторые номера портов протоколов транспортного уровня UDP и TCP, официально зарегистрированные IANA (Internet Authority Network Associate) согласно RFC-1700 см. табл. П1.1.

Таблица П1.1

Порт (TCP/UDP)	Условное обозначение	Описание	Область применения
7	echo	Echo	Отладка сети
9	discard	Discard	Отладка сети
13	daytime	Daytime	Служба времени
19	chargen	Character Generator	Отладка сети
20	ftp-data	File Transfer [Default Data]	Передача файлов
21	ftp	File Transfer [Control]	Передача файлов
22	ssh	SSH Remote Login Protocol	Удаленное управление
23	telnet	Telnet	Удаленное управление
25	smtp	Simple Mail Transfer	Электронная почта
37	time	Time	Служба времени
43	nickname	Who Is	Сетевая безопасность
49	tacacs	Login Host Protocol (TACACS)	Сетевая безопасность
53	domain	Domain Name Server	Служба доменных имен
65	tacacs-ds	TACACS-Database Service	Сетевая безопасность
66	sql*net	Oracle SQL*NET	Базы данных
67	bootps	Bootstrap Protocol Server	Удаленная загрузка
68	bootpc	Bootstrap Protocol Client	Удаленная загрузка
69	tftp	Trivial File Transfer	Передача файлов

Таблица П1.1 (продолжение)

Порт (TCP/UDP)	Условное обозначение	Описание	Область применения
70	gopher	Gopher	Информационная служба
79	finger	Finger	Информационная служба
80	http	World Wide Web HTTP	WWW
88	kerberos	Kerberos	Сетевая безопасность
107	rtnet	Remote Telnet Service	Удаленное управление
108	snagas	SNA Gateway Access Server	Служба SNA
109	pop2	Post Office Protocol — Version 2	Электронная почта
110	pop3	Post Office Protocol — Version 3	Электронная почта
111	sunrpc	SUN Remote Procedure Call	Распределенная обработка
113	auth	Authentication Service	Сетевая безопасность
117	uucp-path	UUCP Path Service	Служба UUCP
118	sqlserv	SQL Services	Базы данных
119	nntp	Network News Transfer Protocol	Электронные новости
123	ntp	Network Time Protocol	Служба времени
129	pwdgen	Password Generator Protocol	Сетевая безопасность
137	netbios-ns	NETBIOS Name Service	Служба NETBIOS
138	netbios-dgm	NETBIOS Datagram Service	Служба NETBIOS
139	netbios-ssn	NETBIOS Session Service	Служба NETBIOS
143	imap	Internet Message Access Protocol	Электронная почта
150	sql-net	SQL-NET	Базы данных
156	sqlsrv	SQL Service	Базы данных
161	snmp	SNMP	Управление сетью
162	snmptrap	SNMPTRAP	Управление сетью

Таблица П1.1 (продолжение)

Порт (TCP/UDP)	Условное обозначение	Описание	Область применения
179	bgp	Border Gateway Protocol	Управление сетью
194	irc	Internet Relay Chat Protocol	Чат
213	ipx	IPX	Инкапсуляция
220	imap3	Interactive Mail Access Protocol v3	Электронная почта
257	set	Secure Electronic Transaction	Сетевая безопасность
389	ldap	Lightweight Directory Access Protocol	Информационная служба
396	netware-ip	Novell Netware over IP	Инкапсуляция
401	ups	Uninterruptible Power Supply	Бесперебойное питание
440	sgcp	sgcp	Удаленное управление
443	https	http protocol over TLS/SSL	WWW
445	microsoft-ds	Microsoft-DS (Windows 2000)	Служба NETBIOS
448	ddm-ssl	DDM-SSL	Удаленное управление
500	isakmp	isakmp	Сетевая безопасность
512	rexec	remote process execution service	Распределенная обработка
513	rlogin	remote login service	Удаленное управление
514	rshell	remote shell service	Удаленное управление
514	syslog	syslog service	Управление сетью
515	printer	spooler	Удаленная печать
518	ntalk		Чат
519	utime	unixtime	Служба времени
520	efs	extended file name server	Служба EFS
520	router	local routing process (on site) Xerox NS	Управление сетью
525	timed	timeserver	Служба времени

Таблица П1.1 (продолжение)

Порт (TCP/UDP)	Условное обозначение	Описание	Область применения
530	courier	rpc	Распределенная обработка
531	conference	chat	Чат
532	netnews	readnews	Электронные новости
540	uucpd	uucpd	Служба UUCP
541	uucp-rlogin	uucp-rlogin	Служба UUCP
563	nntps	nntp protocol over TLS/SSL (was snntp)	Электронные новости
585	imap4-ssl	IMAP4+SSL (use 993 in- stead)	Электронная почта
614	sshell	SSL shell	Удаленное управление
636	ldaps	ldap protocol over TLS/SSL (was sldap)	Информационная служба
749	kerberos-adm	kerberos administration	Сетевая безопасность
750	kerberos-4	kerberos version 4	Сетевая безопасность
989	ftps-data	ftp protocol, data, over TLS/SSL	Передача файлов
990	ftps	ftp protocol, control, over TLS/SSL	Передача файлов
992	telnets	telnet protocol over TLS/SSL	Удаленное управление
993	imaps	imap4 protocol over TLS/SSL	Электронная почта
994	ircs	irc protocol over TLS/SSL	Чат
995	pop3s	pop3 protocol over TLS/SSL (was spop3)	Электронная почта
1080	socks	Socks	Инкапсуляция
1155	nfa	Network File Access	Служба NFA
1352	lotusnote	Lotus Note	Офисные приложения
1433	ms-sql-s	Microsoft-SQL-Server	Базы данных
1434	ms-sql-m	Microsoft-SQL-Monitor	Базы данных
1477	ms-sna-server	ms-sna-server	Служба SNA

Таблица П1.1 (окончание)

Порт (TCP/UDP)	Условное обозначение	Описание	Область применения
1478	ms-sna-base	ms-sna-base	Служба SNA
1525	orasrv	oracle	Базы данных
1527	tlisrv	oracle	Базы данных
1529	coauthor	oracle	Базы данных
1993	snmp-tcp	cisco SNMP TCP port	Управление сетью
1997	gdp-port	cisco Gateway Discovery Protocol	Управление сетью
1998	x25-svc-port	cisco X.25 service (XOT)	Инкапсуляция
1999	tcp-id-port	cisco identification port	Управление сетью
4444	krb524	KRB524	Авторизация
5190	aol	America-Online	Чат
6000-6063	x11	X-Window System	Удаленное управление

Приложение 2

Список номеров портов (протоколы TCP и UDP), занимаемых для управления наиболее распространенными за последние годы троянскими программами. Источник: <http://no-trojans.ics.ucdavis.edu/> см. табл. П2.

Таблица П2.1

Номер порта	Название троянской программы
2	Death
20	Senna Spy FTP Server
21	Back Construction, Doly Trojan, Invisible FTP, Juggernaut 42, Larva, Net Administrator, Senna Spy FTP Server, Traitor 21, WebEx, WinCrash
22	Shaft
23	Fire Hacker, Tiny Telnet Server - TTS, Truva Atl
24	Corsar
25	Ajan, Antigen, Email Password Sender – EPS, EPS II, Gip, Gris, Happy99, Hpteam mail, I love you, Kuang2, Magic Horse, MBT (Mail Bombing Trojan), Moscow Email trojan, Naebi, NewApt worm, ProMail trojan, Shtirlitz, Stealth, Tapiras, Termi
31	Agent 31, Hacker's Paradise, Master's Paradise.
41	Deep Throat, Foreplay or Reduced Foreplay
48	DRAT
49	Miosa
50	DRAT
58	DM Setup
59	DM Setup
79	CDK, Firehotcker
80	AckCmd, Back End, CGI Backdoor, Executor, Hooker, RingZero, Executor

Таблица П2.1 (продолжение)

Номер порта	Название троянской программы
81	RemoConChubo
99	Hidden Port
100	Memory Leak
110	ProMail Trojan
113	Invisible Identd Deamon, Kazimas
119	Happy99
121	Jammer Killah
123	Net Controller
133	Farnaz
142	NetTaxi
146 (TCP/UDP)	Infector
170	A-trojan
334	Backage
405	Beast 2000
420	Breach
421	TCP Wrapper Trojan
456	Hacker's Paradise
513	Grlogin
514	RPC Backdoor
531	Rasmin
555	Ini-Killer , Net Administrator, Phaze Zero, Phase-0, Stealth Spy
605	Secret Service
666	Attack FTP, Back Construction, Cain & Abel, NokNok, Satans Back Door - SBD, ServU, Shadow Phyre
667	SniperNet
669	DP Trojan
692	GayOL

Таблица П2.1 (продолжение)

Номер порта	Название троянской программы
777	AimSpy, Undetected
808	WinHole
911	Dark Shadow
998	Leeloo4
999	Deep Throat, Foreplay or Reduced Foreplay, WinSatan, MisterOk
1000	Der Spaher / Der Spaeher
1001	Der Spaher / Der Spaeher, Le Gardien, Silencer, WebEx
1010	Doly Trojan
1010	Doly Trojan 1.30 (Subm.Cronco)
1011	Doly Trojan
1012	Doly Trojan
1015	Doly Trojan 1.5 (Subm.Cronco)
1020	Vampire
1024	NetSpy
1027	ICQ
1029	ICQ
1032	ICQ
1033	Netspy
1042	Bla1.1
1045	Rasmin
1049	/sbin/initd
1050	MiniCommand
1054	AckCmd
1080	WinHole
1081	WinHole
1082	WinHole
1090	Xtreme

Таблица П2.1 (продолжение)

Номер порта	Название троянской программы
1095	Remote Administration Tool — RAT
1097	Remote Administration Tool — RAT
1098	Remote Administration Tool — RAT
1099	Blood Fest Evolution, Remote Administration Tool — RAT
1170	Psyber Stream Server — PSS, Streaming Audio Server, Voice
1200 (UDP)	NoBackO
1201 (UDP)	NoBackO
1207	SoftWAR
1212	Kaos
1234	Ultors Trojan
1243	BackDoor-G, SubSeven, SubSeven Apocalypse, Tiles
1245	Voodoo Doll
1255	Scarab
1256	Project NEXT
1269	Matrix
1313	NETrojan
1338	Millenium Worm
1349	Bo dll
1492	FTP99CMP
1509	PsyberStreamingServer Nikhil G.
1524	Trinoo
1600	Shivka Burka
1777	Scarab
1807	SpySender Nikhil G.
1966	Fake FTP
1969	OpC BO
1981	Bowl, Shockrave

Таблица П2.1 (продолжение)

Номер порта	Название троянской программы
1981	Shockrave
1998	Remontrik
1999	Back Door, TransScout
2000	Der Spaher / Der Spæher, Insane Network, Millenium
2001	Der Spaher / Der Spæher, Trojan Cow, Millenium, Millenium (Lm)
2023	Ripper, Ripper Pro
2080	WinHole
2115	Bugs
2140 (UDP)	Deep Throat, Foreplay or Reduced Foreplay
2155	Illusion Mailer
2255	Nirvana
2283	Hvi RAT
2283	HVL Rat5
2300	Xplorer
2339 (TCP/UDP)	Voice Spy
2345	Doly Trojan
2565	Striker Trojan
2583	WinCrash
2600	Digital RootBeer
2716	The Prayer
2773	SubSeven, SubSeven 2.1 Gold
2801	Phineas Phucker
2989 (UDP)	Remote Administration Tool — RAT
3000	Remote Shut
3024	WinCrash
3128	RingZero
3129	Masters Paradise

Таблица П2.1 (продолжение)

Номер порта	Название троянской программы
3150	The Invasor
3150 (UDP)	Deep Throat, Foreplay, Reduced Foreplay
3389 (UDP)	Terminal Server Attack
3456	Terror Trojan
3459	Eclipse 2000, Sanctuary
3700	Portal of Doom — POD
3791	Total Solar Eclypse
3801	Total Solar Eclypse
4000	Skydance
4092	WinCrash
4242	Virtual Hacking Machine — VHM
4321	BoBo
4444	Prosiak, Swift Remote
4567	FileNail Danny
4950	ICQ Trogen (Lm)
5000	Back Door Setup, Blazer5, Bubbel, ICKiller, Sockets des Troie, Socket23
5001	Back Door Setup, Sockets des Troie
5002	Shaft, cd00r
5010	Solo
5011	One of the Last Trojans — OOTLT
5025	WM Remote KeyLogger
5031	Net Metropolitan
5321	FileHotcker
5343	wCrat — WC Remote Administration Tool
5400	Back Construction, Blade Runner
5401	Back Construction, Blade Runner
5402	Back Construction, Blade Runner

Таблица П2.1 (продолжение)

Номер порта	Название троянской программы
5512	Illusion Mailer
5550	Xtcp
5555	ServeMe
5556	BO Facil
5557	BO Facil
5569	Robo-Hack
5637	PC Crasher
5638	PC Crasher
5666	Last Week
5742	WinCrash
5760	Portmap Remote Root Linux Exploit
5882 (UDP)	Y3K RAT
5888	Y3K RAT
6000	The Thing
6006	Bad Blood
6272	Secret Service
6400	The Thing
6666	Dark Connection Inside, NetBus worm
6667	ScheduleAgent, Trinity, WinSatan
6669	Host Control, Vampire
6670	BackWeb Server, Deep Throat, Foreplay or Reduced Foreplay, WinNuke eXtreame
6670	Deep Throat
6711	BackDoor-G, SubSeven, VP Killer
6712	Funny trojan, SubSeven
6713	Sub Seven
6723	Mstream

Таблица П2.1 (продолжение)

Номер порта	Название троянской программы
6771	Deep Throat, Foreplay or Reduced Foreplay
6776	2000 Cracks, BackDoor-G, SubSeven, VP Killer
6838 (UDP)	Mstream
6883	Delta Source (Dark Star)
6912	Shit Heep
6939	Indoctrination
6969	GateCrasher, IRC 3, Net Controller, Priority
6970	GateCrasher
7000	Exploit Translation Server, Kazimas, Remote Grab, SubSeven 2.1 Gold
7001	Freak8
7215	SubSeven, SubSeven 2.1 Gold
7300	NetMonitor
7301	NetMonitor
7306	NetMonitor
7307	NetMonitor
7308	NetMonitor
7789	ICKiller, Back Door Setup
7983	Mstream
8080	Brown Orifice, RemoConChubo, RingZero
8787	Back Orifice 2000
8988	BacHack
8989	Rcon, Recon, Xcon
9000	Netministrator
9001	Sunday Times
9325 (UDP)	Mstream
9400	InCommand
9872	Portal of Doom — POD

Таблица П2.1 (продолжение)

Номер порта	Название троянской программы
9873	Portal of Doom — POD
9874	Portal of Doom — POD
9875	Portal of Doom — POD
9876	Cyber Attacker, Rux
10067 (UDP)	Portal of Doom — POD
10528	Host Control
10607	Coma
10666 (UDP)	Ambush
11000	Senna Spy Trojan Generator
11050	Host Control
11051	Host Control
11223	Progenic trojan, Secret Agent
12076	Gjamer
12223	Hack?99, KeyLogger
12345	cron / crontab, Fat Bitch trojan, GabanBus, icmp_pipe.c, Mypic , NetBus, Ultor's Telnet Trojan
12346	Fat Bitch trojan, GabanBus, NetBus, X-bill
12349	BioNet
12362	Whack-a-mole
12456	NetBus
12623 (UDP)	DUN Control
12626	ButtMan
12631	Whack Job
12754	Mstream
13000	Senna Spy Trojan Generator
13010	Hacker Brasil - HBR
14500	PC Invader, Martinica

