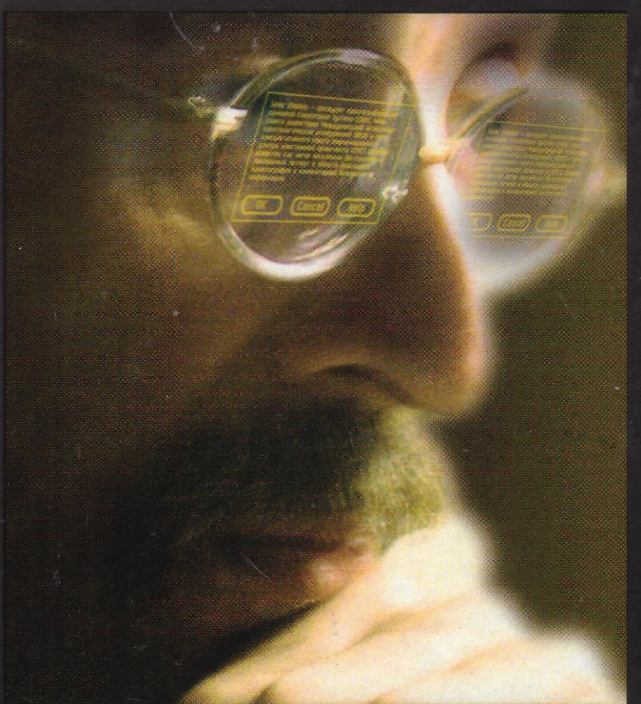
КЛАССИКА COMPUTER SCIENCE

# СЕКРЕТЫ И ЛОЖЬ

## БЕЗОПАСНОСТЬ ДАННЫХ В ЦИФРОВОМ МИРЕ



Б. ШНАЙЕР

WILEY

ПИТЕР®

СЕРИЯ

КЛАССИКА COMPUTER SCIENCE

ПИТЕР®

# SECRETS AND LIES

## DIGITAL SECURITY IN A NETWORKED WORLD

**Bruce Schneier**

**Wiley Computer Publishing**
**John Wiley & Sons, Inc.**

New York • Chichester • Weinheim • Brisbane • Singapore • Toronto

КЛАССИКА COMPUTER SCIENCE

Б. ШНАЙЕР

# СЕКРЕТЫ И ЛОЖЬ

## БЕЗОПАСНОСТЬ ДАННЫХ В ЦИФРОВОМ МИРЕ

ПИТЕР®

Москва • Санкт-Петербург • Нижний Новгород • Воронеж
Ростов-на-Дону • Екатеринбург • Самара
Киев • Харьков • Минск
2003

ББК 32.973.23-07
УДК 681.322
Ш76

**Ш76 Секреты и ложь. Безопасность данных в цифровом мире** / Б. Шнайер. — СПб.: Питер, 2003. — 368 с.: ил. — (Серия «Классика computer science»).

**ISBN 5-318-00193-9**

В этой книге Брюс Шнайер — автор нескольких бестселлеров и признанный специалист в области безопасности и защиты информации, опираясь на собственный опыт, разрушает заблуждения многих, уверенных в конфиденциальности и неприкосновенности информации. Он разъясняет читателям, почему так сложно предотвратить доступ третьих лиц к личной цифровой информации, что нужно знать, чтобы обеспечить ее защиту, сколько средств следует выделять на обеспечение корпоративной безопасности и многое, многое другое.

ББК 32.973.23-07
УДК 681.322


**ISBN 0-471-25311-1 (англ.)**

**ISBN 5-318-00193-9**

# Краткое содержание

# Содержание

*Посвящается Карен*

# Предисловие

**Я** написал эту книгу во многом для того, чтобы исправить собственную ошибку.

Семь лет назад мною была написана книга «Прикладная криптография» («Ap¬ plied Cryptography»). В ней я создал математическую утопию — алгоритмы, тыся¬ челетиями хранящие ваши глубочайшие секреты, протоколы передачи данных, обеспечивающие воистину фантастические возможности: неконтролируемые из¬ вне финансовые операции, необнаружимую аутентификацию, анонимную оплату. И все это — незаметно и надежно. В моем видении криптография была великим технологическим уравнителем: с ее помощью каждому дешевому (и дешевеюще¬ му с каждым годом) компьютеру могла быть обеспечена такая же безопасность, как и компьютерам всемогущего правительства. Во втором издании той книги я зашел так далеко, что написал: «Недостаточно защищать себя с помощью закона; мы нуждаемся и в том, чтобы защитить себя с помощью математики».

Все это — неправда. Криптография не может ничего подобного. И не потому, что она стала хуже с 1994 года или написанное мною тогда перестало быть правдой сегодня, но оттого, что криптография существует не в вакууме.

Криптография — это раздел математики и, как и прочие ее разделы, связана с числами, уравнениями и логикой. Безопасность — реальная, ощутимая безопас¬ ность, столь необходимая нам с вами, — связана с людьми: с уровнем их знаний, их взаимоотношениями и с тем, как они управляются с машинами. Информационная безопасность связана с компьютерами — сложными, нестабильными, несовершен¬ ными компьютерами.

Математика абсолютна; окружающий мир субъективен. Математика совершен¬ на; компьютеры могут ошибаться. Математика логична; люди, как и компьютеры, ошибаются, они своевольны и едва ли предсказуемы.

Ошибка «Прикладной криптографии» была в том, что я рассуждал обо всем независимо от контекста. Я говорил о криптографии так, как будто она и есть Ответ™. Я был потрясающе наивен.

Результат же был вовсе плох. Читатели поверили, что криптография — род не¬ коей магической пыли, которая покроет их программное обеспечение и сделает его неуязвимым. И они произносили магические заклинания вроде «128-битовый ключ» или «инфраструктура открытого ключа». Как-то однажды коллеги поведа¬ ли мне, что мир наполнился плохими системами безопасности, сконструирован¬ ными людьми, прочитавшими «Прикладную криптографию».

С момента написания той книги я занимался тем, что давал консультации по криптографии: по всем вопросам, связанным с разработкой и анализом систем безопасности. К своему несказанному удивлению, я обнаружил, что слабые места в системах безопасности отнюдь не определяются недостатками математических моделей. Они были связаны с аппаратурой, программами, сетями и людьми. Пре¬

**красные математические ходы** становились никчемными из-за небрежного программирования, гнусной операционной системы или просто выбора кем-то плохого пароля.

В поисках слабины я научился смотреть шире, рассматривая криптографию как часть системы. Я начал повторять пару сентенций, которые красной нитью проходят через всю эту книгу: «Безопасность — это цепь: где тонко, там и рвется» и «Безопасность — это процесс, а не продукт».

Любая реальная система — запутанная серия взаимодействий. Защита должна распространяться на все компоненты и соединения этой системы. И в этой книге я старался показать, что в современных системах настолько много компонентов и связей — некоторые из них неизвестны даже создателям, а тем более пользователям, — что угроза для безопасности всегда остается. Ни одна система не совершенна; ни одна технология не есть Ответ™.

Сказанное очевидно каждому, кто знаком с проблемами безопасности на практике. В реальном мире за словом «безопасность» скрывается ряд процессов. Это не только упреждающие мероприятия, но и обнаружение вторжения, его пресечение и целая судебная система, позволяющая выследить виновного и преследовать его по суду. Безопасность — не продукт, она сама является процессом. И если мы должны обеспечить безопасность нашей вычислительной системы, нам необходимо начать разработку этого процесса.

Несколько лет назад я слышал цитату, которую слегка изменил: «Если вы думаете, что технология может решить проблемы безопасности, то вы не понимаете ни проблем безопасности, ни технологии».

Эта книга о проблемах безопасности, о технологических ограничениях и о поиске решения.

## Как читать эту книгу

Читайте эту книгу по порядку, от начала до конца.

И это действительно необходимо. Во многих технических книгах авторы скользят по поверхности, лишь эпизодически залезая поглубже; чаще всего они следуют структуре справочника. Эта книга не такова. В ней прослеживается четкая линия: это повествование, рассказ. И подобно любому хорошему рассказу, мало толку читать ее беспорядочно. Главы основываются одна на другой, и вы сможете вкусить все радости окончательной победы, только пройдя весь путь до конца.

Более того, я хотел бы, чтобы вы прочли книгу один, а потом еще и второй раз.

Эта книга доказывает, что для понимания безопасности системы необходимо рассматривать ее целиком, а не раскладывать на отдельные технологии. Безопасность сама по себе — взаимосвязанная система, и это означает, что сначала нужно приобрести некоторые знания по всем имеющим к ней отношение вопросам, а затем уже углубляться в тот или иной предмет.

Но два прочтения... Возможно, я хочу слишком многого. Забудьте об этом.

Книга состоит из трех частей:

- часть I «Ландшафт» дает общий вид картины: кто такие взломщики, чего они хотят и что нужно делать, чтобы предотвратить угрозу;

- часть II «Технологии» в основном описывает различные технологии безопасности и их ограничения;
- часть III «Стратегии» в соответствии с окружающим ландшафтом и ограничениями технологий определяет, что же мы теперь должны делать.

Я думаю, безопасность информационных систем — самая потрясающая вещь, которой можно заниматься в наши дни, и книга отражает это мое ощущение. Это серьезно, но и весело — несомненно. Читайте и получайте удовольствие.

# Благодарности

Очень многие люди читали эту книгу на разных стадиях ее подготовки. Я хотел бы поблагодарить тех, кто читал наиболее ранний вариант этой книги: Стива Басса, Сьюзен Гринспан, Криса Холла, Джона Келси и Мадж. Их советы помогли мне окончательно определить как содержание, так и стиль изложения. Мне хотелось бы также выразить благодарность Бет Фридман за ее помощь в основательном редактировании книги, когда она была написана еще лишь наполовину, и редактировании других ее частей, а также за помощь в руководстве работой редактора и корректора, Карен Купер за помощь в корректуре и Рафаилу Картеру за помощь в редактировании, когда работа над книгой уже близилась к концу. Ценные замечания при чтении книги или ее частей сделали: Микеланджело, Кен Айер, Стив Басс, Дэвид Дайер-Беннетт, Эд Беннетт, Рассел Бранд, Карен Купер, Дэвид Коуен, Уолта Куртис, Дороти Деннинг, Карл Эллисон, Эндрю Фернандес, Гордон Форс, Эми Форсайт, Дин Гэлон, Дрю Гросс, Грегори Гуерин, Питер Гутманн, Марк Харди, Дейв Инат, Крис Джонстон, Джеймс Джораш, Ариен Ленстра, Стюарт Мак Клур, Гэри Мак Гроу, Дуг Меррилл, Джефф Мосс, Симона Несс, Артимадж Нельсон, Питер Ньюман, Эндрю Одлизко, Дуг Прайс, Джеймс Риордан, Бернард Руссели, Том Роули, Эви Рубин, Риан Рассел, Адам Шостак, Симон Сингх, Джим Уолнер и Элизабет Цвикки. Благодаря этим людям книга стала более завершенной, точной и интересной. Все недостатки, пропущенные ошибки и чрезмерное многословие остаются на совести автора.

# От издательства

Ваши замечания, предложения и вопросы отправляйте по адресу электронной почты comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

Все исходные тексты, приведенные в книге, вы можете найти по адресу http://www.piter.com/download.

Подробную информацию о наших книгах вы найдете на веб-сайте издательства http://www.piter.com.

# Глава 1. Введение

В марте 2000 года я занимался тем, что собирал воедино сведения из различных источников о событиях, связанных с проблемой компьютерной безопасности. Вот эти сведения.

- Кто-то взломал веб-сайт SalesGate.com (электронная коммерция B2B, business-to-business) и украл около 3000 записей, содержащих номера кредитных карт клиентов и информацию частного характера. Часть этой информации он поместил в Интернете.
- В течение нескольких лет частная информация утекала с веб-сайтов (таких как Intuit) к рекламодателям (например, DoubleClick). Когда посетители производили расчеты на сайте Intuit, вводимая ими информация посылалась и на DoubleClick благодаря ошибке, допущенной программистами при создании сайта. Все это происходило без ведома пользователей и, что более удивительно, без ведома Intuit.
- Осужденный за свои преступления хакер Кэвин Митник в показаниях Конгрессу сказал, что наиболее уязвимое место в системе безопасности — «человеческий фактор». Он зачастую выведывал пароли и другую секретную информацию, действуя под чужим именем.
- Опрос службы Гэллапа показал, что каждый третий из тех, кто делает покупки через Интернет, будет делать их, пожалуй, менее охотно в свете последних событий, связанных с проблемами безопасности.
- Частные данные клиентов, заказывавших PlayStation 2 на веб-сайте корпорации Sony, «утекли» к неким другим клиентам. (Это — актуальная проблема всех типов сайтов. Многие посетители «отмечаются» на них в надежде получить информацию о покупателях сайта.)
- Директор ЦРУ отрицал, что Соединенные Штаты участвовали в экономическом шпионаже, но не стал отрицать существования обширной разведывательной сети, называемой ECHELON.
- Некто Пьер-Гай Лавойе, 22 лет, был осужден в Квебеке за взлом системы безопасности нескольких правительственных компьютеров США и Канады. Он провел 12 месяцев в заключении.
- Министерство обороны Японии приостановило внедрение новой компьютерной системы безопасности после того, как установило, что программное обеспечение было разработано членами секты Аум Синрике.

- Новый почтовый вирус, названный «Чудесный парк» (Pretty Park), распространился через Интернет. Это новая разновидность вируса, появившегося годом раньше. Он рассылался автоматически по всем адресам, имеющимся в почте пользователей программы Outlook Express.
- Novell и Microsoft продолжают препираться по поводу замеченных ошибок в системе безопасности Active Directory операционной системы Windows 2000: кто из них должен обеспечивать параметры безопасности, установленные вами для своего каталога (я лично верю, что это конструктивный недостаток Windows, а не ошибка).
- Двое сицилианцев (Джузеппе Руссо и его жена Сандра Элазар) были арестованы за кражу через Интернет около тысячи кредитных карт США. Они использовали эти карты для закупки лучших товаров и лотерейных билетов.
- Отражена серия атак, направленных на отказ в обслуживании, проведенных хакером (на самом деле — скучающим подростком) по имени Кулио (Coolio). Он признал, что в прошлом вскрыл около 100 сайтов, в том числе сайт криптографической компании RSA Security и сайт, принадлежащий государственному департаменту США.
- Злоумышленники организовали атаки, приведшие к отказам в обслуживании на веб-сайте компании Microsoft в Израиле.
- Джонатан Босанак, известный как Гетсби (Gatsby), был приговорен к 18 месяцам заключения за взлом сайтов трех телефонных компаний.
- Военные Тайваня объявили, что обнаружили более 7000 попыток со стороны китайских хакеров войти в систему безопасности страны. Эта жуткая статистика не была конкретизирована.