Таблица П2.1 (продолжение)

Номер порта	Название троянской программы
15092	Host Control
15104	Mstream
15858	CDK
16484	Mosucker
16660	Stacheldraht
16772	ICQ Revenge
16969	Priority
17166	Mosaic
17300	Kuang2
17449	CrazyNet
17777	Nephron
18753 (UDP)	Shaft
19864	ICQ Revenge
20000	Millennium, Black List
20001	AcidkoR, Millennium
20023	VP Killer
20034	NetBus 2.0 Pro, NetRex, Whack Job
22222	Donald Dick, Prosiak
23005	NetTrash
23023	Logged
23032	Amanda
23432	Asylum
23456	Evil FTP, Ugly FTP, Whack Job
23476 (TCP/UDP)	Donald Dick
23477	Donald Dick
26274 (UDP)	Delta Source

Таблица П2.1 (продолжение)

Номер порта	Название троянской программы
26681	Voice Spy
27374	Bad Blood, SubSeven , SubSeven 2.1 Gold
27444 (UDP)	Trinoo
27573	SubSeven
27665	Trinoo
29104	NetTrojan
30029	AOLTrojan1.1
30303	Sockets de Troie
31336	Bo Whack, Butt Funnel
31337	Back Fire, Back Orifice (Lm), Back Orifice russian, Baron Night, Beeone, BOclient, BO Facil, BO spy, BO2, cron / crontab, Freak88, icmp_pipe.c, Sockdmini
31337 (UDP)	Back Orifice, Deep BO
31337	Netpatch
31338	Back Orifice, Butt Funnel, NetSpy (DK)
31338 (UDP)	Deep BO
31785	Hack'a'Tack
31787	Hack'a'tack
31788	Hack'a'Tack
31789 (UDP)	Hack'a'Tack
31790	Hack'a'Tack
31791 (UDP)	Hack'a'Tack
32001	Donald Dick
32100	Peanut Brittle, Project nEXT
32418	Acid Battery
33270	Trinity
33333	Blakharaz, Prosiak, Carmarash
34555 (UDP)	WinTrinoo

Таблица П2.1 (окончание)

Номер порта	Название троянской программы
35555 (UDP)	WinTrinoo
37651	Yet Another Trojan — YAT
40412	The Spy
40412	The Spy
44444	Prosiak, Carmarash
47252	Delta Source
47262 (UDP)	Delta Source
47262 (UDP)	Delta Source
50505	Sockets des Troie
54320	Back Orifice
54320 (UDP)	Back Orifice
54321	Back Orifice, School Bus
54321	School Bus, Back Orifice
54321 (UDP)	BackOrifice 2000
57341	NetRaider
58339	Butt Funnel
60000	Deep Throat, Foreplay or Reduced Foreplay, Sockets des Troie
64101	Taskman / Task Manager
65000	Devil, Sockets des Troie, Stacheldraht
65535	RC1 trojan

Приложение 3

Вероятности $p(C)$ появления символов в среднестатистическом тексте: для русского языка (см. табл. ПЗ.1), английского языка (см. табл. ПЗ.3).

Таблица ПЗ.1. Вероятности $p(C)$ появления символов в среднестатистическом тексте для русского языка

Буква	$p(C)$	Буква	$p(C)$	Буква	$p(C)$
О	0,090	М	0,026	Й	0,010
Е, Ё	0,072	Д	0,025	Х	0,009
А	0,062	П	0,023	Ж	0,007
И	0,062	У	0,021	Ш	0,006
Н	0,053	Я	0,018	Ю	0,006
Т	0,053	З	0,016	Ц	0,004
С	0,045	Ы	0,016	Щ	0,003
Р	0,040	Б	0,014	Э	0,003
В	0,038	Ь, Ь	0,014	Ф	0,002
Л	0,035	Г	0,013		
К	0,028	Ч	0,012		

Таблица ПЗ.2. Вероятности $p(C)$ появления символов в среднестатистическом тексте для английского языка

Буква	$p(C)$	Буква	$p(C)$	Буква	$p(C)$
Е	0,131	Н	0,071	Н	0,053
Т	0,104	Р	0,068	Д	0,038
А	0,082	І	0,063	L	0,034
О	0,080	S	0,061	F	0,029

Таблица ПЗ.2 (окончание)

Буква	p(C)	Буква	p(C)	Буква	p(C)
C	0,028	P	0,020	X	0,002
M	0,025	W	0,015	J	0,001
U	0,024	B	0,014	Q	0,001
G	0,020	V	0,009	Z	0,001
Y	0,020	K	0,004		

Приложение 4

Табличные подстановки S0...S7 алгоритма DES (см. табл. П4.1—П4.8)

Таблица П4.1. Табличные подстановки S0

xxxx	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0xxx0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0xxx1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
1xxx0	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
1xxx1	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Таблица П4.2. Табличные подстановки S1

xxxx	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0xxx0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
0xxx1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
1xxx0	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
1xxx1	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Таблица П4.3. Табличные подстановки S2

xxxx	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0xxx0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
0xxx1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
1xxx0	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1xxx1	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

Таблица П4.4. Табличные подстановки S3

xxxx	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0xxxx0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
0xxxx1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
1xxxx0	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
1xxxx1	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Таблица П4.5. Табличные подстановки S4

xxxx	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0xxxx0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
0xxxx1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
1xxxx0	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
1xxxx1	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

Таблица П4.6. Табличные подстановки S5

xxxx	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0xxxx0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
0xxxx1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
1xxxx0	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
1xxxx1	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

Таблица П4.7. Табличные подстановки S6

xxxx	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0xxxx0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
0xxxx1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1xxxx0	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
1xxxx1	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

Таблица П4.8. Табличные подстановки S7

xxxx	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0xxxx0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
0xxxx1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
1xxxx0	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
1xxxx1	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Приложение 5

Табличная подстановка S алгоритма MARS (разрядность: 9х32 бита)

0x09d0c479, 0x28c8ffe0, 0x84aa6c39, 0x9dad7287, 0x7dff9be3, 0xd4268361,
0xc96da1d4, 0x7974cc93, 0x85d0582e, 0x2a4b5705, 0x1ca16a62, 0xc3bd279d,
0x0f1f25e5, 0x5160372f, 0xc695c1fb, 0x4d7ff1e4, 0xae5f6bf4, 0x0d72ee46,
0xff23de8a, 0xb1cf8e83, 0xf14902e2, 0x3e981e42, 0x8bf53eb6, 0x7f4bf8ac,
0x83631f83, 0x25970205, 0x76afe784, 0x3a7931d4, 0x4f846450, 0x5c64c3f6,
0x210a5f18, 0xc6986a26, 0x28f4e826, 0x3a60a81c, 0xd340a664, 0x7ea820c4,
0x526687c5, 0x7eddd12b, 0x32a11dd1, 0x9c9ef086, 0x80f6e831, 0xab6f04ad,
0x56fb9b53, 0x8b2e095c, 0xb68556ae, 0xd2250b0d, 0x294a7721, 0xe21fb253,
0xae136749, 0xe82aae86, 0x93365104, 0x99404a66, 0x78a784dc, 0xb69ba84b,
0x04046793, 0x23db5c1e, 0x46cae1d6, 0x2fe28134, 0x5a223942, 0x1863cd5b,
0xc190c6e3, 0x07dfb846, 0x6eb88816, 0x2d0dcc4a, 0xa4ccae59, 0x3798670d,
0xcbfa9493, 0x4f481d45, 0xeafc8ca8, 0xdb1129d6, 0xb0449e20, 0x0f5407fb,
0x6167d9a8, 0xd1f45763, 0x4daa96c3, 0x3bec5958, 0xababa014, 0xb6ccd201,
0x38d6279f, 0x02682215, 0x8f376cd5, 0x092c237e, 0xbfc56593, 0x32889d2c,
0x854b3e95, 0x05bb9b43, 0x7dcd5dcd, 0xa02e926c, 0xfae527e5, 0x36a1c330,
0x3412e1ae, 0xf257f462, 0x3c4f1d71, 0x30a2e809, 0x68e5f551, 0x9c61ba44,
0x5ded0ab8, 0x75ce09c8, 0x9654f93e, 0x698c0cca, 0x243cb3e4, 0x2b062b97,
0x0f3b8d9e, 0x00e050df, 0xfc5d6166, 0xe35f9288, 0xc079550d, 0x0591aee8,
0x8e531e74, 0x75fe3578, 0x2f6d829a, 0xf60b21ae, 0x95e8eb8d, 0x6699486b,
0x901d7d9b, 0xfd6d6e31, 0x1090acef, 0xe0670dd8, 0xdab2e692, 0xcd6d4365,
0xe5393514, 0x3af345f0, 0x6241fc4d, 0x460da3a3, 0x7bcf3729, 0x8bf1d1e0,
0x14aac070, 0x1587ed55, 0x3afd7d3e, 0xd2f29e01, 0x29a9d1f6, 0xefb10c53,
0xcf3b870f, 0xb414935c, 0x664465ed, 0x024acac7, 0x59a744c1, 0x1d2936a7,
0xdc580aa6, 0xcf574ca8, 0x040a7a10, 0x6cd81807, 0x8a98be4c, 0xaccea063,
0xc33e92b5, 0xd1e0e03d, 0xb322517e, 0x2092bd13, 0x386b2c4a, 0x52e8dd58,
0x58656dfb, 0x50820371, 0x41811896, 0xe337ef7e, 0xd39fb119, 0xc97f0df6,
0x68fea01b, 0xa150a6e5, 0x55258962, 0xeb6ff41b, 0xd7c9cd7a, 0xa619cd9e,
0xbcf09576, 0x2672c073, 0xf003fb3c, 0x4ab7a50b, 0x1484126a, 0x487ba9b1,
0xa64fc9c6, 0xf6957d49, 0x38b06a75, 0xdd805fcd, 0x63d094cf, 0xf51c999e,

0x1aa4d343, 0xb8495294, 0xce9f8e99, 0xbffcd770, 0xc7c275cc, 0x378453a7,
0x7b21be33, 0x397f41bd, 0x4e94d131, 0x92cc1f98, 0x5915ea51, 0x99f861b7,
0xc9980a88, 0x1d74fd5f, 0xb0a495f8, 0x614deed0, 0xb5778eea, 0x5941792d,
0xfa90c1f8, 0x33f824b4, 0xc4965372, 0x3ff6d550, 0x4ca5fec0, 0x8630e964,
0x5b3fbbd6, 0x7da26a48, 0xb203231a, 0x04297514, 0x2d639306, 0x2eb13149,
0x16a45272, 0x532459a0, 0x8e5f4872, 0xf966c7d9, 0x07128dc0, 0x0d44db62,
0xafc8d52d, 0x06316131, 0xd838e7ce, 0x1bc41d00, 0x3a2e8c0f, 0xea83837e,
0xb984737d, 0x13ba4891, 0xc4f8b949, 0xa6d6acb3, 0xa215cdce, 0x8359838b,
0x6bd1aa31, 0xf579dd52, 0x21b93f93, 0xf5176781, 0x187dfdde, 0xe94aeb76,
0x2b38fd54, 0x431de1da, 0xab394825, 0x9ad3048f, 0xdfea32aa, 0x659473e3,
0x623f7863, 0xf3346c59, 0xab3ab685, 0x3346a90b, 0x6b56443e, 0xc6de01f8,
0x8d421fc0, 0x9b0ed10c, 0x88f1a1e9, 0x54c1f029, 0x7dead57b, 0x8d7ba426,
0x4cf5178a, 0x551a7cca, 0x1a9a5f08, 0xfcd651b9, 0x25605182, 0xe11fc6c3,
0xb6fd9676, 0x337b3027, 0xb7c8eb14, 0x9e5fd030,

0x6b57e354, 0xad913cf7, 0x7e16688d, 0x58872a69, 0x2c2fc7df, 0xe389ccc6,
0x30738df1, 0x0824a734, 0xe1797a8b, 0xa4a8d57b, 0x5b5d193b, 0xc8a8309b,
0x73f9a978, 0x73398d32, 0x0f59573e, 0xe9df2b03, 0xe8a5b6c8, 0x848d0704,
0x98df93c2, 0x720a1dc3, 0x684f259a, 0x943ba848, 0xa6370152, 0x863b5ea3,
0xd17b978b, 0x6d9b58ef, 0x0a700dd4, 0xa73d36bf, 0x8e6a0829, 0x8695bc14,
0xe35b3447, 0x933ac568, 0x8894b022, 0x2f511c27, 0xddfbcc3c, 0x006662b6,
0x117c83fe, 0x4e12b414, 0xc2bca766, 0x3a2fec10, 0xf4562420, 0x55792e2a,
0x46f5d857, 0xcda25ce, 0xc3601d3b, 0x6c00ab46, 0xefac9c28, 0xb3c35047,
0x611dfee3, 0x257c3207, 0xfdd58482, 0x3b14d84f, 0x23becb64, 0xa075f3a3,
0x088f8ead, 0x07adf158, 0x7796943c, 0xfacabf3d, 0xc09730cd, 0xf7679969,
0xda44e9ed, 0x2c854c12, 0x35935fa3, 0x2f057d9f, 0x690624f8, 0x1cb0bafd,
0x7b0dbdc6, 0x810f23bb, 0xfa929a1a, 0x6d969a17, 0x6742979b, 0x74ac7d05,
0x010e65c4, 0x86a3d963, 0xf907b5a0, 0xd0042bd3, 0x158d7d03, 0x287a8255,
0xbba8366f, 0x096edc33, 0x21916a7b, 0x77b56b86, 0x951622f9, 0xa6c5e650,
0x8cea17d1, 0xcd8c62bc, 0xa3d63433, 0x358a68fd, 0xf9b9d3c, 0xd6aa295b,
0xfe33384a, 0xc000738e, 0xcd67eb2f, 0xe2eb6dc2, 0x97338b02, 0x06c9f246,
0x419cflad, 0x2b83c045, 0x3723f18a, 0xcb5b3089, 0x160bead7, 0x5d494656,
0x35f8a74b, 0x1e4e6c9e, 0x000399bd, 0x67466880, 0xb4174831, 0xacf423b2,
0xca815ab3, 0x5a6395e7, 0x302a67c5, 0x8bdb446b, 0x108f8fa4, 0x10223eda,
0x92b8b48b, 0x7f38d0ee, 0xab2701d4, 0x0262d415, 0xaf224a30, 0xb3d88aba,
0xf8b2c3af, 0xdaf7ef70, 0xcc97d3b7, 0xe9614b6c, 0x2baebff4, 0x70f687cf,
0x386c9156, 0xce092ee5, 0x01e87da6, 0x6ce91e6a, 0xbb7bcc84, 0xc7922c20,
0x9d3b71fd, 0x060e41c6, 0xd7590f15, 0xe03bb47, 0x183c198e, 0x63eeb240,