Вот еще несколько сообщений о нарушениях систем безопасности в марте 2000 года:

- Лазейка, обнаруженная в Microsoft Internet Explorer 5 (в Windows 95, Windows 98, Windows NT 4.0 и Windows 2000), позволила злоумышленнику создать веб-страницу, дающую ему возможность запустить любую программу на компьютере посетителя сайта.
- Модифицировав URL, некий нарушитель сумел полностью обойти механизм аутентификации, защищающий удаленных пользователей серверов Axis StarPoint CD-ROM.
- Обнаружено, что если атакующий пошлет Netscape Enterprise Server 3.6 некое длинное сообщение, то переполнение буфера приведет к прекращению работы программы, а атакующий сможет выполнить в этом случае на сервере любой код.
- Стало известно, что некоторые атаки (приводящие к отказу в обслуживании или направленные на подавление сценария CGI) могут быть проведены таким образом, что программа RealSecure Network Intrusion Detection их не обнаружит.
- Выяснилось, что, посылая определенный URL на сервер, обслуживаемый программой ColdFusion компании Allaire, злоумышленник может получить

сообщение об ошибке, содержащее информацию о физических адресах различных файлов.

- Omniback — это система резервного копирования компании Hewlett-Packard. Злоумышленник может использовать ее для проведения атаки, приводящей к отказу в обслуживании.
- Эмулятор DOS Dosemu, поставляемый с Corel Linux 1.0, имеет слабое место, позволяющее пользователю выполнять привилегированные команды.
- Манипулируя значениями некоторых параметров DNSTools 1.8.0, злоумышленник может воспользоваться недостатками программы и выполнить произвольный код.
- Пакет InfoSearch для работы с CGI автоматически конвертирует текстовые документы в HTML. Ошибка в CGI-сценарии позволяет нарушителю выполнять на сервере команды на уровне привилегий веб-сервера.
- Найдены слабые места в почтовой программе The Bat!, позволяющие злоумышленникам красть файлы с компьютеров пользователей.
- Clip Art Gallery компании Microsoft позволяет пользователю загружать файлы с клипами с веб-страниц. При определенных условиях видоизмененный файл клипа может вызвать выполнение произвольного кода на компьютере пользователя.
- Если вы посылаете определенное имя пользователя и пароль (даже если он неправильный) на FTP-сервер 3.5 Bison Ware, он выйдет из строя.
- Используя специальным образом модифицированные URL, злоумышленник может повредить Windows 95 и Windows 98 на компьютерах пользователей.

Ниже представлен список 65 веб-сайтов, которые, согласно информации, опубликованной на сайте attrition.org, были обезображены в течение месяца. В данном контексте слово «обезображены» означает, что некто вскрыл сайт и заменил главную страницу.

Tee Plus; Suede Records; Masan City Hall; The Gallup Organization; Wired Connection; Vanier College; Name Our Child; Mashal Books; Laboratorio de Matematica Aplicada da Universidade Federal do Rio de Janeiro; Elite Calendar; Parliament of India; United Network for Organ Sharing; UK Jobs; Tennessee State University; St. Louis Metropolitan Sewer District; College of the Siskioyous; Russian Scientific Center for Legal Information; Ministry of Justice; RomTec Pic; Race Lesotho; Monmouth College; Association of EDIUsers; Bitstop, Inc; Custom Systems; Classic Amiga; 98 Sskate; CU Naked; Korea National University of Education; PlayStation 2; Assotiation for Windows NT User Group Bloem S.A.; Aware, Inc.; Ahmedabad Telephone Online Directory; Ahmedabad Telecom District; Fly Pakistan: Quality Business Solutions; Out; Internet Exposure; Belgium Province de Hainvan Wervings en Sectiebureaus; Engineering Export Promotion Council, Ministry of Commerce, India; AntiOnline's Anticode; Pigman; Lasani; What Online; Weston High School; Vasco Boutique; True Systems; Siemens Italy; Progress Korea; Phase Device Ltd.; National Postal Mail Handlers Union; Metrics; Massachusetts Higher Education Network; The London Institute; Fort Campbell School System; and MaxiDATA Tecnologia e Informatica Ltda.

И наконец, атаки на домашний компьютер, принадлежащий одному моему другу и подсоединяющийся к Интернету через модем:

- двадцать шесть просмотров с целью обнаружить слабые точки;
- четыре обнаруженные попытки взлома компьютера во время поиска слабых точек;
- множество других хакерских трюков.

Фиксируя подобные события только в течение первой недели марта 2000 года, я воистину устал от этого занятия.

Обозревая получившийся список, можно сказать, что устрашающе действует прежде всего большое разнообразие проблем, слабых точек и атак. Слабые места имеются в программах, которые мы считаем защищенными; есть они даже в самих системах безопасности. Некоторые из них присутствуют в системах электронной коммерции, при конструировании которых наверняка учитывались вопросы безопасности. Одни из них есть в новых программах, другие — в программах, которые продаются уже в течение многих лет. Нередко поставщики программного обеспечения никак не хотят согласиться, что у их продукции имеются проблемы с безопасностью.

Первые семь дней марта 2000 года не были исключением. И в другие недели случались подобные происшествия, а некоторые из них были еще показательнее. Действительно, факты говорят о том, что положение вещей ухудшается: число «дыр» в системах безопасности, взломов и отказов постоянно возрастает. Даже если мы будем больше знать о безопасности (как сконструировать криптографические алгоритмы, как построить безопасную операционную систему), все равно обеспечить полную безопасность будет невозможно. Почему это так и что можно с этим сделать — и является предметом нашего разговора.

# Системы

Понятие «система» относительно ново для науки. Восточные философы уже давно рассматривали мир как единую систему, состоящую из различных компонентов, но в западной традиции было принято разделять его на отдельные явления, развивающиеся в различных направлениях.

Машины только недавно стали системами. Подъемный блок — это машина; но лифт — это комплексная система, включающая много различных механизмов. Системы взаимодействуют: лифт взаимодействует с электрической системой здания, с его системой пожарной безопасности и, возможно, даже с его системой защиты окружающей среды. Компьютеры, взаимодействуя, образуют сети; сети, взаимодействуя, образуют более крупные сети... в общем, вы уловили идею.

Адмирал Грейс Хоппер говорил: «Жизнь была проще перед Второй мировой войной. После этого у нас появились системы». Это — очень проницательный взгляд на вещи.

Если вы используете концепцию систем, можно проектировать и строить в более крупном масштабе. Есть разница между созданием особняка и небоскреба, оружия и ракеты Patriot, посадочной полосы и аэропорта. Каждый может изготовить

светофор, но разработать городскую систему транспортного контроля значительно сложнее.

Интернет, возможно, наиболее сложная система, которая когда-либо разрабатывалась. Она включает в себя миллионы компьютеров, объединенных в непредсказуемо сложные физические сети. На каждом компьютере работают сотни программ, некоторые из них взаимодействуют с программами, установленными на том же компьютере, другие — через сеть с программами, установленными на удаленных компьютерах. Система принимает сигналы от миллионов пользователей, часто одновременно.

Один человек сказал: «Сэр, это зрелище подобно собаке, стоящей на задних лапах. Удивительно не то, что она делает это недостаточно хорошо, а то, что она вообще способна это делать».

Системы имеют несколько интересных свойств, которые уместно здесь обсудить.

Во-первых, они сложны. Механизмы просты: молоток, дверные петли, нож для бифштексов. Системы же намного сложнее: они имеют компоненты, петли обратной связи, среднее время отклика, инфраструктуру. Цифровые системы весьма затейливы: даже простая компьютерная программа состоит из тысяч строчек кода, описывающего всевозможные разновидности различных операций. Сложная программа имеет тысячи компонентов, которые могут работать как по отдельности, так и во взаимодействии друг с другом. Именно для сложных цифровых систем было разработано объектно-ориентированное программирование.

Во-вторых, системы взаимодействуют, формируя еще более крупные системы. Это может быть сделано намеренно — программисты используют объекты, чтобы дробить большие системы на малые, инженеры подразделяют большие механические системы на малые подсистемы, и так далее, — или происходит естественным образом. Изобретение автомобиля привело к развитию современной системы дорог и магистралей, а она, в свою очередь, взаимодействуя с другими системами, существующими в нашем мире, породила то, что мы называем мегаполисом. Система контроля авиалиний взаимодействует с навигационной системой самолета и с системой предсказания погоды. Тело человека взаимодействует с другими человеческими организмами и с другими системами на планете. Интернет переплетается со всеми наиболее важными системами в нашем обществе.

В-третьих, системы обладают неожиданными свойствами. Другими словами, они иногда могут делать вещи, которых разработчики и пользователи не ожидают от них. Телефонная система, например, породила новый способ взаимодействия людей. (Александр Грэхем Белл не имел представления, что телефон станет прибором для осуществления персональной связи, — он предполагал использовать его для сообщений о приходе телеграммы.) Автомобили изменили пути, на которых люди встречаются, назначают свидания и влюбляются. Системы защиты окружающей среды в зданиях оказывают влияние на здоровье людей, что также сказывается на работе системы здравоохранения. Системы обработки текстов изменили способ их написания. У Интернета полно неожиданных свойств: вспомните об электронных покупках, виртуальном сексе, совместном авторстве.

И — четвертое: в системах имеются «баги»[1]. Они являются особой разновидностью ошибки. Их наличие — неожиданное свойство системы, не предусмотренное при ее создании. «Баги» принципиально отличаются от сбоев. Если где-то происходит сбой, продолжение работы невозможно. Когда же имеется «баг», работа продолжается, хотя объект, ее производящий, ведет себя «плохо»: возможно, неустойчиво, возможно, необъяснимо. «Баги» — уникальное свойство систем. Машины могут ломаться или портиться, или не работать вовсе, но только системы могут иметь «баги».

# Системы и безопасность

Все перечисленные свойства оказывают влияние на безопасность систем. Ухищрения — вот точное определение для безопасности на сегодняшний день, поскольку обезопасить сложную систему, подобную Интернету, трудно именно в силу ее сложности. Безопасность систем — дело сложное, а безопасность сложных систем в особенности.

Обычный для компьютеризованных систем механизм тиражирования игнорирует наличие системы как таковой и сосредоточен на отдельных машинах — такова технология... Поэтому мы загружены работой по выработке технологий безопасности: криптография, брандмауэры, инфраструктура ключей общего доступа, сопротивление несанкционированному доступу. Эти технологии просты для понимания и обсуждения и достаточно просты в использовании. Но было бы наивно полагать, что они способны неким таинственным образом наполнить системы свойством:

```
<reverence type - 'hushed'> Security </reverence>
(<тип уважения = 'секретность'> Безопасность </уважение>)
```

Увы, так не случается, и подтверждение тому можно видеть в моем отчете за 7 дней марта 2000 года. Причина большинства событий, связанных с нарушением безопасности, коренится в четырех свойствах систем, рассмотренных ранее:

- **Сложность.** Проблемы безопасности в Active Directory операционной системы Windows 2000 прямо вытекают из сложности любой компьютерной системы каталогов. Я думаю, что они зиждятся на недостатке, заложенном при проектировании: Microsoft применила конструкторское решение, обеспечивающее удобство пользователям, но небезупречное с точки зрения безопасности.
- **Взаимодействие.** Взаимодействие между программным обеспечением вебсайта Intuit и программным обеспечением DoubleClick, производящее отображение объявлений пользователей, привело к утечке информации от одного к другому.
- **Неожиданность.** Судя по сообщениям в прессе, программисты Sony не знают, как происходит утечка информации о кредитных картах от одного пользователя к другому. Она просто происходит.

[1] От английского bug (жучок). Однозначного русского перевода не существует. — *Примеч. ред.*

- **«Баги».** Уязвимость Netscape Enterprise Server 3/6 была следствием программного «бага». Нарушитель мог использовать этот «баг», породив проблему для безопасности.

Многие страницы этой книги (особенно в третьей ее части) посвящены детальному объяснению, почему безопасность мыслится как система внутри большой системы, но пока я хочу, чтобы для начала вы просто держали в голове две вещи.

Первое — это соотношение между теорией и практикой безопасности. Существует целая куча теорий безопасности: теория криптографии, теория брандмауэров и обнаружения вторжения, теория биометрик. В истории полно примеров, когда система была основана на великой теории, но терпела поражение на практике. Йоги Берра однажды сказал: «В теории нет различия между теорией и практикой. На практике есть».

Теоретические изыскания лучше всего подходят для идеальных условий и лабораторных установок. Самая популярная шутка на занятиях физикой в моем колледже была: «Рассмотрим сферическую корову с равномерно распределенной плотностью». Некоторые вычисления мы можем производить только для идеализированной системы: реальный мир гораздо сложнее, чем теория. Цифровые системы безопасности также подчиняются этому закону: мы можем сконструировать идеализированные операционные системы так, что они, вероятно, будут безопасными, но мы не можем заставить их действительно безопасно работать в реальном мире. В реальном мире существуют несоответствия проекту, неприметные изменения и неправильные реализации.

Реальные системы не подчиняются теоретическим решениям. Совпадения случаются только тогда, когда сферическая корова обладает такими же неожиданными свойствами, как и реальная Буренка. Именно по этой причине ученые — не инженеры.

Вторая важная вещь, которую нужно помнить — это соотношение между предупреждением, обнаружением и реагированием. Хорошая защита объединяет все три звена: безопасное хранилище, чтобы сохранить ценности, сигнализацию, чтобы обнаружить грабителей, если они захотят туда проникнуть, и полицию, которая отреагирует на сигнал тревоги и поймает грабителей. В системах компьютерной безопасности наблюдается тенденция полагаться в основном на упреждающие меры: криптография, брандмауэры и т. д. В большинстве случаев в них не заложено обнаружения и почти никогда нет реагирования и преследования. Такая стратегия оправдана только тогда, когда предупредительные меры совершенны: в противном случае кто-нибудь наверняка сможет сообразить, как их обойти. Большинство уязвимых мест и, соответственно, нападений, описанных в данном разделе, — это результат несовершенства превентивных механизмов. В реальности же нашего мира обнаружение и реагирование очень существенны.

# Часть I

# Ландшафт

Компьютерную безопасность часто представляют абстрактно: «Эта система защи¬щена». Продавец программного обеспечения может сказать: «Эта программа га¬рантирует защиту вашей сети» или «Мы обеспечиваем безопасность электронной коммерции». Подобные заявления неизбежно несут на себе отпечаток наивности и упрощенчества. Это означает, что обращается больше внимания на безопасность программы, чем на безопасность системы. Первый вопрос, который следует задать в таком случае: «От кого и от чего защищена система?»

Это актуальный вопрос. Представьте себе продающуюся безопасную опера¬ционную систему. Обеспечит ли она защиту от ручной гранаты, если та попадет прямиком в ваш процессор? Или от того, кто нацелит видеокамеру непосред¬ственно на вашу клавиатуру или экран монитора? От того, кто «просочился» в ва¬шу компанию? Скорее всего нет: не потому, что эта операционная система плоха, но потому, что некто более или менее осознанно воплощает конструкторские ре¬шения, в которых определено, какие виды возможных атак эта операционная система будет предотвращать (и, возможно, предотвратит), а какие она будет иг¬норировать.