0x2ddbf49a, 0x6d5cba54, 0x923750af, 0xf9e14236, 0x7838162b, 0x59726c72,
0x81b66760, 0xbb2926c1, 0x48a0ce0d, 0xa6c0496d, 0xad43507b, 0x718d496a,
0x9df057af, 0x44b1bde6, 0x054356dc, 0xde7ced35, 0xd51a138b, 0x62088cc9,
0x35830311, 0xc96efca2, 0x686f86ec, 0x8e77cb68, 0x63e1d6b8, 0xc80f9778,
0x79c491fd, 0x1b4c67f2, 0x72698d7d, 0x5e368c31, 0xf7d95e2e, 0xa1d3493f,
0xdcd9433e, 0x896f1552, 0x4bc4ca7a, 0xa6d1baf4, 0xa5a96dcc, 0x0bef8b46,
0xa169fda7, 0x74df40b7, 0x4e208804, 0x9a756607, 0x038e87c8, 0x20211e44,
0x8b7ad4bf, 0xc6403f35, 0x1848e36d, 0x80bdb038, 0x1e62891c, 0x643d2107,
0xbf04d6f8, 0x21092c8c, 0xf644f389, 0x0778404e, 0x7b78adb8, 0xa2c52d53,
0x42157abe, 0xa2253e2e, 0x7bf3f4ae, 0x80f594f9, 0x953194e7, 0x77eb92ed,
0xb3816930, 0xda8d9336, 0xbf447469, 0xf26d9483, 0xee6faed5, 0x71371235,
0xde425f73, 0xb4e59f43, 0x7dbe2d4e, 0x2d37b185, 0x49dc9a63, 0x98c39d98,
0x1301c9a2, 0x389b1bbf, 0x0c18588d, 0xa421c1ba, 0x7aa3865c, 0x71e08558,
0x3c5cfcaa, 0x7d239ca4, 0x0297d9dd, 0xd7dc2830, 0x4b37802b, 0x7428ab54,
0xaeee0347, 0x4b3fbb85, 0x692f2f08, 0x134e578e, 0x36d9e0bf, 0xae8b5fcf,
0xedb93ecf, 0x2b27248e, 0x170eb1ef, 0x7dc57fd6, 0x1e760f16, 0xb1136601,
0x864e1b9b, 0xd7ea7319, 0x3ab871bd, 0xcfa4d76f, 0xe31bd782, 0x0dbeb469,
0xabb96061, 0x5370f85d, 0xffb07e37, 0xda30d0fb, 0xebc977b6, 0x0b98b40f,
0x3a4d0fe6, 0xdf4fc26b, 0x159cf22a, 0xc298d6e2, 0x2b78ef6a, 0x61a94ac0,
0xab561187, 0x14eea0f0, 0xdf0d4164, 0x19af70ee

Приложение 6

Табличная подстановка S алгоритма RIJNDAEL (разрядность 8x8 бит)

99,124,119,123,242,107,111,197, 48, 1,103, 43,254,215,171,118,
202,130,201,125,250, 89, 71,240,173,212,162,175,156,164,114,192,
183,253,147, 38, 54, 63,247,204, 52,165,229,241,113,216, 49, 21,
4,199, 35,195, 24,150, 5,154, 7, 18,128,226,235, 39,178,117,
9,131, 44, 26, 27,110, 90,160, 82, 59,214,179, 41,227, 47,132,
83,209, 0,237, 32,252,177, 91,106,203,190, 57, 74, 76, 88,207,
208,239,170,251, 67, 77, 51,133, 69,249, 2,127, 80, 60,159,168,
81,163, 64,143,146,157, 56,245,188,182,218, 33, 16,255,243,210,
205, 12, 19,236, 95,151, 68, 23,196,167,126, 61,100, 93, 25,115,
96,129, 79,220, 34, 42,144,136, 70,238,184, 20,222, 94, 11,219,
224, 50, 58, 10, 73, 6, 36, 92,194,211,172, 98,145,149,228,121,
231,200, 55,109,141,213, 78,169,108, 86,244,234,101,122,174, 8,
186,120, 37, 46, 28,166,180,198,232,221,116, 31, 75,189,139,138,
112, 62,181,102, 72, 3,246, 14, 97, 53, 87,185,134,193, 29,158,
225,248,152, 17,105,217,142,148,155, 30,135,233,206, 85, 40,223,
140,161,137, 13,191,230, 66,104, 65,153, 45, 15,176, 84,187, 22;

Инверсия табличной подстановки S

82, 9,106,213, 48, 54,165, 56,191, 64,163,158,129,243,215,251,
124,227, 57,130,155, 47,255,135, 52,142, 67, 68,196,222,233,203,
84,123,148, 50,166,194, 35, 61,238, 76,149, 11, 66,250,195, 78,
8, 46,161,102, 40,217, 36,178,118, 91,162, 73,109,139,209, 37,
114,248,246,100,134,104,152, 22,212,164, 92,204, 93,101,182,146,
108,112, 72, 80,253,237,185,218, 94, 21, 70, 87,167,141,157,132,
144,216,171, 0,140,188,211, 10,247,228, 88, 5,184,179, 69, 6,
208, 44, 30,143,202, 63, 15, 2,193,175,189, 3, 1, 19,138,107,
58,145, 17, 65, 79,103,220,234,151,242,207,206,240,180,230,115,
150,172,116, 34,231,173, 53,133,226,249, 55,232, 28,117,223,110,

71,241, 26,113, 29, 41,197,137,111,183, 98, 14,170, 24,190, 27,
252, 86, 62, 75,198,210,121, 32,154,219,192,254,120,205, 90,244,
31,221,168, 51,136, 7,199, 49,177, 18, 16, 89, 39,128,236, 95,
96, 81,127,169, 25,181, 74, 13, 45,229,122,159,147,201,156,239,
160,224, 59, 77,174, 42,245,176,200,235,187, 60,131, 83,153, 97,
23, 43, 4,126,186,119,214, 38,225,105, 20, 99, 85, 33, 12,125;

Приложение 7

Входная перестановка бит алгоритма Serpent (см. табл. П7.1).

Элемент со значением V в списке на позиции P означает, что при перестановке бит в выходном потоке на позиции P должен находиться бит, стоявший ранее в позиции V . Например, в списке на 3-й позиции (счет ведется с 0) записано число 96 — $P = 3$, $V = 96$. Это означает, что при перестановке в выходном потоке на 3-й позиции должен находиться бит, стоявший на 96-й позиции во входном блоке данных.

Таблица П7.1

Позиция (P)	Значение (V)	Позиция (P)	Значение (V)
1	0	17	4
2	32	18	36
3	64	19	68
4	96	20	100
5	1	21	5
6	33	22	37
7	65	23	69
8	97	24	101
9	2	25	6
10	34	26	38
11	66	27	70
12	98	28	102
13	3	29	7
14	35	30	39
15	67	31	71
16	99	32	103

Таблица П7.1 (продолжение)

Позиция (Р)	Значение (V)	Позиция (Р)	Значение (V)
33	8	60	110
34	40	61	15
35	72	62	47
36	104	63	79
37	9	64	111
38	41	65	16
39	73	66	48
40	105	67	80
41	10	68	112
42	42	69	17
43	74	70	49
44	106	71	81
45	11	72	113
46	43	73	18
47	75	74	50
48	107	75	82
49	12	76	114
50	44	77	19
51	76	78	51
52	108	79	83
53	13	80	115
54	45	81	20
55	77	82	52
56	109	83	84
57	14	84	116
58	46	85	21
59	78	86	53

Таблица П7.1 (окончание)

Позиция (Р)	Значение (V)	Позиция (Р)	Значение (V)
87	85	108	122
88	117	109	27
89	22	110	59
90	54	111	91
91	86	112	123
92	118	113	28
93	23	114	60
94	55	115	92
95	87	116	124
96	119	117	29
97	24	118	61
98	56	119	93
99	88	120	125
100	120	121	30
101	25	122	62
102	57	123	94
103	89	124	126
104	121	125	31
105	26	126	63
106	58	127	95
107	90	128	127

Приложение 8

Табличные подстановки S0...S7 алгоритма Serpent (разрядность 4x4 бита)

S0	3	8	15	1	10	6	5	11	14	13	4	2	7	0	9	12
S1	15	12	2	7	9	0	5	10	1	11	14	8	6	13	3	4
S2	8	6	7	9	3	12	10	15	13	1	14	4	0	11	5	2
S3	0	15	11	8	12	9	6	3	13	1	2	4	10	7	5	14
S4	1	15	8	3	12	0	11	6	2	5	4	10	9	14	7	13
S5	15	5	2	11	4	10	9	12	0	3	14	8	13	6	7	1
S6	7	2	12	5	8	4	6	11	14	9	1	15	13	3	10	0
S7	1	13	15	0	14	8	2	11	7	4	12	10	9	3	5	6

Приложение 9

Линейное преобразование алгоритма Serpent (см. табл. П9.1)

Запись вида $\{V_1 V_2 \dots V_N\}$, в позиции P означает, что бит выходного потока в позиции P вычисляется как исключающее ИЛИ (XOR) входных бит в позициях $V_1, V_2, \dots, V_N$. Например, запись $\{36\ 90\ 103\}$, в позиции 3 (счет ведется с нуля) описывает формулу:

$$\text{OUT}[3] = \text{IN}[36] \text{ XOR } \text{IN}[90] \text{ XOR } \text{IN}[103]$$

Таблица П9.1.

Позиция (P)	Список индексов $\{V_1 V_2 \dots V_N\}$
1	{16 52 56 70 83 94 105}
2	{72 114 125}
3	{2 9 15 30 76 84 126}
4	{36 90 103}
5	{20 56 60 74 87 98 109}
6	{1 76 118}
7	{2 6 13 19 34 80 88}
8	{40 94 107}
9	{24 60 64 78 91 102 113}
10	{5 80 122}
11	{6 10 17 23 38 84 92}
12	{44 98 111}
13	{28 64 68 82 95 106 117}
14	{9 84 126}
15	{10 14 21 27 42 88 96}
16	{48 102 115}

Таблица П9.1 (продолжение)

Позиция (Р)	Список индексов $\{V_1 V_2 \dots V_N\}$
17	{32 68 72 86 99 110 121}
18	{2 13 88}
19	{14 18 25 31 46 92 100}
20	{52 106 119}
21	{36 72 76 90 103 114 125}
22	{6 17 92}
23	{18 22 29 35 50 96 104}
24	{56 110 123}
25	{1 40 76 80 94 107 118}
26	{10 21 96}
27	{22 26 33 39 54 100 108}
28	{60 114 127}
29	{5 44 80 84 98 111 122}
30	{14 25 100}
31	{26 30 37 43 58 104 112}
32	{3 118 }
33	{9 48 84 88 102 115 126}
34	{18 29 104}
35	{30 34 41 47 62 108 116}
36	{7 122 }
37	{2 13 52 88 92 106 119}
38	{22 33 108}
39	{34 38 45 51 66 112 120}
40	{11 126 }
41	{6 17 56 92 96 110 123}
42	{26 37 112}
43	{38 42 49 55 70 116 124}

Таблица П9.1 (продолжение)

Позиция (P)	Список индексов $\{V_1 V_2 \dots V_N\}$
44	{2 15 76}
45	{10 21 60 96 100 114 127}
46	{30 41 116}
47	{0 42 46 53 59 74 120}
48	{6 19 80}
49	{3 14 25 100 104 118 }
50	{34 45 120}
51	{4 46 50 57 63 78 124}
52	{10 23 84}
53	{7 18 29 104 108 122 }
54	{38 49 124}
55	{0 8 50 54 61 67 82}
56	{14 27 88}
57	{11 22 33 108 112 126 }
58	{0 42 53}
59	{4 12 54 58 65 71 86}
60	{18 31 92}
61	{2 15 26 37 76 112 116}
62	{4 46 57}
63	{8 16 58 62 69 75 90}
64	{22 35 96}
65	{6 19 30 41 80 116 120}
66	{8 50 61}
67	{12 20 62 66 73 79 94}
68	{26 39 100}
69	{10 23 34 45 84 120 124}

Таблица П9.1 (продолжение)

Позиция (P)	Список индексов $\{V_1 V_2 \dots V_N\}$
70	{12 54 65}
71	{16 24 66 70 77 83 98}
72	{30 43 104}
73	{0 14 27 38 49 88 124}
74	{16 58 69}
75	{20 28 70 74 81 87 102}
76	{34 47 108}
77	{0 4 18 31 42 53 92}
78	{20 62 73}
79	{24 32 74 78 85 91 106}
80	{38 51 112}
81	{4 8 22 35 46 57 96}
82	{24 66 77}
83	{28 36 78 82 89 95 110}
84	{42 55 116}
85	{8 12 26 39 50 61 100}
86	{28 70 81}
87	{32 40 82 86 93 99 114}
88	{46 59 120}
89	{12 16 30 43 54 65 104}
90	{32 74 85}
91	{36 90 103 118 }
92	{50 63 124}
93	{16 20 34 47 58 69 108}
94	{36 78 89}
95	{40 94 107 122 }

Таблица П9.1 (продолжение)

Позиция (P)	Список индексов $\{V_1 V_2 \dots V_N\}$
96	{0 54 67}
97	{20 24 38 51 62 73 112}
98	{40 82 93}
99	{44 98 111 126 }
100	{4 58 71}
101	{24 28 42 55 66 77 116}
102	{44 86 97}
103	{2 48 102 115 }
104	{8 62 75}
105	{28 32 46 59 70 81 120}
106	{48 90 101}
107	{6 52 106 119 }
108	{12 66 79}
109	{32 36 50 63 74 85 124}
110	{52 94 105}
111	{10 56 110 123 }
112	{16 70 83}
113	{0 36 40 54 67 78 89}
114	{56 98 109}
115	{14 60 114 127 }
116	{20 74 87}
117	{4 40 44 58 71 82 93}
118	{60 102 113}
119	{3 18 72 114 118 125 }
120	{24 78 91}
121	{8 44 48 62 75 86 97}

Таблица П9.1 (окончание)

Позиция (Р)	Список индексов $\{V_1 V_2 \dots V_N\}$
122	{64 106 117}
123	{1 7 22 76 118 122 }
124	{28 82 95}
125	{12 48 52 66 79 90 101}
126	{68 110 121}
127	{5 11 26 80 122 126 }
128	{32 86 99}

Приложение 10

Замена табличных подстановок Serpent в режиме bitslice

(e|f|g|h) = S0 (a|b|c|d)

```
t1 = a XOR d;  
t2 = a AND d;  
t3 = c XOR t1;  
t6 = b AND t1;  
t4 = b XOR t3;  
t10 = NOT t3;  
h = t2 XOR t4;  
t7 = a XOR t6;  
t14 = NOT t7;  
t8 = c OR t7;  
t11 = t3 XOR t7;  
g = t4 XOR t8;  
t12 = h AND t11;  
f = t10 XOR t12;  
e = t12 XOR t14
```

(e|f|g|h) = S1 (a|b|c|d)

```
t1 = NOT a;  
t2 = b XOR t1;  
t3 = a OR t2;  
t4 = d OR t2;  
t5 = c XOR t3;  
g = d XOR t5;  
t7 = b XOR t4;  
t8 = t2 XOR g;  
t9 = t5 AND t7;  
h = t8 XOR t9;  
t11 = t5 XOR t7;  
f = h XOR t11;  
t13 = t8 AND t11;  
e = t5 XOR t13
```