Проблемы усугубляются, когда подобные решения принимаются недостаточ¬но обдуманно. И не все всегда так очевидно, как в предыдущем примере. Защища¬ет ли система безопасности телефонной линии от того, кто может вас случайно подслушать, от перехватчика, имеющего большие финансовые возможности или от национального разведывательного управления? Защищает ли система безопас¬ности банка от мошенничества клиентов, от мошенничества продавцов, от мо¬шенничества банковского кассира или от мошенничества управляющего банком? Приведет использование другого изделия к усилению или ослаблению безопасно¬сти? Точное понимание того, что может обеспечить отдельно взятая технология безопасности и чего она не может, является слишком сложным для большинства людей.

Безопасность нельзя представить только в белом или только в черном цве¬те; контекст часто играет большую роль, чем сама технология. Тот факт, что надежная операционная система не может защитить от ручной гранаты, не означает, что она бесполезна; он означает только, что мы не можем просочить¬ся сквозь стены, дверные замки и оконные решетки. Каждая конкретная техно¬логия занимает свое важное место в общей концепции безопасности системы. Система может быть защищена от обыкновенных преступников, или от про¬мышленного шпионажа определенного типа, или же от национального разве¬дывательного управления с его шпионской сетью. Система может быть защи¬щена до тех пор, пока не появились некие новые достижения математики, или в течение некоторого определенного промежутка времени, или от определен¬ных типов атак. Как любое прилагательное, слово «безопасный» не имеет смысла вне контекста.

В первой части книги я предпринял попытку рассмотреть основы контекста безопасности. Я расскажу о возможных угрозах для цифровых систем, о типах атак

и о типах нападающих. Затем я расскажу о том, что желательно для системы безопасности. Я сделаю это прежде, чем обсуждать конкретные технологии, поскольку невозможно грамотно оценить технологии безопасности без знания ландшафта. Так же как вы не сможете понять, каким образом стены замка защищают его жителей, не погрузившись в атмосферу средневековья, вы не сможете понять, каким образом работает брандмауэр или Интернет с шифрованием данных вне контекста той среды, в которой они действуют. Кто такие нападающие? Чего они хотят? Какими инструментами располагают? Без базовых представлений о таких вещах невозможно разумное обсуждение понятия безопасности.

# Глава 2. Опасности цифрового мира

Этот мир — опасное место. Грабители подстерегают, чтобы наброситься на вас, когда вы идете по плохо освещенной аллее; искусные мошенники строят планы, как лишить вас денег, отложенных на старость; коллеги всеми силами стремятся разрушить вашу карьеру. Синдикаты организованной преступности распространяют коррупцию, наркотики и страх с поразительной эффективностью. В наличии также сумасшедшие террористы, ненормальные диктаторы и неконтролируемые остатки бывших супердержав, которые отличаются скорее взрывоопасностью, чем здравым смыслом. Кроме того, если верить газетам, есть еще монстры в пустынях, руки, торчащие из могил, от которых мороз пробирает, и ужасные инопланетяне, перевозящие детей Элвиса. Иногда удивительно все-таки, что мы прожили настолько долго, чтобы построить достаточно стабильное общество для проведения этого обсуждения.

Но мир — еще и безопасное место. Хотя опасности в индустриальном обществе вполне реальны, все же они — скорее исключения. В это иногда трудно верить в наш век сенсаций — газеты лучше продаются с заголовком «Трое прохожих случайно застрелены в перестрелке», чем «У двухсот семидесяти миллионов американцев небогатый событиями день» — но это правда. Почти все ходят каждый день по улице и не подвергаются нападению грабителей. Почти никто не умирает от случайного выстрела, не страдает от надувательства мошенников и, приходя домой, не сталкивается с сумасшедшими мародерами. Большинство компаний не являются жертвами вооруженного грабежа, жуликоватых банковских менеджеров или насилия на рабочих местах. Менее одного процента удаленных взаимодействий, позволяющих вести дела на расстоянии, без непосредственного контакта, приводят к жалобам какого-либо рода. Люди, как правило, честны: они обычно твердо придерживаются неписанных общественных правил. Законность в нашем обществе, в общем, на высоком уровне.

(Я осознаю, что предыдущий абзац представляет собой чрезмерное упрощение сложного мира. Я пишу эту книгу в Соединенных Штатах в конце второго тысячелетия. Я пишу ее не в Сараево, Хевроне или Рангуне. У меня нет опыта, исходя из которого я мог бы рассказать, каково жить в тех местах. Мои личные представления о безопасности происходят из опыта жизни при стабильной демократии. Эта книга о безопасности с точки зрения индустриального мира, а не мира, раздираемого на части войной, подавляемого тайной полицией или контролируемого криминальными структурами. Эта книга об относительно небольших опасностях в обществе, в котором главные угрозы уже отведены.)

Нападения, криминальные или нет, представляют собой исключения. Это со¬бытия, застающие людей врасплох, «новости» в прямом смысле слова. Они нару¬шают общественный уклад, разрушают жизнь тех, кто стал их жертвой.

## Неизменная природа атаки

Что отличает киберпространство от его двойника — реального мира в «плоти и крови», если отбросить технологические изыски и графические пользовательские интерфейсы? Как и физический мир, виртуальный мир населен людьми. Эти люди взаимодействуют с другими, образуют сложные социальные и деловые взаимоот¬ношения, живут и умирают. В киберпространстве существуют сообщества, боль¬шие и малые. Киберпространство наполнено коммерцией. Там заключаются со¬глашения и контракты, там случаются разногласия и конфликты.

И опасности в цифровом мире отображают опасности физического мира. Если в последнем существуют опасности хищений и растрат, то точно так же они суще¬ствуют и в первом. Если грабят физические банки, то ограбят и цифровые. Втор¬жение в частную жизнь — всегда проблема, независимо от того, имеет ли оно об¬лик фотографа с телеобъективом или хакера, который перехватывает сообщения частного характера. Правонарушения в киберпространстве включают в себя все, что вы можете видеть в физическом мире: воровство, рэкет, вандализм, страсть к подглядыванию и подслушиванию, эксплуатацию, вымогательство, мошенниче¬ство и обман. Присутствует даже опасность реального физического ущерба в ре¬зультате выслеживания, нападения на систему контроля воздушных перевозок и т. п. В первом приближении сообщество живущих в режиме онлайн такое же, как и сообщество тех, кто далек от компьютерных сетей. И, тоже в первом прибли¬жении, вторжения в цифровые системы будут такими же, как и нападения на их реальные аналоги.

Это означает, что, бросая взгляд в прошлое, мы можем точнее увидеть, что при¬несет будущее. Нападения будут выглядеть по-разному — взломщик будет опери¬ровать цифровыми взаимодействиями и точками входа в базы данных вместо от¬мычек и монтировок, террористы выберут мишенью информационные системы вместо самолетов — но мотивация и психология будут теми же. Это также значит, что нам не понадобится совершенно другая правовая система в будущем. Если наше грядущее похоже на былое — за исключением некоторых особенностей, — тогда правовая система, которая работала в прошлом, вероятно, будет работать и в буду¬щем.

Вилли Саттон грабил банки, потому что там хранились деньги. Сегодня деньги находятся не в банке, они перемещаются по компьютерным сетям. Каждый день банки мира переводят друг другу миллиарды долларов при помощи простого изменения чисел в компьютерной базе данных. Между тем, средняя величина единичной кражи из физических банков ненамного превышает 1500 долларов. И киберпространство будет представляться злоумышленникам все более собла¬знительным — объем электронной торговли возрастает с каждым годом.

Где деньги — там преступность. Ограбление банка или магазина спиртного, наде¬вание лыжной маски и размахивание пистолетом 45-го калибра не совсем устарело.

Но это — не тот метод, который предпочтет преступник, чтобы продумать план, если он совсем не одурел от наркотиков. Организованная преступность предпочитает нападать на крупные системы, чтобы получать большую добычу. Мошенничество с кредитными картами и контрольными системами стало с годами более изощренным. Жульничество с расчетными автоматами следует тому же принципу. Если мы не увидели до сих пор широко распространенного мошенничества с системой платежей в Интернете, то это потому, что там еще нельзя получить крупные деньги. Когда деньги там будут, преступники попытаются их украсть. И как видно из истории, они в этом преуспеют.

Нарушение конфиденциальности также не является чем-то новым. Удивительное множество юридических документов является достоянием общественности — это записи с общим доступом: имущественные сделки, продажа кораблей, гражданские и уголовные судебные разбирательства и приговоры, банкротство. Хотите узнать, кто владеет вон тем кораблем и сколько он за него заплатил? Это можно сделать. Такова роль общего доступа к базам данных. Еще больше приватной информации содержится в 20 000 (или около того) персональных баз данных (в США), которые находятся в корпорациях: финансовые подробности, медицинская информация, особенности образа жизни.

Сыщики (частные и полицейские) уже давно используют эти и другие данные, чтобы разыскивать людей. Для этого используют даже данные, которые по общему мнению считаются конфиденциальными. Ни один частный следователь из телесериала не обходится без друга в местной полиции, готового поискать имя, номер автомашины или уголовные записи в полицейских файлах. Полиция постоянно пользуется корпоративными базами данных. И каждые несколько лет какого-нибудь скучающего оператора информационно-поисковой системы ловят на выискивании налоговых деклараций знаменитостей.

Специалисты по маркетингу уже давно используют любые данные, которые могут заполучить, чтобы «вычислить» определенных людей. В Соединенных Штатах личные данные принадлежат не тому человеку, которого они характеризуют, а организации, собравшей их. Информация о ваших финансах не является вашей собственностью, она принадлежит вашему банку. Ваша медицинская информация принадлежит не вам, а вашему врачу. Врач дает клятву, что сохранит ваши личные секреты, но страховые агенты и этого не делают. Вы действительно хотите, чтобы любой знал о вашем пороке сердца или наследственной глаукоме? Как насчет вашего алкоголизма или неприятностей с венерическим заболеванием двадцатилетней давности?

Нарушение конфиденциальности может легко привести к мошенничеству. В рассказе «Бумажная луна» Джой Дэвид Браун пишет о трюках, к которым прибегали в период Великой депрессии для продажи Библий и родственных товаров, цены на которые внезапно упали. В другого рода аферы сейчас вовлечены матери и вдовы солдат, погибших на заокеанской войне: «Всего за несколько пенни в день мы позаботимся об их могилах» — и обманщики получают свой куш. Во многих частях страны коммунальные службы установили систему на основе телефонной связи, позволяющую считывать показания счетчиков: воды, электричества и т. п. Эта идея будет казаться грандиозной до тех пор, пока какие-нибудь предприимчивые преступники не начнут использовать ее для определения времени отъезда хо-

зяев в отпуск. Или пока они не догадаются использовать систему аварийного кон¬троля, которая с точностью до цента показывает еще и подробности аренды зда¬ния. Если данные могут быть использованы, кто-нибудь попробует это сделать — с компьютером или без него.

В киберпространстве нет ничего нового. Детская порнография — старая исто¬рия. Отмывание денег — уже видели. Причудливые культы, предлагающие веч¬ную жизнь в обмен на ваш чек, — даже надоело. Преступники не хуже и не лучше деловых людей понимают, для чего можно использовать сеть; они только переина¬чивают свои старые трюки под новые возможности, учитывая тонкие подробности и эксплуатируя размах сети и ее тенденцию к росту.

# Изменяющаяся природа атаки

Опасности могут быть теми же, но киберпространство все видоизменяет по-свое-му. Хотя нападения в цифровом мире могут иметь те же цели и использовать многие из тех методов, что и нападения в физическом мире, все же они будут существенно различаться. Они будут проще и шире распространены. Будет слож-нее проследить, поймать, доказать вину злоумышленника. И их действие будет более разрушительным. У Интернета есть три новых свойства, которые помога-ют осуществить атаку. Любое из них — угроза, все вместе они способны вселить ужас.

## Автоматизация

Автоматизация — это друг нападающего. Если умный фальшивомонетчик изоб-рел способ чеканки безукоризненных пятицентовых монет, никто не будет беспо-коиться. Фальшивомонетчик не сумеет сделать достаточно много этих монет, что-бы оправдать время и усилия. Телефонные хулиганы могли звонить бесплатно из телефонов-автоматов в пределах определенной местности практически сколько угодно с 1960-х до середины 1980-х. Конечно, телефонная компания была недо-вольна, что привело к грандиозным шоу с попыткой поймать этих людей, — но хулиганы не сильно повлияли на итоговую прибыль компании. В самом деле, вы не можете украсть столько десятицентовых телефонных звонков, чтобы это сказа-лось на доходе компании с многомиллиардным капиталом, особенно если себесто-имость услуг близка к нулю.

В киберпространстве все иначе. Компьютеры имеют неоспоримое преимуще-ство при решении повторяющихся, скучных задач. Наши фальшивомонетчики могут отчеканить миллион электронных пятицентовиков во время сна. Так назы-ваемая тактика «поэтапных нападений» — кража каждый раз небольшой части де-нег, по кусочкам, от каждого счета, приносящего процентный доход, — прекрас-ный пример того, что было невозможным без компьютеров.

Если вы спланировали крупную аферу, при помощи которой можно очистить чьи-нибудь карманы, а она срабатывает только один раз из 100 000 попыток, вы умрете с голоду, прежде чем ограбите кого-то. В киберпространстве вы можете настроить свой компьютер на поиск одного шанса из ста тысяч. Возможно, вы бу¬

дете находить по целой дюжине таких шансов ежедневно. Если вы в состоянии привлечь другие компьютеры, то сможете находить сотни шансов.

Быстрая автоматика совершает атаки, даже если возможное число успешных попыток мизерно. Атаки, которые были слишком несущественны, чтобы обращать на них внимание в физическом мире, могут быстро стать основной угрозой в цифровом. Многие коммерческие системы совершенно не заботятся об этих мелочах: дешевле игнорировать их, чем с ними бороться. Им придется думать иначе с приходом цифровых систем.

Киберпространство, кроме того, прокладывает новую просторную дорогу для нарушения конфиденциальности просто в результате факта появления автоматизации. Предположим, вы проводите маркетинговую кампанию, направленную на богатых любящих родителей, вместе с детьми коллекционирующих марки с изображениями пингвинов. Это слишком трудоемко — ходить по всему городу и находить богатых граждан с детьми, которые любят пингвинов и интересуются марками. Для компьютерной сети нет ничего проще, чем сопоставить маркетинговую базу данных почтовых индексов людей с определенным годовым доходом, записи о датах рождения, списки подписчиков rec.collecting.stamps[1] и данные покупателей книг о пингвинах на Amazon.com. Интернет дает в руки средства, позволяющие собрать все данные о человеке, когда-либо внесенные в пользовательскую сеть. Бумажные данные, даже если они общедоступны, трудно искать и трудно сопоставлять. Компьютеризованные данные найти существенно проще. Данные, внесенные в сеть, можно найти удаленно и сравнить с другими базами данных.

При определенных обстоятельствах получение этих данных незаконно. Частных лиц неоднократно преследовали в судебном порядке за просмотр секретных файлов полиции или информационно-поисковых систем. При других условиях это действие вполне обоснованно называется *добычей данных*. Например, большие компании, имеющие базу данных кредитных карт: Experian (раньше — TRW), TransUnion и Equifax, имеют горы сведений почти о любом человеке в США. Эту информацию собирают, сортируют и продают любому, кто готов за нее заплатить. Базы данных кредитных карт содержат ошеломляющее количество фактов о том, как люди привыкли тратить деньги: где они живут, что едят, как проводят отпуск — все это можно там найти. DoubleClick пытается построить базу данных, содержащую информацию об индивидуальных привычках веб-серферов[2]. Даже магазин бакалейных товаров ведет специальные карты постоянных покупателей, что позволяет получать данные о предпочтениях последних. Компания Acxiom специализируется на соединении информации частных и общедоступных баз данных.