(e|f|g|h) = S2 (a|b|c|d)

```
t1 = NOT a;  
t2 = b XOR d;  
t3 = c AND t1;  
t13 = d OR t1;  
e = t2 XOR t3;  
t5 = c XOR t1;  
t6 = c XOR e;  
t7 = b AND t6;  
t10 = e OR t5;  
h = t5 XOR t7;  
t9 = d OR t7;  
t11 = t9 AND t10;
```

(e|f|g|h) = S3 (a|b|c|d)

```
t1 = a XOR c;  
t2 = d XOR t1;  
t3 = a AND t2;  
t4 = d XOR t3;  
t5 = b AND t4;  
g = t2 XOR t5;  
t7 = a OR g;  
t8 = b OR d;  
t11 = a OR d;  
t9 = t4 AND t7;  
f = t8 XOR t9;  
t12 = b XOR t11;
```

$e|f|g|h) = S2 (a|b|c|d)$ (окончание) $(e|f|g|h) = S3 (a|b|c|d)$ (окончание)

$t14 = t2 \text{ XOR } h;$

$g = a \text{ XOR } t11;$

$t15 = g \text{ XOR } t13;$

$f = t14 \text{ XOR } t15$

$t13 = g \text{ XOR } t9;$

$t15 = t3 \text{ XOR } t8;$

$h = t12 \text{ XOR } t13;$

$t16 = c \text{ AND } t15;$

$e = t12 \text{ XOR } t16$

$(e|f|g|h) = S4 (a|b|c|d)$

$t1 = a \text{ XOR } d;$

$t2 = d \text{ AND } t1;$

$t3 = c \text{ XOR } t2;$

$t4 = b \text{ OR } t3;$

$h = t1 \text{ XOR } t4;$

$t6 = \text{NOT } b;$

$t7 = t1 \text{ OR } t6;$

$e = t3 \text{ XOR } t7;$

$t9 = a \text{ AND } e;$

$t10 = t1 \text{ XOR } t6;$

$t11 = t4 \text{ AND } t10;$

$g = t9 \text{ XOR } t11;$

$t13 = a \text{ XOR } t3;$

$t14 = t10 \text{ AND } g;$

$f = t13 \text{ XOR } t14$

$(e|f|g|h) = S5 (a|b|c|d)$

$t1 = \text{NOT } a;$

$t2 = a \text{ XOR } b;$

$t3 = a \text{ XOR } d;$

$t4 = c \text{ XOR } t1;$

$t5 = t2 \text{ OR } t3;$

$e = t4 \text{ XOR } t5;$

$t7 = d \text{ AND } e;$

$t8 = t2 \text{ XOR } e;$

$t10 = t1 \text{ OR } e;$

$f = t7 \text{ XOR } t8;$

$t11 = t2 \text{ OR } t7;$

$t12 = t3 \text{ XOR } t10;$

$t14 = b \text{ XOR } t7;$

$g = t11 \text{ XOR } t12;$

$t15 = f \text{ AND } t12;$

$h = t14 \text{ XOR } t15$

$(e|f|g|h) = S6 (a|b|c|d)$

$t1 = \text{NOT } a;$

$t2 = a \text{ XOR } d;$

$t3 = b \text{ XOR } t2;$

$t4 = t1 \text{ OR } t2;$

$t5 = c \text{ XOR } t4;$

$f = b \text{ XOR } t5;$

$t13 = \text{NOT } t5;$

$t7 = t2 \text{ OR } f;$

$t8 = d \text{ XOR } t7;$

$t9 = t5 \text{ AND } t8;$

$(e|f|g|h) = S7 (a|b|c|d)$

$t1 = \text{NOT } c;$

$t2 = b \text{ XOR } c;$

$t3 = b \text{ OR } t1;$

$t4 = d \text{ XOR } t3;$

$t5 = a \text{ AND } t4;$

$t7 = a \text{ XOR } d;$

$h = t2 \text{ XOR } t5;$

$t8 = b \text{ XOR } t5;$

$t9 = t2 \text{ OR } t8;$

$t11 = d \text{ AND } t3;$

(e f g h) = S6 (a b c d) (окончание)	(e f g h) = S7 (a b c d) (окончание)
g = t3 XOR t9;	f = t7 XOR t9;
t11 = t5 XOR t8;	t12 = t5 XOR f;
e = g XOR t11;	t15 = t1 OR t4;
t14 = t3 AND t11;	t13 = h AND t12;
h = t13 XOR t14	g = t11 XOR t13;
	t16 = t12 XOR g;
	e = t15 XOR t16

Приложение 11

Большие простые числа для схемы Эль Гамаль

768-битное простое число

FFFFFFFF FFFFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A63A3620 FFFFFFFFFF FFFFFFFFFF

1024-битное простое число

FFFFFFFF FFFFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE65381
FFFFFFFF FFFFFFFFFF

1536-битное простое число

FFFFFFFF FFFFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE45B3D
C2007CB8 A163BF05 98DA4836 1C55D39A 69163FA8 FD24CF5F
83655D23 DCA3AD96 1C62F356 208552BB 9ED52907 7096966D
670C354E 4ABC9804 F1746C08 CA237327 FFFFFFFFFF FFFFFFFFFF

Приложение 12

Алфавит схемы транспортного кодирования Base64

Значение	Код	Значение	Код	Значение	Код	Значение	Код
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v	заполнитель =	
14	O	31	f	48	w		
15	P	32	g	49	x		
16	Q	33	h	50	y		

Приложение 13

Checklist настроек информационной безопасности. По материалам Krishni Naidu, опубликованным на сайте <http://www.sans.org/>. Авторы обращают внимание, что указанные настройки носят рекомендательный и в некоторых случаях "излишне осторожный" характер и могут не соответствовать требованиям конкретных производственных задач.

Операционная система Windows 2000

- ☐ Убедитесь, что последние обновления (service packs и hotfixes) вовремя получены от авторизованного источника и проверены перед запуском на рабочем оборудовании.
- ☐ Необходима процедура создания и безопасного сохранения диска аварийного восстановления ERD и копии реестра. Желательно сверять текущий реестр с контрольной копией на предмет несанкционированных изменений.
- ☐ Все диски должны иметь формат файловой системы NTFS (путь настройки — Settings | Control panel | Administrative tools | Computer management | storage | Disk management). Должны быть установлены соответствующие корректные разрешения на доступ к критическим данным, приложениям и папкам.
- ☐ Необходимо установить обновление Windows 2000 High encryption pack (c:\winnt\system32\scahnnel.dll — **Properties** | **Version** | **Description** = **TLS/SSL Security Provider**).
- ☐ Необходимо убедиться, что используется шаблон hisec*.inf template. Выбор шаблона зависит от конфигурации оборудования, например, для более старых программ на Terminal Server применяется notssid.inf. Стандартные шаблоны соответствуют следующим системным настройкам:
 - basicwk.inf // рабочая станция с настройками безопасности по умолчанию;
 - basicsv.inf // сервер с настройками безопасности по умолчанию;
 - basicdc.inf // контроллер домена с настройками безопасности по умолчанию;
 - compatws.inf // рабочая станция или сервер с настройками безопасности, обратно совместимыми с предыдущими версиями;

- notssid.inf // обратная совместимость с предыдущими версиями терминального сервера;
 - securews.inf // защищенная рабочая станция или сервер;
 - hisecws.inf // тщательно защищенная рабочая станция или сервер;
 - securedc.inf // защищенный контроллер домена;
 - hisecdc.inf // тщательно защищенный контроллер домена.
- ☐ Если в шаблоны внесены изменения, необходимо убедиться, что они санкционированы. Шаблоны должны сохраняться на CD или флорпи-диске. Необходимо выполнять процедуру регулярного анализа безопасности, выявления уязвимостей и мероприятий по улучшению безопасности.
- ☐ Запустить процедуру анализа безопасности. Для этого выполнить **Start Run "mmc" OK**. В открывшемся окне **Microsoft Management Console** в меню консоли выбрать **Add/Remove Snap-in....** В окне **Add/Remove Snap-in...** выбрать вкладку **Standalone**, нажать кнопку **Add** (Добавить), в окне **Add Standalone Snap-in**. В списке выбрать **Security Configuration and Analysis**, нажать кнопку **Add**, а затем кнопку **Close**. В окне **Add/Remove Snap-in...** нажать кнопку **OK**. В окне **Microsoft Management Console** слева в списке добавленных оснасток выбрать пункт **Open Data Base**, набрать имя базы. В окне **Import template**, выбрать **hisecdc.inf**. Щелкнуть правой кнопкой мыши по **Security Configuration and Analysis** и выбрать **Analyze now**. Указать место расположения отчета.
- ☐ Проанализировать результат отчета. Галочка напротив строки в отчете означает, что данный пункт соответствует требованиям шаблона, красный крест — что нет.
- ☐ Установки **Account Policies**.
- **Password Policy**.
 - Password History = 5 previous passwords.
 - Maximum password age = 30 days.
 - Minimum password age = 1 day.
 - Minimum password length = 8 characters.
 - Password must meet complexity requirements = enabled.
 - Store passwords using reversible encryption = disabled.
 - **Account Lockout**.
 - Account is locked out until administrator unlocks it.
 - Account lockout threshold = 3 invalid passwords
 - Reset account lockout counter after 30 minutes.

☐ Установки Local Policies.

- Audit Policy.
 - Audit account logon events – Success and Failure.
 - Audit account management – Success and Failure.
 - Audit directory service access – Success and Failure.
 - Audit logon events – Success and Failure.
 - Audit object access – Success and Failure.
 - Audit policy changes – Success and Failure.
 - Audit privilege use – Success and Failure.
 - Audit Process tracking – Success and Failure.
 - Audit System events – Success and Failure.
- User Rights.
 - Act as part of the operating system: Disable.
 - Add workstations to the domain: разрешено только для system administrator.
 - Backup files and directories: разрешено только для backup operators.
 - Change the system time: запрещено или разрешено только для system administrator.
 - Create a token object: разрешено только для system administrator.
 - Create permanent shared objects: разрешено только для system administrators.
 - Debug programs: разрешено только для developers и system administrators.
 - Deny logon as a batch job: Enable.
 - Deny logon as a service: Enable.
 - Force shutdown from a remote system: Disable.
 - Generate security audits: Enable.
 - Increase scheduling priority: разрешено только для system administrator.
 - Load and unload device drivers: разрешено только для system administrator.
 - Lock pages in memory: разрешено только для system administrator.
 - Logon as a batch job: Disable.
 - Logon as a service: Disable.
 - Manage auditing and security log: разрешено только для system administrator.

- Restore files and directories: разрешено только для system administrators и backup operators.
- Shutdown the system: разрешено только для system administrators.
- Synchronize directory service data: разрешено только для system administrator.
- Take ownership of files or other objects: разрешено только для system administrator.
- Security Options.
 - Allow server operators to schedule tasks (domain controllers only): разрешено только для server operators и system administrators.
 - Allow systems to be shutdown without having to logon: Disable.
 - Allowed to eject removable NTFS media: разрешено только для system administrator.
 - Amount of idle time allowed before disconnecting a session: Maximum 2 hours.
 - Audit the access of global system objects: Enable.
 - Audit use of backup and restore privilege: Enable.
 - Automatically log off users when log on time expires (local): Enable.
 - Clear virtual memory page file when system shuts down: Enable.
 - Don't display last username in logon screen: Enable.
 - LAN Manager authentication level: NTLMv2.
 - Message text for users attempting to log on: разрешено с юридическим предупреждением.
 - Message title for users attempting to log on: разрешено с юридическим предупреждением.
 - Number of previous logons to cache in case of domain controllers not available: 1.
 - Prevent system maintenance of computer account password: Enabled.
 - Prevent users from installing print drivers: Enabled.
 - Prompt user to change password before expiration: 14 days.
 - Recovery console: Allow automatic administrative logon: Disable.
 - Recovery console: Allow floppy copy and access to all drives and all folders: разрешено только для system administrator.
 - Rename administrator account: Enabled.
 - Rename guest account: Enabled.
 - Restrict CD ROM access to locally logged on user only: Enabled.

- Restrict floppy access to locally logged on user only: Enabled.
- Secure channel: Require strong (Windows 2000 or later) session key: Enabled.
- Shut down system immediately if unable to log security audits: Enabled.
- Strengthen default permissions of global system objects e.g. symbolic links: Enabled.

☐ Установки Event Log.

- Максимальный размер журнала — минимум 2 Мбайта.
- Ограничить доступ пользователей группы Guests к журналам.
- Метод очистки журнала при заполнении — только вручную.
- Автоматически останавливать компьютер при переполнении журнала событий.

☐ Установки System Services: должны присутствовать только сервисы, которые нужны для работы. Следующие сервисы должны быть отключены:

- Remote registry Service;
- Telnet;
- Alerter;
- Server;
- Messenger;
- SNMP.

☐ Мониторинг производительности должен быть активизирован минимум по следующим параметрам:

- Processor: %processor time, interrupts/sec, processor queue length;
- Disk: Pages/sec, %disk time, disk bytes/transfer, Current disk queue length.

Путь — **Settings | Control Panel | Administrative tools | Performance.**

Сообщения должны быть настроены для отправки администратору в случае появления неадекватной активности (по e-mail). Необходимо регулярно сверять текущие показания с контрольными замерами.

- ☐ Необходимо производить регулярное сканирование уязвимостей автоматизированными средствами.
- ☐ Для защищенной работы сервера IIS необходимо убрать неиспользуемые протоколы, особенно это касается NETBIOS. Если требуется доступ к серверу из локальной сети, то необходимо использование двух сетевых

карт: одной для Интернета (только протоколы TCP/IP и Client for MS networks), другой — для локальной сети с соответствующими протоколами. В TCP/IP-конфигурации должен быть блокирован протокол NETBIOS поверх TCP/IP. Необходимо установить последние обновления (hotfix). Ненужные соединения ODBC должны быть убраны. Следующие сервисы должны запускаться на сервере автоматически:

- DNS client;
- Event log;
- Logical disk manager;
- Network connections;
- Protected storage;
- RPC;
- SAM;
- WMI;
- WMI Driver extensions.

Следующие сервисы должны запускаться на сервере вручную:

- Logical disk manager administrative service;
- IIS Admin service;
- WWW publishing service.

Следующие сервисы не нужны для Web-сервера:

- Alerter;
- Clipbook server;
- Computer browser;
- DHCP client;
- Messenger;
- Netlogon;
- Network monitor agent;
- Simple TCP/IP services;
- Spooler;
- NetBIOS interface;
- TCP/IP NetBIOS helper;
- NWLink NetBIOS.

❑ Каждый раз, когда на сервере делается изменение конфигурации, оно должно быть протестировано. Веб-сайт должен сохраняться на отдельном

логическом диске от операционной системы. Следующие утилиты должны быть удалены, переименованы или иметь соответствующие разрешения:

- Xcopy.exe;
- At.exe;
- Regeditr.exe;
- Cacs.exe;
- Regedt32.exe;
- Cmd.exe;
- Regini.exe;
- Cscrpit.exe;
- Regsrv32.exe;
- Debug.exe;
- Rexec.exe;
- Edlin.exe;
- Rsh.exe;
- Finger.exe.