Новым здесь является не то, что данные выплывают наружу, а то, как просто их можно собирать, использовать и злоупотреблять ими. И дела становятся все хуже: собирается все больше данных. Банки, авиалинии, каталоги компаний, фонды медицинского страхования — все они хранят информацию частного характера. Множество веб-сайтов собирают и продают персональные данные. А почему нет? Хранение данных дешево и, может быть, когда-нибудь пригодится. Такие разнообразные

[1] Группа рассылки новостей в сети USENET, посвященная коллекционированию марок. — *Примеч. ред.*

[2] Путешественники по Интернету. — *Примеч. ред.*

архивы в конце концов появляются в общедоступной сети. И все больше и больше данных оказывается собрано и снабжено перекрестными ссылками. Автоматизация переводит добычу информации на новую ступень.

## Действие на расстоянии

Как любят подчеркивать специалисты по технологиям, Интернет не имеет границ или естественных ограничений. Любые две вещи одинаково тесно связаны, будь они расположены в разных концах комнаты или планеты. Одинаково просто активизировать работу компьютера в Тулузе с компьютера в Тунисе и с компьютера в Таллахассии (Tallahassee). Не нравятся законы о цензуре или законы о компьютерных преступлениях в вашей стране? Найдите страну, которая вам нравится больше. Страны вроде Сингапура пытались ограничить возможности своих граждан в отношении поиска в Сети, но строение Интернета делает задачу блокировки его отдельных частей неосуществимой. По мнению Джона Гилмора, «цензура Интернета — это его повреждение и разгром».

Это означает, что нападающим в Интернете не нужно находиться где-то рядом со своей добычей. Нападающий может сидеть за компьютером в Санкт-Петербурге и атаковать компьютер Ситибанка в Нью-Йорке. Такое изменение положения вещей породило гигантские последствия для безопасности. Раньше, если вы строили товарный склад в Буффало, вам приходилось беспокоиться только о преступниках, которые могли бы приехать в Буффало и вломиться в ваш склад. С тех пор как благодаря Интернету все компьютеры стали равноудалены от любого другого компьютера, вам надлежит принимать во внимание преступность всего мира.

Глобальная природа Интернета также затрудняет поиск преступников и их обвинение. Найти нападающих, ловко скрывающих свое местонахождение, может быть почти невозможно, и, если вы даже найдете их, что будете делать? Преступность ограничена только в том, что касается политических границ. Но если у Интернета не существует физической территории, на которой можно было бы его контролировать, то кто обеспечит его безопасность?

К настоящему времени все правоохранительные органы, которые могли бы предъявить претензии к Интернету, уже пытались это сделать. Данные пришли из Германии? Тогда это в юрисдикции немецких законов. Они адресованы в Соединенные Штаты? Тогда это дело американского правительства. Они проходили через Францию? Если так, французские власти ответят *qu'il s'est passe*[1]. В 1994 году операторов компьютера BBS в Милпитасе, штат Калифорния, — где находились и люди и компьютеры — судили и признали виновными в суде Теннесси, потому что кто-то в Теннесси сделал междугородный телефонный звонок в Калифорнию и загрузил порнографические картинки, которые, как обнаружилось, разрешены в Калифорнии, но неприличны в Теннесси. Операторы BBS никогда до судебного процесса не бывали в Теннесси. В июле 1997 года 33-летняя женщина была осуждена швейцарским судом за отправку порнографии через Интернет — хотя с 1993 года она жила в США. Имеет ли это какой-то смысл?

[1] Так это уже прошло *(фр.)*. — *Примеч. ред.*

Тем не менее обычно преследование судебным порядком невообразимо трудно. До того как их «вычислят», преступники могут использовать неразбериху в качестве ширмы. В 1995 году 29-летний хакер из Санкт-Петербурга заработал 12 миллионов долларов, вломившись в компьютер Ситибанка. Ситибанк случайно обнаружил взлом и вернул себе большую часть денег, но встретил огромные сложности с выдачей хакера, чтобы подвергнуть его суду.

Такая разница в законах между различными штатами и странами может даже привести к высокотехнологичной форме выбора области юрисдикции. Иногда это работает в пользу обвинителя, как, например, осуждение в Теннесси калифорнийской BBS. В других случаях она идет на пользу преступникам: преступному синдикату, у которого достаточно денег, чтобы предпринять широкомасштабную атаку на финансовую систему, не так трудно найти страну, у которой слабые законы о компьютерной преступности, несомненно продажные офицеры полиции и никаких договоров о выдаче преступников.

## Распространение технических приемов

Третье свойство — это легкость, с которой опыт удачных атак распространяется по киберпространству. HBO — компанию, занимающуюся кабельным телевидением, не очень заботит, что кто-нибудь может создать дешифратор для их базы. Для этого требуется время, сноровка и некоторое количество денег. Но что, если этот «кто-нибудь» опубликует простой метод получения любым человеком бесплатного спутникового телевидения? Без всякой работы? Безо всякого оборудования. «Просто наберите эти семь цифр на вашем пульте дистанционного управления, и вам никогда больше не придется платить за кабельное телевидение». Это может увеличить количество пользователей, не производящих платежи, до миллионов и ощутимо повлиять на прибыльность компании.

Физические фальсификации — это сложная, но решаемая задача. Двадцать лет назад мы продавали шаху Ирана какие-то старые машины глубокой печати. Когда к власти пришел аятолла Хомейни, он сообразил, что гораздо выгоднее печатать стодолларовые банкноты, чем иранские риалы. ФБР называет эти подделки супернзнаками, поскольку они почти безупречны. (Вот почему в США изменили дизайн денег.) В то время как ФБР и секретные службы заламывали руки, Министерство финансов произвело некоторые подсчеты: иранские печатные станки могут печатать только какое-то количество денег в минуту, в году только столько-то минут, то есть можно посчитать максимальное количество фальшивых банкнот, которое иранцы в силах выпустить. Казначейство решило, что это количество фальшивок не повлияет на денежную стабильность, так что не стоит серьезно беспокоиться о национальной безопасности.

Если бы производство фальшивых денег использовало электронику, все было бы по-другому. Электронный фальшивомонетчик может автоматизировать процесс, написать программу и поместить ее где-нибудь на одном из веб-сайтов. Люди будут загружать эту программу и запускать необнаруживаемое производство фальшивых электронных денег. К полудню эта информация могла бы оказаться в руках первых 1000 фальшивомонетчиков; следующие 100 000 получили бы ее в течение недели. Денежная система США смогла бы разрушиться за неделю. Уже не

существовало бы максимального предела усиления мощности этой атаки — в киберпространстве она может возрастать экспоненциально.

Интернет представляет собой еще и совершенную среду для распространения удачных приемов нападения. Только первому нападающему приходится быть изобретательным, все остальные могут просто использовать его программы. После того как автор изобретения включает их в какой-нибудь архив, удобно расположенный где-нибудь в экономически отсталых странах, любой способен загрузить и использовать их. И, однажды выпущенные в свет, они уже не поддаются контролю.