❑ Утилита syskey.exe должна использоваться для изменения места сохранения ключа шифрования базы данных SAM, а сам ключ должен храниться на нескольких дискетах в безопасном месте. Должны быть установлены следующие ключи реестра.

- **HKLM | System | CurrentControlSet | Control\LSA\Restrict anonymous** установлен на значение "1".
- **RestNullSessAccess** установлен на значение "1".
- **HKLM | System | CurrentControlSet\Services\TCPIP\Parameters | SynAttackProtect** установлен на значение "2".

❑ Если не используются в производственной деятельности, административные разделения ресурсов (sharing) должны быть убраны. Подсистемы имен OS/2 и POSIX должны быть отключены. Если не используется IPSec, должна быть активизирована фильтрация TCP/IP или Routing and Remote Access Service. Если используется аутентификация "Anonymous", то учетная запись IUSR_ComputerName должна быть заблокирована, а для анонимного доступа создана новая запись.

❑ Если используется основная аутентификация, то необходимо активизировать SSL. Если используется аутентификация с помощью сертификатов, то сертификат должен быть получен от доверенного СА. Настройки безопасности метабазы.

- Все корневые папки HTTP и FTP перенесены из тома %systemroot%
- Ключ реестра, определяющий расположение метабазы, защищен.

- Регистрируются все отказы в доступе к метабазе.
 - Файл `lissync.exe` удален из `%systemroot%\system32\ineterv`.
 - Разрешения полного контроля при доступе к метабазе должны быть только у администратора и системы.
 - Резервное копирование метабазы выполнено.
 - Регистрируются все отказы в доступе к папке резервной копии метабазы.
 - Разрешения полного контроля при доступе к папке резервной копии метабазы должны быть только у администратора и системы.
- ☐ Разрешения полного контроля при доступе к папке `\adminscripts` и `%systemroot%\system32\cscrip.exe` должны быть только у администратора
 - ☐ Регистрируются все отказы в доступе к папке `\adminscripts`. Файлы `metaedit.exe` и `metautil.dll` перенесены из `\program files` в `%systemroot%\system32\ineterv` и ярлык (shortcut) перенастроен.
 - ☐ Аудит должен быть установлен на все папки `scripts` и `bin`, и на каталоги, содержащие файлы, публично доступные в рамках Веб-сайта.
 - ☐ Учетная запись `Guest` заблокирована. Неиспользуемые типы файлов заблокированы. Неиспользуемые папки и файлы, особенно примеры (sample) и документы, по умолчанию удалены. На все папки установлены соответствующие разрешения `read/write/execute`. Просмотр (browsing) каталога должен быть заблокирован. Ограничения на адреса и доменные имена, настройки безопасности каталога должны соответствовать правилам и настройкам межсетевого экрана. Тайм-аут соединения для FTP должен быть 300 секунд. Соответствующие предупреждения на первой странице необходимо установить на `messages tab for ftp`. Соответствующие разрешения должны быть установлены на каталоги FTP.
 - ☐ Разрешения полного контроля при доступе к папке `winnt` должны быть только у администратора и системы.
 - ☐ Политики использования IPSec по умолчанию должны быть следующими.
 - Client (Respond Only): конфигурация должна позволять предоставление дополнительных мер безопасности, если другой системе это требуется.
 - Secure Server (Require Security): предназначена для сервера с критической информацией или шлюза VPN.
 - Server (Request Security): предназначена для системы, где предпочтительно иметь IPSec.

Если создается новая политика или используется имеющаяся, ее необходимо проанализировать следующим образом:

- методы IKE должны быть установлены с 3DES и Diffie-Hellman Medium 2;

- частота смены ключей должна соответствовать критичности данных;
 - установлены адекватные системные проверки для изменений политики;
 - правила для фильтрации удовлетворяют следующему:
 - правила применяются для всего трафика IP;
 - если используется аутентификация сертификатами, то только для доверенных СА;
 - адрес получения — это авторизованный адрес или подсеть;
 - зеркалирование установлено с тем, чтобы правила применялись для пакетов, посылаемых в обратном направлении;
 - протокол коммуникации — TCP — и соответствующий порт установлены;
 - фильтрующие действия должны требовать соответствующих механизмов безопасности;
 - должна быть активизирована безопасность согласований (negotiated security).
- ☐ Должен быть включен аудит успешных и неуспешных VPN-сессий.
- ☐ Для безопасности SNMP при сбое аутентификации должен быть установлен Send authentication trap. В SNMP-сообществе должны быть добавлены только авторизованные хосты. Должны быть установлены соответствующие разрешения на хосте для обработки SNMP-пакетов (**ComputerManagement | Services | SNMP**, включить опцию **AuthenticationTrap: SNMP | Action | Properties | Security**). Следующий ключ реестра должен иметь соответствующие разрешения:
HKLM\SYSTEM\CurrentControlSet\Services\SNMP\Parameters
- Разрешения: Administrators, System, Creator Owner:Full
- ☐ Доверительные отношения в междоменном сообществе должны быть адекватными.
- ☐ Критические данные должны быть зашифрованы на уровне файловой системы EFS. Агент восстановления должен быть указан в Encrypted Data Recovery Policy. Секретный ключ агента восстановления должен быть убран с портативных компьютеров и сервера, скопирован на дискеты и храниться в безопасном месте. Средство резервного копирования должно быть совместимо с Windows 2000, с тем чтобы зашифрованные данные не были утеряны.
- ☐ Для Kerberos-аутентификации должны быть установлены ограничения на регистрацию пользователя. Эта опция проверяет запросы на билеты по политике прав пользователя компьютера назначения. Максимальное

время жизни пользовательского или сервисного билета должно быть 8 часов. Максимальное время для обновления пользовательского билета — 5 дней. Максимальная чувствительность для синхронизации часов компьютера должна быть 5 минут.

- ❑ По безопасности утилит Resource Kit требования следующие.
 - Appsec.exe: только авторизованные пользователи имеют доступ к этому приложению по сети.
 - Flopplock.exe: должно быть активизировано на сервере.
 - Gpoutil.exe: должно быть запущено и предоставлять информацию об искаженных политиках.
 - Tracelog.exe: должно быть запущено.
- ❑ Для удаленного доступа необходимо использовать расширенные средства безопасности с одной из следующих установок:
 - CHAP;
 - SPAP;
 - MS-CHAPv2;
 - EAP-MD5;
 - EAP-TLS.

Опция удаленного доступа должна запрашивать имя и пароль пользователя.

- ❑ Для работы Терминального сервера требуется определить один из следующих уровней шифрования данных между клиентом и сервером.
 - Низкий — 40 — 56 бит. Только для критических данных типа пароля. Направление от клиента к серверу.
 - Средний — 40 — 56 бит. Шифрация в обоих направлениях.
 - Высокий — 128 бит. Шифрация в обоих направлениях.
 - Только авторизованные пользователи имеют право установить соединение. Перед регистрацией на сервере запрашивается пароль. В списке сервисов среды указаны корректные пути и имена файлов. Соответствующие разрешения установлены для удаленного доступа. Доступ "Guest" заблокирован. Для различных пользователей должны быть установлены соответствующие разрешения из следующего списка:
 - Query information;
 - Set information;
 - Reset;
 - Remote control;

- Logon;
 - Logoff;
 - Message;
 - Connect;
 - Disconnect;
 - Virtual channels.
- Установлен аудит на разрешения.

Межсетевой экран

- Необходимо просмотреть набор правил МСЭ, с тем чтобы убедиться, что порядок их следования соответствуют следующим требованиям.
 - Фильтры против подмены адреса (анти-спуфинг), т. е. блокировка частных адресов, запрет прохождения внутренних адресов наружу.
 - Разрешительные правила для пользователей (например, разрешение HTTP к публично доступному Веб-серверу).
 - Разрешительные правила для управления (например, SNMP-trap к серверу управления сетью).
 - Блокировка помех (например, сброс пакетов OSPF или HSRP).
 - Сообщение о запретах (выдача сообщения администратору о подозрительном трафике).
 - Фиксация запретов (запись трафика в регистрационный журнал для последующего анализа).

Обычно МСЭ работают на основе первого совпадения (т. е. срабатывает первое совпавшее с пакетом правило), поэтому указанный порядок важен для того, чтобы подозрительный трафик был остановлен, а не пропущен из-за неверного порядка правил.

- МСЭ прикладного уровня.

Необходимо убедиться, что администратор, используя журналы аудита МСЭ, отслеживает любую попытку обойти политику безопасности. Иногда МСЭ прикладного уровня предоставляют дополнительную функциональность интеграции с системой обнаружения атак (СОА). В этом случае необходимо убедиться, что для интеграции указан авторизованный хост с СОА. Должна присутствовать процедура обновления базы данных уязвимостей и сигнатур атак.

Если эти обновления загружаются с сайта поставщика, необходимо убедиться в доверенности этого сайта. Если обновления поступают по электронной почте, необходимо снабжать их электронно-цифровыми подписями (ЭЦП) и проверять корректность ЭЦП при получении.

Следующие команды должны быть заблокированы для протокола SMTP на МСЭ уровня приложения:

- EXPN (expand);
- VRFY (verify);
- DEBUG;
- WIZARD.

Следующая команда должна быть заблокирована в необходимых случаях для протокола FTP:

- PUT.

Следует просмотреть актуальность списка запрещенных URL, например, запрет адресов хакерских сайтов, сайтов только для взрослых и деструктивных сайтов.

Только авторизованные пользователи должны иметь право аутентификации на МСЭ.

☐ МСЭ с контролем соединения (stateful inspection).

Следует просмотреть таблицы состояний для обеспечения адекватности правил в терминах адресов отправителя и получателя, портов и таймаутов. Тайм-ауты должны быть установлены так, чтобы не дать злоумышленнику возможности успешно реализовать атаку.

- Если используется сервер фильтрации URL, то следует убедиться, что его определение сделано корректно. Если фильтрующий сервер является внешним по отношению к предприятию, то необходимо, чтобы источник был доверенным.
- Если список URL загружается из файла, необходимо предотвратить неавторизованную модификацию этого файла.

Необходимо корректировать трафик, содержащий ActiveX и Java-компоненты перед прохождением во внутреннюю сеть.

Если присутствует фильтрация по MAC-адресам, необходимо проверить фильтры на соответствие политике безопасности.

☐ Регистрационные журналы.

Регистрация должна быть включена и журналы должны анализироваться для выявления признаков, которые могут соответствовать атакам.

☐ Обновления (patches, updates, hotfixes).

Необходимо проверить и установить самые последние обновления, соответствующие МСЭ.

Если обновления загружаются с сайта поставщика, то этот сайт должен быть доверенным. Если обновления поступают по электронной почте, то должен применяться механизм установки и проверки ЭЦП.

❑ Размещение демилитаризованной зоны (ДМЗ).

Если используется ДМЗ в конфигурации Веб-сервер между двумя МСЭ, то каждый МСЭ должен быть оснащен двумя сетевыми интерфейсами. Как дополнительное средство защиты можно порекомендовать, чтобы эти МСЭ были различных типов, чтобы потенциальному злоумышленнику пришлось собирать в два раза больше информации о конструкции и уязвимостях МСЭ. Оба МСЭ должны иметь различные наборы правил в зависимости от расположения по отношению к Веб-серверу и внутренней сети.

❑ Тестирование и оценка уязвимостей.

Необходимо разработать процедуру выявления открытых TCP-портов (например, с помощью утилиты nmap) и закрытия неиспользуемых портов. Необходимо протестировать набор правил для того, чтобы не создать ситуацию случайного отказа в обслуживании или, наоборот, не оставить невыявленных уязвимостей.

❑ Соответствие политике безопасности.

Следует убедиться, что набор правил соответствует политике безопасности предприятия.

❑ Требуется убедиться, что следующие подменяемые, приватные (по RFC-1918) и запрещенные адреса блокированы.

- Стандартные немаршрутизируемые
 - 255.255.255.255
 - 127.0.0.0
- Приватные
 - 10.0.0.0 – 10.255.255.255
 - 172.16.0.0 – 172.31.255.255
 - 192.168.0.0 – 192.168.255.255
- Зарезервированные
 - 240.0.0.0 – 240.255.255.255
- Запрещенные
 - 0.0.0.0
- UDP echo и ICMP broadcast (по RFC-2644)

Трафик с указанных адресов не должен передаваться с сетевого интерфейса.

- ❑ Следует убедиться, что блокирована маршрутизация от источника loose source routing и strict source routing (lsrr и ssrr).
- ❑ Ограничения по портам.

Следующие порты должны быть заблокированы (формат — сервис, тип порта, номер порта).

- Перенос зоны DNS (кроме как от внешних вторичных DNS серверов), TCP 53.
- Демон TFTP, UDP 69.
- Link, TCP 87.
- SUN RPC, TCP и UDP 111.
- BSD UNIX, TCP 512 — 514.
- LPD, TCP 515.
- UUCPD, TCP 540.
- Open Windows, TCP и UDP 2000.
- NFS, TCP и UDP 2049.
- X-Windows, TCP и UDP 6000 — 6255.
- "Сервисы на низких портах", TCP и UDP 20 и ниже.
- FTP, TCP 21.
- SSH, TCP 22.
- Telnet, TCP 23.
- SMTP (кроме внешних пересылок почты), TCP 25.
- NTP, TCP и UDP 37.
- Finger, TCP 79.
- HTTP (кроме как к внешним Веб-серверам), TCP 80.
- POP, TCP 109 и TCP 110.
- NNTP, TCP 119.
- NTP, TCP 123.
- NetBIOS для Windows NT, TCP и UDP 135.
- NetBIOS для Windows NT, UDP 137 и UDP 138
- NetBIOS, TCP 139.
- IMAP, TCP 143.
- SNMP, UDP и TCP 161 и 162.
- BGP, TCP 179.
- LDAP, TCP и UDP 389.

- SSL (кроме как к внешним Веб-серверам), TCP 443.
 - NetBIOS в Win2k, TCP и UDP 445.
 - Syslog, UDP 514.
 - SOCKS, TCP 1080.
 - Порт Cisco AUX, TCP 2001.
 - Порт Cisco AUX (stream), TCP 4001.
 - Lockd (уязвимость Linux DoS), TCP и UDP 4045.
 - Порт Cisco AUX (binary), TCP 6001.
 - Общие порты HTTP высокого порядка, TCP 8000, 8080, 8888.
 - Порты троянских программ, имеющих повышенное распространение в данный момент времени.
- ☐ Удаленный доступ.
- Если используется удаленный доступ, необходимо убедиться, что используется сервис SSH (порт 22) вместо сервиса Telnet.
- ☐ Передача файлов.
- Если необходимо использование сервиса FTP, следует убедиться, что сервер FTP размещен на другой подсети, вне внутренней сети.
- ☐ Почтовый трафик.
- Следует определиться, какой почтовый протокол используется, и блокировать исходящий почтовый трафик, кроме как от внутреннего почтового сервера.
- ☐ ICMP (ICMP 8, 11, 3).
- Необходимо убедиться, что блокируются пакеты ICMP echo request и reply.
- Необходимо убедиться, что блокируются исходящие сообщения time exceeded и unreachable.
- ☐ Переадресация/сокрытие IP.
- Необходимо установить опцию сокрытия внутренних адресов от внешней недоверенной сети.

Маршрутизатор Cisco

- ☐ Вся работа по поддержке маршрутизатора необходимо производить по сессии с локальным входом. Должны быть отключены входящий Telnet и Telnet-демон. Если все же необходимо использовать Telnet для технической поддержки, установите доступ только для отдельных станций и только со стороны внутренней сети. Все сервисы поддержки с доступом

со стороны внешней сети должны быть отключены или ограничены. Везде где возможно следует использовать пароли. Для всех типов паролей должна использоваться опция `service password encryption` с тем, чтобы при отображении паролей выводилась только их MD5-сумма. Пароль EXEC должен быть добавлен портам AUX и Console. Протоколы динамической маршрутизации RIP и OSPF должны быть заблокированы на входах в сеть, CDP блокирован на всех сетевых интерфейсах. Баннер на загрузке должен быть снабжен соответствующим сообщением, но при этом не содержать информации о маршрутизаторе, его имени, модели, программном обеспечении, владельце. Если возможно, сервис SNMP должен быть отключен.

- Каждый модем или другое сетевое устройство, имеющее доступ к консольному порту, должны подключаться с помощью соответствующего механизма безопасности. Для модема необходимо установить пароль для входящих звонков удаленного доступа. Для каждой асинхронной или модемной линии должна быть применена команда `transport input none`, не дающая возможность получать соединение от сетевых пользователей. Один и тот же модем не должен иметь одновременно возможности dial-in и dial-out, обратное Telnet-соединение (`reverse telnet connection`) должно быть заблокировано для dial-in линий. Для терминальных VTU-линий (VTU lines) необходимо установить поддержку только тех протоколов, которые действительно необходимы. Если присутствует поддержка шифрования, то следует использовать только протокол SSH. Команда `ip access class` должна быть использована для ограничения адресов, с которых VTU поддерживает соединения. Один `ip access-class` для VTU должен быть ограничен только для одной административной станции для предотвращения DoS атаки. Тайм-ауты VTU должны быть сконфигурированы командой `exec-timeout`. На входящие соединения должен быть установлена защита командой `tsp-keepalives-in`. Все не-IP протоколы для удаленного доступа должны быть отключены, а для удаленного соединения с маршрутизатором использоваться только IPSec. Для маршрутизаторов, поддерживающих CEF (Cisco Express Forwarding), должна быть установлена проверка RPF (Reverse path forwarding) с помощью команды `ip verify unicast rpf`.
- Все непривилегированные пароли должны сохраняться на серверах (TACACS или RADIUS). Только авторизованные адреса TACACS-демонов должны быть указаны в команде `tacacs-server host`. Если используется команда `enable use-tacacs`, она должна использоваться с расширенным TACACS, в противном случае любой владелец зарегистрированного идентификатора/пароля получит доступ к привилегированному режиму EXEC. Должна быть активизирована команда `login tacacs`. Команду `tacacs-server notify` необходимо сконфигурировать для

отправки сообщений в случаях, когда пользователь устанавливает TCP соединение, вводит команду `enable`, завершает работу. Команда `tacacs-server attempts` должна быть сконфигурирована для трех попыток входа с линии выделенной для TACACS. Для включения расширенного режима TACACS используйте команду `tacacs-server extended`.

- ☐ Следующие сервисы должны быть заблокированы (формат – сервис, тип порта, номер порта).
- Перенос зоны DNS (кроме как от внешних вторичных DNS серверов), TCP 53.
 - Демон TFTP, UDP 69.
 - Link, TCP 87.
 - SUN RPC, TCP и UDP 111.
 - BSD UNIX, TCP 512 – 514.
 - LPD, TCP 515.
 - UUCPD, TCP 540.
 - Open Windows, TCP и UDP 2000.
 - NFS, TCP и UDP 2049.
 - X-Windows, TCP и UDP 6000 – 6255
 - Сервисы на низких портах, TCP и UDP 20 и ниже.
 - FTP, TCP 21.
 - SSH, TCP 22.
 - Telnet, TCP 23.
 - SMTP (кроме внешних пересылок почты), TCP 25.
 - NTP, TCP и UDP 37.
 - Finger, TCP 79.
 - HTTP (кроме как к внешним Веб-серверам), TCP 80.
 - POP, TCP 109 и TCP 110.
 - NNTP, TCP 119.
 - NTP, TCP 123.
 - NetBIOS для Windows NT, TCP и UDP 135.
 - NetBIOS для Windows NT, UDP 137 и UDP 138.
 - NetBIOS, TCP 139.
 - IMAP, TCP 143.
 - SNMP, UDP и TCP 161 и 162.

- BGP, TCP 179.
 - LDAP, TCP и UDP 389.
 - SSL (кроме как к внешним Веб-серверам), TCP 443.
 - NetBIOS в Win2k, TCP и UDP 445.
 - Syslog, UDP 514.
 - SOCKS, TCP 1080.
 - Порт Cisco AUX, TCP 2001.
 - Порт Cisco AUX (stream), TCP 4001.
 - Lockd (уязвимость Linux DoS), TCP и UDP 4045.
 - Порт Cisco AUX (binary), TCP 6001.
 - Общие порты HTTP высокого порядка, TCP 8000, TCP 8080, TCP 8888.
- ❑ Следующие типы ICMP трафика блокированы на внешнем интерфейсе: входящие echo request (ping и traceroute), исходящие echo reply, time exceeded, сообщения unreachable, редиректы ICMP.
- ❑ На интерфейсе маршрутизатора должны быть заблокированы и регистрироваться:
- входящие пакеты с адресом отправителя из приватного (немаршрутизируемого) диапазона адресов;
 - входящие пакеты с адресом отправителя вида 127.0.0.X;
 - входящие пакеты с зарезервированным адресом отправителя;
 - входящие пакеты с адресом отправителя из диапазона адресов, принадлежащим внутренней сети предприятия;
 - исходящие пакеты с адресом отправителя из приватного (немаршрутизируемого) диапазона адресов;
 - исходящие пакеты с адресом отправителя вида 127.0.0.X.
 - исходящие пакеты с зарезервированным адресом отправителя.
- ❑ Маршрутизация от источника должна быть заблокирована. Запросы на широковещание IP (IP directed broadcast) на всех интерфейсах должны блокироваться и регистрироваться (no ip directed-broadcast).
- ❑ Сервис синхронизации времени NTP должен позволять обновления только от внутренних NTP-серверов и при этом быть заблокирован на внешнем интерфейсе для входящих и исходящих пакетов.

- ❑ Должна проводиться регулярная процедура проверки неиспользуемых портов. Возможно использование сниффера для проверки пакетов, проходящих через листы доступа ACL.
- ❑ Необходимо использовать средства обнаружения нестандартной активности (такие как Jinao, fdget.c и Agilent Advisor).
- ❑ Уязвимости должны регулярно выявляться сканерами, необходима процедура устранения уязвимостей.
- ❑ При необходимости протокола SNMP следует использовать версию не ниже 2. Имена community string, такие как *public* и *private*, не следует использовать, пароли для community string должны регулярно изменяться. Стойкость community string должна регулярно проверяться атаками грубой силы (brute force attack), такими как Solarwinds. Только авторизованные хосты управления должны иметь доступ к маршрутизаторам с SNMP. Извещения SNMP-trap должны быть сконфигурированы для рассылки только на авторизованные компьютеры, в том числе и в случае, если произошел сбой аутентификации community string.
- ❑ Выполнение команд на маршрутизаторе должно быть ограничено соответствующим привилегированным уровнем.
- ❑ Должна быть активизирована команда `ip tcp intercept list`.
- ❑ Необходимо использовать сетевые системы обнаружения атак (COA), такие как Cisco Secure IDS, Realsure и т.д.
- ❑ Следует убедиться, что мощность процессора маршрутизатора удовлетворяет производственным задачам.
- ❑ Необходимо установить регистрацию событий, лучше всего на отдельный syslog-сервер. Если регистрация на маршрутизаторе невозможна, следует использовать регистрацию событий сниффера или COA. Регистрируемые события должны содержать отметку времени (timestamps log datetime msec). Должен быть активизирован сервис `access list logging`.
- ❑ Возвратный трафик SSH должен быть разрешен.
- ❑ Если используется FTP, следует разрешить пассивный FTP-трафик.
- ❑ Для СВАС (Context based access control) использовать команду `show ip inspection interfaces` для отображения правил инспекции и листов доступа, чтобы убедиться что:
 - присутствует инспекция Java;
 - присутствует инспекция Unix R команд, FTP, TFTP;
 - присутствует инспекция TCP/UDP;

- правила инспекции установлены для соответствующих интерфейсов;
 - листы доступа принимают трафик от дружественных сайтов и блокируют трафик от враждебных сайтов.
- ❑ Должен быть активизирован `ip inspect audit trail`.
 - ❑ Средства, использующие фильтры (`npair`), должны быть протестированы перед запуском на рабочем оборудовании.
 - ❑ Если необходимо использование DNS, DNS-трафик должен быть разрешен только к DNS-серверу. От экранируемой сети (`screened subnet`) должны исходить только DNS-запросы.
 - ❑ Если необходим трафик HTTP, он должен быть разрешен только к Веб-серверу.
 - ❑ В случае применения протокола HTTP для управления устройством доступ должен быть ограничен только авторизованными хостами, командой `ip http access-class`. TACACS+ или RADIUS сервера должны использоваться для аутентификации или интерактивной регистрации.
 - ❑ Для управляющего или интерактивного доступа из недоверенной сети должны использоваться такие протоколы, как SSH или Kerberized Telnet либо протокол IPSec для управляющего трафика. Также возможно использовать системы одноразовых паролей S/Key или OPIE с TACACS+ или RADIUS-серверами для управления привилегированным доступом. Для случаев DoS-атак должен быть альтернативный канал управления (например, через модем).
 - ❑ Должна быть заказана процедура получения и тестирования обновления программных продуктов Cisco.
 - ❑ Баннер аутентификации должен быть сконфигурирован командой `aaa authentication banner`. Сервер сетевого доступа должен запрашивать авторизационную информацию перед тем, как разрешить пользователю обратную Telnet-сессию (команда `aaa authorisation reverse-access {radius | tacacs+}`). Необходимо активизировать учет для регистрации `network`, `EXEC`, `commands`, `connection` и `system`. Информация учета должна регулярно анализироваться командой `show accounting`.
 - ❑ Перед получением обновлений таблиц маршрутизации соседние маршрутизаторы должны быть аутентифицированы на основе алгоритма MD5. К ключам при пересылке и хранении следует применить механизмы безопасности, предотвращающие их раскрытие и модификацию.
 - ❑ Необходимо проанализировать политику обмена ключами (команда `show crypto isakmp policy`). Для коммуникаций повышенного риска требуется установить более сильную конфигурацию.
- Хэш-алгоритм SHA-1.

- Способ аутентификации — RSA сигнатуры.
 - Групповой идентификатор Диффи-Хеллмана — 1024 бит.
 - Время жизни ассоциаций безопасности — 1 день (86 400 секунд).
- ❑ Для сокрытия внутренних адресов использовать команду `no ipsec transform-group`.
 - ❑ Для проверки использования криптографического алгоритма (типа DES) примените команду `show crypto cisco algorithms`. 40-битная версия не должна использоваться, если это возможно. Необходимо сделать резервную копию криптографической конфигурации.
 - ❑ CRL optional должен быть установлен в значение "no". Для получения сертификатов опция CA должна быть доступна. Для загрузки последних списков отзыва сертификатов следует регулярно использовать команду `crypto ca crl request`.
 - ❑ После внесения изменений конфигурация должна быть сохранена командой `copy running-config startup-config`.

Приложение 14

Примеры форматов заявок

ЗАЯВКА 01

Исх. _____

Вх. _____

**Начальнику Службы
информационных технологий**

Прошу Вас обеспечить установку/перерегистрацию персонального компьютера

(ненужное зачеркнуть)

в _____

(полное наименование подразделения)

с _____

*только для перерегистрации — (Ф. И. О.
и должность предыдущего ответственного)*

по причине _____

только для перерегистрации — (причина перерегистрации)

и зарегистрировать на нем сотрудников:

1. _____

(Ф. И. О. и должность)

— ответственный за ПК

2. _____

(Ф. И. О. и должность)

Подтверждаю, что перечисленные сотрудники подготовлены для работы на персональном компьютере в информационной сети АБВ. Контактное лицо:

(Ф. И. О.)

телефон _____

(Номер телефона)

с _____

только для переноса — (текущее место расположения ПК)

На _____

только для переноса — (требуемое место расположения ПК)

Наиболее удобное время для работ _____

(время в течение рабочего дня)

Контактное лицо:

телефон _____

(Ф. И. О.)

(Номер телефона)

(Ф. И. О. начальника)

(Подпись)

(Поля ниже заполняются службой информационных технологий)

Поручения Начальника Службы информационных технологий
(_____) ____/____/____/

Службе _____
(Ф. И. О., выполнение, дата, подпись)

Службе _____
(Ф. И. О., выполнение, дата, подпись)

(Примечания)

Дата завершения "____"

Принято на хранение _____

(Подпись)

ЗАЯВКА 03

Исх. _____

Вх. _____

начальнику Службы информационных технологий

Прошу Вас обеспечить работу пользователей

1. _____

(Ф. И. О. и должность)

2. _____

(Ф. И. О. и должность)

(полное наименование подразделения)

В

1. _____

(Название сервиса)

(Обоснование необходимости)

2. _____

(Название сервиса)

(Обоснование необходимости)

Подтверждаю, что указанные сотрудники подготовлены для работы с указанной программой/ сервисом и будут использовать их только в служебных целях. Контактное лицо:

телефон _____

(Ф. И. О.)

(Номер телефона)

(Ф. И. О. начальника)

(Подпись)

— поля ниже заполняются службой информационных технологий —

Поручения Начальника Службы информационных технологий
() _/_/_/

Службе _____

(Ф. И. О., выполнение, дата, подпись)

Службе _____

(Ф. И. О., выполнение, дата, подпись)

(примечания)

Дата завершения "___"

Принято на хранение _____

(Подпись)

ЗАЯВКА 06

Исх. _____

Вх. _____

начальнику Службы информационных технологий

Прошу Вас включить в план автоматизации создание/модификацию средства для
решения производственной задачи (ненужное зачеркнуть)

в _____

(полное наименование подразделения)

Сущность задачи описана в Приложении на _____ листах

Планируемые результаты _____

(краткое описание результата, полученного при создании/модификации)

Контактное лицо:

телефон _____

(Ф. И. О.)

(Номер телефона)

Подтверждаю, что указанный сотрудник уполномочен для постановки задачи и имеет необходимую квалификацию.

(Ф. И. О. начальника)

(Подпись)

— поля ниже заполняются службой информационных технологий —

Поручения Начальника Службы информационных технологий
(_____) ____/____/____/

Службе _____
(Ф. И. О., выполнение, дата, подпись)

Службе _____
(Ф. И. О., выполнение, дата, подпись)

(примечания)

Дата завершения “ ____ ”

Принято на хранение _____
(Подпись)

ЗАЯВКА 07

Исх. _____

Вх. _____

начальнику Службы информационных технологий

Прошу Вас выполнить следующие работы

(краткое наименование работы)

в _____

(полное наименование подразделения)

Сущность работ состоит в:

Причина, по которой необходимо проведение работ

(краткое обоснование необходимости работ)

Со стороны нашего подразделения работы будут координироваться сотрудником

Контактное лицо:

телефон _____

(Ф. И. О.)

(Номер телефона)

Подтверждаю, что указанный сотрудник уполномочен для координации работ и имеет необходимую квалификацию.

(Ф. И. О. начальника)

(Подпись)

— поля ниже заполняются службой информационных технологий —

Поручения Начальника Службы информационных технологий
_____ (_____) ____/____/____

Службе _____
(Ф. И. О., выполнение, дата, подпись)

Службе _____
(Ф. И. О., выполнение, дата, подпись)

(примечания)

Дата завершения “ ____ ”

Принято на хранение _____

(Подпись)

Примеры форматов приложений к заявкам

Приложение PR-1

Приложение к заявке исх _____ от “ ____ ” _____

Для создания / модификации отчета

(ненужное зачеркнуть)

(для модификации указать наименование программного обеспечения)

(указать наименование создаваемого или модифицируемого отчета)

Данные по новому отчету

Периодичность отчета _____

(ежедневный, ежемесячный, ежеквартальный и т. д.)

Внешний вид

(указать графы, подграфы, их наименования, размеры столбцов и строк и пр.)

Правила формирования

(указать по каким правилам формируются данные для каждого столбца в соответствующих строках)

Пример отчета

(привести пример отчета или его части с пояснениями)

Приложение PR-2

Приложение к заявке исх _____ от “ ____ ” _____

Для создания / модификации автоматизированного расчета

(ненужное зачеркнуть)

(для модификации указать наименование программного обеспечения)

(наименование расчетного процесса)

Данные по расчету

Источник данных

(указать, какие данные служат основой расчета)

Формулы и правила расчета

(указать, как производится расчет или его части)

Форма представления расчета

(указать, в какой форме представить результаты расчета)

Пример расчета

(привести пример расчета с пояснениями)

Приложение PR-3

Приложение к заявке исх _____ от “ ____ ” _____

Для внесения изменения в систему справочников для программного обеспечения

(указать наименование программного обеспечения)

(указать наименование существующего или нового справочника)

Данные по справочнику

Наименования полей справочника

(указать наименования полей нового или обновленного справочника)

Форматы полей справочника

(указать, что служит основой для заполнения справочника)

Контроль соответствия

(если необходимо контролировать выбранные из справочника данные на соответствие другим данным или условиям — указать)

Пример справочника

(указать пример справочника или его части с пояснениями)

Приложение PR-4

Приложение к заявке исх _____ от “ ____ ” _____

Для внесения изменения в систему логического контроля для программного обеспечения

(указать наименование программного обеспечения)

Данные по контролю

Источники данных для контроля

(указать, какие данные необходимо контролировать)

Формула контроля

(указать какие условия необходимо контролировать)

Результат контроля

(указать, что должно происходить в результате наступления контролируемого события)

Пример контроля

(указать пример контролируемого события с результатом)

Приложение PR-5

Приложение к заявке исх _____ от “ ____ ” _____

Для создания / модификации

(ненужное зачеркнуть)

(указать задачу для создания/модификации)

(для модификации указать наименование программного обеспечения)

Данные по задаче

Источники данных

(указать, что служит основой для выполнения задачи)

Описание процесса

(указать, что будет выполнять задача)

Результат

(указать, что будет получено в результате выполнения задачи)

Пример задачи

(привести пример выполнения задачи с пояснениями)

Перечень документов и литературы

- [Atkins] Atkins Derek, Buis Paul, Hare Chris, Kelley Robert, Nachenberg Carey, Nelson Anthony B., Phillips Paul, Ritchey Tim, Sheldon Tom, Snyder Joel. Internet Security Professional Reference. — Second Edition / New Riders Publishing. ISBN: 156205760x.
- [Aziz] Aziz A., Patterson M. Design and Implementation of SKIP. <http://skip.incog.com/inet-95.ps>.
- [Black] Блэк У. Интернет: протоколы безопасности. Учебный курс / Пер. с англ. — СПб.: Питер, 2001.
- [Blake] Blake Sonya Q. The Clark-Wilson Security Model, May 2000, <http://rr.sans.org/policy/clark-wilson.php>
- [CC] Common Criteria for Information Technology Security Evaluation: Version 2.1. CCIMB-99-031. August 1999.
- [Cisco] The Science of Intrusion Detection System, Attack Identification. Cisco Systems, Inc., 2002.
- [Diffie] Диффи У. Первые десять лет криптографии с открытым ключом / Пер. с англ. — ТИИЭР, Т. 76, М.: "Мир", 1988.
- [DOD1] Trusted Computer Systems Evaluation Criteria. Department of Defense standard 5200.28-STD. December 1985.
- [DOD2] Password Management Guide. Department of Defense standard CSC-STD-002-85. April 1985.
- [DT] Automated information systems security policy CIS HB 1400-05. Department of the Treasury, United States Customs Service.
- [Feistel] Feistel H. Block Cipher Cryptographic System. 1974
- [FIPS-46] Federal Information Processing Standards. FIP PUB 46. Data Encryption Standard (DES). 1977.

- [FIPS-81] Federal Information Processing Standards. FIPS PUB 81. DES Modes of Operation. 1980.
- [GASSP] Generally Accepted System Security Principles. The International Information Security Foundation. Version 2.0. June 1999.
- [Harris] Harris Shon, McGraw Hill. CISSP Certification. All in one / Osborne, 2002. ISBN: 0-072-19354-9.
- [HISM] Krause Micki, Tipton Harold F. Handbook of Information Security Management. Auerbach Publications. ISBN: 0-849-39947-5.
- [ISA] Northcutt Stephen, Cooper Mark, Fearnow Matt, Frederick Karen. Intrusion Signatures and Analysis. New Riders, 2001. ISBN: 0-735-71063-5.
- [ITSC] The IT Security Cookbook,
<http://security.tsu.ru/info/misc/boran>
- [ITSS] Information Technology Security Standard. Financial and Information Management Branch / Treasury Board Secretariat, Canada Communication Group.
- [Kurtz] Kurtz Ronald L., Vines Russell Dean. The CISSP Prep Guide: Mastering the Ten Domains of Computer Security. — Wiley Computer Publishing, 2001. ISBN: 0-471-41356-9.
- [Menezes] Menezes A., Oorschot P. van, Vanstone S. Handbook of Applied Cryptography. — CRC Press, 1996.
- [Messay] Мессеи Дж. Л. Введение в современную криптологию. ТИИЭР. Т. 76. № 5 (Май 1988). М.: Мир, 1988.
- [MS] Maximum Security: A Hacker's Guide to Protecting Your Internet Site and Network. Macmillan Computer Publishing.
- [NCSC] Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria (NCSC-TG-005). National Computer Security Center. July 1987.
- [NIST1] Information Technology Security Training Requirements : A Role- and Performance-Based Model: National Institute of Standards and Technology. NIST. Special Publication 800-16.
- [NIST2] Establishing a Computer Security Incident Response Capability (CSIRC): NIST Special Publication 800-3 / Wack John P. Computer Systems Laboratory. National Institute of Standards and Technology. NIST.

- [Pernul] Pernul Gunther. Database security Institut für Angewandte Informatik und Informationssysteme Abteilung für Information Engineering. Universität Wien. Austria, Vienna.
- [RFC-1319] Kaliski B. The MD2 Message-Digest Algorithm. April 1992.
- [RFC-1320] Rivest R. The MD4 Message-Digest Algorithm. April 1992.
- [RFC-1321] Rivest R. The MD5 Message-Digest Algorithm. April 1992.
- [RFC-1334] PPP Authentication Protocols. Network Working Group, 1992.
- [RFC-1421] Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures. Network Working Group, 1993.
- [RFC-1492] An Access Control Protocol, Sometimes Called TACACS. Network Working Group, 1993
- [RFC-1510] The Kerberos Network Authentication Service (V5). Network Working Group, 1993.
- [RFC-1661] The Point-to-Point Protocol (PPP). July 1994.
- [RFC-1760] The S/KEY One-Time Password System. Network Working Group, 1995.
- [RFC-1847] Security Multiparts for MIME: Multipart / Signed and Multipart / Encrypted Galvin J., Murphy S., Crocker S., Freed N. October 1995.
- [RFC-1928] SOCKS Protocol Version 5. Network Working Group, 1996.
- [RFC-1994] PPP Challenge Handshake Authentication Protocol (CHAP). Network Working Group, 1996.
- [RFC-2104] Krawczyk H., Bellare M., Canetti R. HMAC: Keyed-Hashing for Message Authentication. February 1997.
- [RFC-2144] Adams C. The CAST-128 Encryption Algorithm. May 1997.
- [RFC-2196] Fraser B. Site Security Handbook. Network Working Group, 1997
- [RFC-2246] The TLS Protocol. Network Working Group, 1999.
- [RFC-2268] Rivest R. A Description of the RC2(R) Encryption Algorithm. January 1998.
- [RFC-2284] PPP Extensible Authentication Protocol (EAP). Network Working Group, 1998.
- [RFC-2341] Cisco Layer Two Forwarding (Protocol) "L2F". Network Working Group, 1998.
- [RFC-2350] Brownlee N., Guttman E. Expectations for Computer Security Incident Response. — Network Working Group. 1998.

- [RFC-2401] Security Architecture for the Internet Protocol. — Network Working Group, 1998
- [RFC-2412] Orman H. The OAKLEY Key Determination Protocol. — November 1998.
- [RFC-2459] Internet X.509: Public Key Infrastructure, Certificate and CRL Profile. Network Working Group, 1999.
- [RFC-2612] Adams C., Gilchrist J. The CAST-256 Encryption Algorithm. June 1999.
- [RFC-2633] S/MIME. Version 3 Message Specification. Network Working Group, 1999.
- [RFC-2637] Point-to-Point Tunneling Protocol (PPTP). Network Working Group, 1999.
- [RFC-2661] Layer Two Tunneling Protocol "L2TP". Network Working Group, 1999.
- [RFC-2784] Generic Routing Encapsulation (GRE). Network Working Group, 2000.
- [RFC-2865] Remote Authentication Dial In User Service (RADIUS). Network Working Group, 2000.
- [RFC-2866] Rigney C. RADIUS Accounting. Livingston. June 2000.
- [Schneier] Schneier Bruce. Applied Cryptography: Protocols, Algorithms, and Source Code in C (Second Edition). John Wiley & Sons, Inc., 1996.
- [Shamir] Shamir Adi. Eran Tromer Factoring Large Numbers with the TWIRL Device. January 2003.
- [SPMH] Security Policy-Manager's Handbook. Information, Communication and Security Policy, Treasury Board of Canada, Secretariat.
- [Алферов] Алферов А. П. и др. Основы криптографии: Учебное пособие. — М.: Гелиос АРВ, 2001.
- [Анин] Анин Б. Ю. Защита компьютерной информации. — СПб.: БХВ—Петербург, 2000.
- [Анохин] Анохин М. И., Варновский Н. П., Сидельников В. М., Ященко В. В. Криптография в банковском деле.
- [Беляев1] Беляев А. Идентификация удаленного пользователя без раскрытия пароля // Программист, 2001, № 5—6.
- [Беляев2] Беляев А. Алгоритмы сжатия: семейство LZ // Программист, 2001, № 7.

- [Беляев3] Беляев А. Стеганография: скрытие информации // Программист, 2001, № 10.
- [Беляев4] Беляев А. В., Панасенко С. П., Петренко С. А. Перспективы прикладной криптографии // Защита информации. Конфидент. 2001, № 6.
- [ГОСТ Р 34.10–2001] ГОСТ Р 34.10–2001. Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной подписи на базе асимметричного криптографического алгоритма.
- [ГОСТ Р 34.10–94] ГОСТ Р 34.10–94. Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной подписи на базе асимметричного криптографического алгоритма.
- [ГОСТ Р 34.11–94] ГОСТ Р 34.11–94. Информационная технология. Криптографическая защита информации. Функция хэширования.
- [ГОСТ 28147–89] ГОСТ 28147–89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования.
- [ГТК1] Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации: Руководящий документ / Государственная техническая комиссия при Президенте Российской Федерации.
- [ГТК2] Средства вычислительной техники. Межсетевые экраны. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации: Руководящий документ / Государственная техническая комиссия при Президенте Российской Федерации.
- [Жельников] Жельников В. Криптография от папируса до компьютера. М.: АБФ, 1996.
- [Каторин] Каторин Ю. Ф. и др. Большая энциклопедия промышленного шпионажа. – СПб.: ООО Издательство Полигон, 2000.
- [Конеев1] Конеев И. Р. Методика тестирования программного обеспечения на соответствие 2000 году // Мир ПК, 1999, № 4.

- [Конеев2] Конеев И. Р. Внутреннее расследование // Computerworld, 1999, № 44.
- [Конеев3] Конеев И. Р. Классификатор данных с точки зрения безопасности // Директору ИС, 2000, № 7—8.
- [Конеев4] Конеев И. Р. Правовые аспекты систем удаленного обслуживания клиентов // Директору ИС, 2001, № 6.
- [Конеев5] Конеев И. Р. Требования по информационной безопасности автоматизированных банковских систем // Директору ИС, 2002, № 4.
- [Лукацкий1] Лукацкий А. В. Выявление уязвимостей компьютерных сетей
<http://www.citforum.ru/internet/securities/vulnerability.shtml>
- [Лукацкий2] Лукацкий А. В. Обнаружение атак. — СПб.: БХВ-Петербург, 2001.
- [Мамаев] Мамаев М., Петренко С. Технологии защиты информации в Интернете: Специальный справочник. — СПб.: Питер, 2002.
- [Медведовский] Медведовский И. Д., Семьянов П. В., Леонов Д. Г. Атака на Интернет. — ДМК, 1999.
- [Нечаев] Элементы криптографии (Основы теории защиты информации): Учеб. пособие для ун-тов и пед. вузов / Под. ред. В. А. Садовниченко. — М.: Высшая школа, 1999.
- [Спесивцев] Спесивцев А. В. и др. Защита информации в персональных компьютерах. М., Радио и связь, 1992.
- [Чмора] Чмора А.Л. Современная прикладная криптография. — М.: Гелиос АРВ, 2001.
- [Ященко] Введение в криптографию / Под общ. ред. В. В. Ященко. — М.: МЦНМО, 1998, 1999, 2000.

Предметный указатель

A

access
◊ role-based 512
◊ rule-based 512
ACL 474
AES
◊ конкурс 261
◊ победитель 292
AH 428
Annualized Loss Expectancy 195
Annualized Rate of Occurrence 195
Application Programming Interface 480

B

Base 64, 350
bitslice operating 282
Business contingency 602

C

CA 179, 357
CBC 343
Certification Authority 179, 357
CFB 343
chaining modes 342
challenge handshake 363
CHAP 382
Common Criteria for Information
Technology Security Evaluation 73
CRL 448

D

DDoS 162
debugger 138
decryption 31
deny-of-service 160
◊ distributed 162
DES 246
digital signature 32
disassembler 138
Disaster recovery 602
DMZ 472
DoS 160

E

EAP 384
ECB 342
EFS 359
Encrypted File System 359
encryption 31
ESP 429

F

Firewall 466
Four eyes 91

G

GOST 259

H

Hash 322
Honey pot 149, 631

I

IDEA 255
IDS 483
IGE 345
Initial Vector 343
Intrusion detection system 483
IPSec 427

- ◊ authentication header 428
- ◊ Encapsulating Security Payload 429
- ◊ IKE 433
- ◊ IKMP 433
- ◊ ISAKMP 433
- ◊ Oakley 434
- ◊ security association 432
- ◊ SKEMI 434

ISAKMP 433

K

KASLT-сеть 242, 277
Kerberos 353, 399
key sharing 359

L

L2F 436
L2TP 436
least privileges 92
LFSR 228
LZ77, 349
LZH 349

M

Man-in-the-middle 179, 356
MARS 264
masquerading 166
MD4, 329
MD5, 327
memory dump 34
Mora Model 194
MS-CHAP 384

N

NAT 473

O

Oakley 434
OFB 343

P

PAP 380
password 31
PKI 443
profile 31
Promiscuous mode 174
Public Key Infrastructure 443

Q

Quantative Risk Model 194

R

RADIUS 394
RAID 607
RC5, 273
RC6, 273
rights 30
Rijndael 276
RLE 349
round 241
RSA 303

S

S/KEY 387
S/MIME 454
SA 432
S-box 238
Script kiddie 117
Secret sharing 359

- ◊ threshold 360

seed 334
separation of duties 91
Serpent 282
SESAME 407
SHA-1 329

Shared secret 359
S-HTTP 410
Side-channel attacks 296
Single Loss Expectancy 195
Single-sign-on 593
SKEMI 434
SKIP 435
Sniffer 122, 137
Sniffer-detector 174
SOCKS 422
SPAP 382
SP-network 242
SSH 417
SSL 412
Substitute-box 238
system garbage 34

T

TACACS 389
TACACS+, 389
TEA 253
Timestamp 347
token 34
trapdoor 33

A

Аварийный план 27, 540, 601
◊ запасное здание 605
◊ классификация событий 602
Авария 30, 602
Авторизация 25
Администратор 31
Алгоритм Лемпеля-Зива 349
◊ создания цепочек 342
Анонимайзер 118
Антивирус 464
Антивирусное обеспечение 27
Апеллируемость 24
Аппаратная закладка 126, 532
Архивирование 614
Атака 30
◊ Buffer overflow 171
◊ CGI 168
◊ Fraggle 165

trojan 33
TwoFish 286

U

Unified Modeling Language 110
username 121

V

Virtual Local Area Network 478
Virtual private network 477
VLAN 478
vulnerability 30

W

weak key 251
whitening 246
WHOIS 148

X

XOR 217
XTACACS 389

◊ MAC-шторм 176
◊ Ping смерти 163
◊ Smurf 164
◊ Sniffing 173
◊ Source routing 167
◊ Spoofing 166
◊ STP 176
◊ SYN-flood 163
◊ URL 169
◊ безусловная 148
◊ динамическая маршрутизация 181
◊ загружаемый код 170
◊ злоумышленник-посредник 179
◊ интерактивная 147
◊ корреляционная на комбинирующий
 поточный шифр 233
◊ локальная 115
◊ маршрутизация от источника 167
(окончание рубрики см. на с. 728)

Атака (окончание)

- ◊ на доступность 146
 - ◊ на конфиденциальность 146
 - ◊ на маршрутизацию 167
 - ◊ на основе:
 - выбранного открытого текста 222
 - известного открытого текста 222
 - шифртекста 222
 - ◊ на поток данных 115
 - ◊ на целостность 147
 - ◊ переполнение буфера 171
 - ◊ перехват сессии 183
 - ◊ повтором 177
 - ◊ подмена 166
 - ◊ поток SYN запросов 163
 - ◊ прослушивание сети 173
 - ◊ салями 523
 - ◊ удаленная 115
- Аттестация объектов 576
- Аудит 25
- Аутентификация 25, 130, 361
- ◊ запрос-ответ 363

Б

- Бастион 475
- Безопасность
- ◊ данных 28
 - ◊ компьютерная 28
 - ◊ физическая 27
- Билет 353
- Биометрические параметры 130
- Брандмауэр 466

В

- Ветвь сети Файштеля 242
- Взломщик шифров 138
- Виртуальная частная сеть 477
- Вирус 32, 153
- ◊ Красный Код 157
 - ◊ Курниковой 156
 - ◊ Нимда 158
 - ◊ Покойник 160
- Восстановление ключей 359

Г

- Гамма 225
- Генератор:
- ◊ вирусов 138
 - ◊ номеров кредитных карт 138
 - ◊ последовательностей наибольшей длины 230
- Годовая частота происшествий 195
- ГОСТ 28147-89, 258
- Группа реагирования 621

Д

- Дамп памяти 34
- Демилитаризованная зона 472
- Депонирование ключей 358
- Дешифрование 31
- Дизассемблер 138
- Дискретный логарифм 309
- Дифференциальный криптоанализ 295
- Длина ключа 236
- Доступ:
- ◊ по правилам 512
 - ◊ по ролям 512
- Доступность 24
- Дружеский взлом 591
- Дублирование 27

Ж

- Жизненная корректировка 590
- Жизненный цикл 589

З

- Забеливание 246
- Закладка 33, 126
- Защивание ЛРС 229
- Заявка на выполнение работ 564
- Зловредная программа 32
- Злоумышленник 31
- Зомби 162

И

- Идентификатор 31
- Идентификация 25, 130

Имя пользователя 31
Инвентаризация 57
 ◇ направления проведения 61
 ◇ принципы проведения 60
Интеллектуальная собственность 507,
 575
Информационная:
 ◇ безопасность 24
 ◇ емкость 330
 ◇ система 30
Информационное пространство 30
Информационный актив 192
Информация 29
Инцидент 626
Исключающее ИЛИ 217

К

Квадрат:
 ◇ Кардано 213
 ◇ Полибия 213
Классификация 68
 ◇ по наличию 71
 ◇ по несанкционированной
 модификации 71
 ◇ по разглашению 72
Ключ 218
 ◇ восстановление 359
 ◇ депонирование 358
 ◇ закрытый 32, 220, 300
 ◇ ключ сеанса 341
 ◇ мастер-ключ 338
 ◇ открытый 32, 220, 300
 ◇ приватный 32
 ◇ публичный 32
 ◇ разделение 359
 ▫ пороговое 360
 ◇ раунда 241
 ◇ сеанса 339
 ◇ секретный 32, 219
 ◇ система управления 351
 ◇ слабый 251
Код Хаффмана 347
Кодирование 218
Компьютерный полигон 155
Контролер 31
Контроль доступа 25
Конфиденциальность 24

Кракер 115
Криптоанализ 31, 212
Криптоатака 293
 ◇ man-in-the-middle 356
 ◇ дифференциальный
 криптоанализ 295
 ◇ злоумышленник-посредник 356
 ◇ метод встречи посередине 293
 ◇ линейный криптоанализ 295
 ◇ на отказах 298
 ◇ по весу Хэмминга 297
 ◇ по времени исполнения 296
 ◇ по потребляемой мощности 297
 ◇ по частоте попадания в кэш 297
Криптография 26
 ◇ асимметричная 299
 ◇ с ключом 218
Криптосистема 337

Л

Линейные регистры сдвига 228
Линейный криптоанализ 295
Лист доступа 474
Лицензирование деятельности 579
Логин 31
Логическая бомба 33, 523
ЛРС 228
Люк 33

М

Маскировка 166
Материал ключа 240
Межсетевой экран 27, 466
 ◇ пакетный фильтр 467
 ◇ посредник приложения 469
 ◇ распределенный
 или персональный 470
 ◇ с контролем виртуального
 соединения 468
Минимизация привилегий 92
Модели доступа 509
Модель:
 ◇ безопасности 26, 32
 ◇ информационных потоков 105

Модель рисков:

- ◊ качественной оценки 194
 - ◊ количественная 194
 - ◊ Миоры 194
 - ◊ основная линия 194
- Мониторинг 25

Н

- Наиболее опасные сервисы 152
- Неотрекаемость 24
- Несанкционированное действие 30
- Нормативное обеспечение 26

О

Обмен ключами:

- ◊ схема Диффи-Хеллмана 355
 - ◊ схема Шамира 354
- Образующая функция сети Файштеля 242

Обратимая функция 224

Объект 30, 67

Обязательное управление доступом 512

Одноалфавитная:

- ◊ замена 212
 - ◊ подстановка 212
- Одноразовый блокнот 217

Ожидаемый:

- ◊ годовой ущерб 195
- ◊ единичный ущерб 195

Основная теорема безопасности 512

Отказ в обслуживании 160

- ◊ распределенный 162

Отладчик 138

Ошибка 32

П

Парадокс дней рождения 323

Пароль 31, 216, 219, 338

- ◊ по умолчанию 121

Переназначаемое управление доступом 512

ПИН 33

Побочное электромагнитное излучение и наводки 533

Политика безопасности 25, 539

- ◊ концепция 539

- ◊ методы 540

- ◊ процедуры 539

- ◊ стандарты 539

Пользователь 31

Последовательность наибольшей длины 230

Потайная дверь 301

Права 30

Правила информационной безопасности 30

Принцип Кергоффа 218

Приоритеты восстановления систем 603

Программа

- ◊ закладка 33

- ◊ троянская 33

- ◊ повышения прав 137

- ◊ подбора паролей 138

- ◊ поиска уязвимостей 150

Программная закладка 522

Проектирование механизмов безопасности в ПО 508

Прокси-службы 118

Протоколы аутентификации:

- ◊ CHAP 382

- ◊ EAP 384

- ◊ Kerberos 399

- ◊ MS-CHAP 384

- ◊ RADIUS 394

- ◊ S/Key 387

- ◊ SESAME 407

- ◊ SPAP 382

- ◊ TACACS 389

- ◊ TACACS+, 389

- ◊ XTACACS 389

- ◊ PAP 380

Профиль 31

Псевдослучайные числа 333

Р

Разделение ключа:

- ◊ пороговое 360

Разделение ключей:

- ◊ схема Блэкли 361

- ◊ схема Шамира 360

Разделение секрета 359
Разложение на множители 305
Размер ключа 236
Разрядность:
 ♦ блока 236
 ♦ ЛРС 229
Распределение ответственности 91
Расследование 626
 ♦ Honey pot 631
 ♦ сбор улики 631
Расшифрование 31
Раунд 241
Реагирование 25
Резервирование 27
Резервное копирование 27, 614
 ♦ метод различий или
 дифференцированный 615
 ♦ наращиваемое или
 инкрементальное 615
 ♦ Полное 615
РЕМ 450
RPTP 436
Руководитель службы ИБ 47

С

С2, 69
сертификат 357
 ♦ владелец ключа 357
 ♦ заверяющее лицо 357
 ♦ удостоверяющий центр 357
 ♦ цепочка доверия 357
Сертификация средств 578
Сеть Файштеля 242
Сжатие данных 347
Система обнаружения атак 27
Системный мусор 34
Системы обнаружения и
 предотвращения атак 483
Сканер:
 ♦ безопасности 27
 ♦ уязвимостей 481
Сканирование портов 120
Скремблер 226
Сложение по модулю 2, 216, 217
Слой:
 ♦ сети 241
Случайные числа 330

Сниффер 122, 137
Состояние ЛРС 228
Социальная инженерия 124
Средство работы с информацией 30,
 67
Субъект 30, 67
 ♦ информационного пространства 90

Т

Таблица Виженера 215
Табличная подстановка 238
Тайна:
 ♦ государственная 571
 ♦ коммерческая 571
Тайнопись 218
Такт ЛРС 228
Термин linear feedback shift registers
 228
Тестирование:
 ♦ анализ результатов 597
 ♦ жизненная корректировка 590
 ♦ полномасштабное 590
 ♦ практическое 590
 ♦ теоретическое 589
Токен 34, 131
Топология атакуемой сети 120
Трансляция сетевых адресов 473
Транспортное кодирование 349
Троянская программа 33, 153
Троянский конь 33

У

Угроза 30
 ♦ firmware 128
 ♦ GetAdmin 142
 ♦ l0phtcrack 142
 ♦ NTFSDOS 142
 ♦ аппаратное обеспечения 126
 ♦ внешний носитель 129
 ♦ имитация системного
 приглашения 134
 ♦ повышение привилегий 136
 ♦ постороннее программное
 обеспечение 137
 ♦ слабый пароль 134
Узлы замен 259

Уплотнение операционной системы 475
Управление рисками 25, 191
Установленные производителем административные счета 121
Учет 24
Уязвимость 30, 481
◊ BIND 200
◊ CGI 201
◊ IIS 201
◊ IMAP и POP 202
◊ ISAPI 204
◊ Remote procedure calls 201
◊ R-команд 205
◊ sadmind и mountd 201
◊ Sendmail 201
◊ sharing 201, 204
◊ SNMP 202
◊ Unicode в IIS 203
◊ аутентификационных данных 201
◊ нулевая сессия 204
◊ сервиса LPD 205
◊ хэширования в базе данных SAM 204

Ф

Файрвол 466
Факторизация 305
Фрикер 115

Х

X.509, 446
Хакер 115
Хэш-коллизия 323
Хэш-функция 322
◊ MD4, 329
◊ MD5, 327
◊ SHA-1, 329
◊ криптостойкая 323
◊ на блочном шифре 326

Ц

Целостность 24
Центр сертификации 357
Цикл:
◊ сети 241
Цифровой сертификат 443

Ч

Червь Моррисона 150

Ш

Шифр:

◊ AES 292
◊ DES 246
◊ IDEA 255
◊ KASLT-сеть 242
◊ MARS 264
◊ RC5 273
◊ RC6 273
◊ Rijndael 276
◊ RSA 303
◊ Serpent 282
◊ SkipJack 358
◊ SP-сеть 242
◊ TEA 253
◊ TwoFish 286
◊ абсолютно стойкий 217
◊ асимметричный 219
◊ блочный 220
◊ блочный абсолютно симметричный 236
◊ Вернама 216
◊ гаммирования 225
◊ ГОСТ 28147-89, 258
◊ квадрат Кардано 213
◊ квадрат Полибия 213
◊ комбинационный 241
◊ Мак-Элиса 312
◊ многоалфавитной замены 215
◊ на открытом ключе 220
◊ на секретном ключе 219
◊ на эллиптических кривых 311
◊ надежный 223
◊ одноразовый блокнот 217
◊ перестановочный 214
◊ по книге 216
◊ поточный 220
 ▫ динамический 234
 ▫ комбинирующий 232
 ▫ фильтрующий 232
◊ практически стойкий 222
◊ прямой 241
◊ Рабина 307
◊ сеть Файштеля 242
◊ симметричный 219

- ◊ таблица Виженера 215
- ◊ Цезаря 211
- ◊ Эль Гамаль 309
- Шифрование 31
- ◊ асимметричное 299
- ◊ на открытом ключе 299
- Шифрующая последовательность 224
- Штамп времени 347

Э

Экспорт данных 106

Электронная цифровая
подпись 32, 301

- ◊ корректная 32

Электронный документ 31

Энтропия 330

ЭЦП 32, 301

- ◊ RSA 314
- ◊ ГОСТ Р 34.10-2001, 319
- ◊ Рабина 316
- ◊ симметричная 324
- ◊ Эль Гамаль 317