Мы рассматривали эту проблему на примере компьютерных вирусов: дюжина сайтов позволяет вам загрузить компьютерные вирусы, наборы инструментов для их конструирования и модели самих вирусов. Ту же картину мы видим с хакерскими инструментами: комплекты программного обеспечения, которые вызывают поломку компьютеров, крушат серверы, обходят защиту от копирования или используют недостатки конструкции браузеров, чтобы красть данные из машин пользователей. Вирусы Интернета всегда порождают вирусы на гибких дисках, что выглядит как необычная и привлекательная забава. Организация атак вида «отказ в обслуживании», направленных на основные веб-сайты, как в 2000, не требует никакой изобретательности: достаточно загрузить и запустить программу-сценарий. И поскольку электронные коммерческие системы широко распространены, мы видим, что автоматизированные нападения направлены и на них тоже.

Компьютерные атаки доказывают то, что преступникам не нужно обладать сноровкой, чтобы преуспеть.

## Упреждающие меры вместо ответных

Традиционно коммерческие системы бегали наперегонки с мошенничеством: оперативная, в режиме онлайн, проверка кредитных карт была лишь ответной мерой на участившиеся кражи последних. Этот метод не сработает в Интернете, потому что время в нем течет очень быстро. Кто-то может провести успешную атаку на систему кредитных карт Интернета, написать программу, чтобы автоматизировать нападение, и за 24 часа программа станет доступна для полумиллиона людей по всему миру, и большинство из них невозможно будет потом ни в чем обвинить. Я представляю себе консультанта по безопасности, который входит в офис главного администратора и говорит: «У нас есть два варианта. Мы можем принять любую транзакцию как действительную, будь она законной или мошеннической, или не принимать ни те, ни другие». Главный администратор окаменел бы перед необходимостью такого выбора.

# Глава 3. Атаки

Я собираюсь обсудить три основных класса атак. Преступные атаки относятся к наиболее явным, и этому типу я уделю основное внимание. Но два других класса — атаки, приводящие к огласке, и «законные» атаки — могут быть гораздо более разрушительными.

## Преступные атаки

Что такое преступные атаки, понять просто: каким образом, атакуя эту систему, я смогу получить максимальную финансовую выгоду? Нападающие бывают разные: от преступников-одиночек до изощренных синдикатов организованной преступности, от «своих людей», решивших заработать «быстрых денег», до правительств, нацеленных на борьбу с инфраструктурой другой страны.

## Мошенничество

Мошенничать пытались со всеми когда-либо изобретенными коммерческими системами. Недобросовестные торговцы пользовались неправильными весами для того, чтобы обсчитывать покупателей; те же, в свою очередь, соскабливали серебро и золото с ободков монет. Подделывали все: деньги, акции, кредитные карты, чеки, аккредитивы, заказы на поставку, фишки казино. Современные финансовые системы — чеки, кредитные карты и сети банковских автоматов — ежегодно терпят многомиллионные потери из-за мошенничества. Электронная торговля не будет ничем отличаться, не изменятся и методы преступников.

## Аферы

Национальная компьютерная лига выделила пять наиболее распространенных в сети афер — это продажа интернет-услуг, продажа разнообразных товаров, аукционы, пирамидальные и многоуровневые маркетинговые схемы и благоприятные возможности для деловой деятельности. Люди читают какое-нибудь соблазнительное письмо в электронной почте или посещают привлекательный веб-сайт и отсылают деньги в какой-то абонентский ящик, и все заканчивается тем, что они или не получают ничего взамен, или получают ничего не стоящий хлам. Все очень похоже на физический мир: множество людей просто обмануто.

## Разрушительные атаки

Разрушительные атаки — это сфера деятельности террористов, наемных служащих, склонившихся к мести, или хакеров, ушедших в подполье. Разрушение — это преступная атака (редко бывает, чтобы разрушение чужого имущества было законным), но часто при этом отсутствуют корыстные мотивы. Вместо них нападающие просто задают себе вопрос: «Каким образом я могу вызвать наибольшие повреждения, нападая на эту систему?»

Существует много разновидностей разрушительных атак. В 1988 году кто-то написал компьютерный вирус, направленный непосредственно на компьютеры, принадлежащие Electronic Data System. Он не произвел слишком больших разрушений (в действительности нанес небольшой ущерб NASA), но цель была именно такова. В начале 2000 года мы наблюдали массовые атаки, приводящие к отказам в обслуживании, на Yahoo!, Amazon.com, E*Trade, Buy.com, CNN и eBay. Умелый нападающий, вероятно, может неделями подавлять работу провайдера. Фактически хакер, обладающий достаточным мастерством и подходящими моральными устоями, может, вероятно, «разрушить» весь Интернет.

На другом конце спектра — те, кто открыто вламывается через парадную дверь с бомбой. Нападения США на иракские системы коммуникаций в Персидском заливе, по-видимому, лучший тому пример. Группа французских террористов Comitee Liquidant ou Detournant les Ordinateurs (Комитет компьютерной ликвидации и сдерживания) в начале 1980-х разбомбила компьютерные центры в районе Тулузы. Более эффектным было лишь сожжение Александрийской библиотеки в 47 году до н. э. (Юлием Цезарем), в 391 году н. э. (христианским императором Феодосием I) и в 642 году н. э. (Омаром, халифом Багдада) — все это, кстати, превосходные уроки, наглядно демонстрирующие важность резервного копирования.

## Кража интеллектуальной собственности

Интеллектуальная собственность — это не только производственные секреты и базы данных компаний. Это также электронные версии книг, журналов и газет, цифровое видео, музыка и фотоснимки, программные средства и частные базы данных с платным доступом. При этом сложность задачи заключается не в сохранении права собственности на данные, а в том, как организовать контроль, чтобы получать соответствующие выплаты за данные прежде, чем они станут общедоступными.

Компании, занимающиеся программным обеспечением, хотят продавать свои программы легальным покупателям, избегая того, чтобы пираты изготовили миллионы нелегальных копий и продали (или раздали) их всем желающим. В 1997 году Альянс коммерческого программного обеспечения (Business Software Alliance) установил на своей веб-странице счетчик, который показывал, какие потери несет индустрия из-за пиратства: 482 доллара в секунду; 28 900 долларов в минуту; 1,7 миллиона долларов в час; 15 миллиардов долларов в год. Эти цифры завышены, так как сделано ложное допущение, что всякий, кто пользуется пиратской копией, например 3D Studio Max компании Autodesk, заплатил бы за легальную продукцию 2995 долларов (или 3495 долларов, если брать розничную цену). Рас¬

пространенность пиратства в сфере программного обеспечения сильно зависит от страны: считается, что 95% программ в Китайской Народной Республике — пиратские, тогда как в Канаде таких только 50%. (Лидирует Вьетнам с 98% пиратского программного обеспечения.) Эти потери выводят из себя компании, занимающиеся программными средствами.

Пиратство может принимать различные масштабы. Это и распространение дисков среди друзей, и загрузка из Интернета, и крупномасштабные операции по изготовлению копий (обычно проводящиеся на Ближнем Востоке).

Пиратство также может быть совершено по отношению к данным. Подделывают ли музыкальные диски, которые продают в закоулках Бангкока, или MP3-файлы с той же музыкой, которые распространяют через Веб, — электронную интеллектуальную собственность воруют все время. (И, конечно, это в полной мере относится к цифровым изображениям, цифровому видео и электронному тексту.)

Общая нить этих рассуждений состоит в том, что компании хотят контролировать распространение своей интеллектуальной собственности. Такая позиция, хотя она совершенно обоснована, противоречит тому, что, собственно, представляет собой электронный мир. Последний имеет свои особенности: в отличие от физических вещей, информация может находиться в двух местах одновременно. Ее можно копировать бесконечно. Кто-то может отдавать часть информации и одновременно сохранять ее. Когда информация распространится повсюду, может стать невозможным проследить ее местонахождение. Если цифровая копия *Короля льва* когда-нибудь будет распространяться по Интернету, Дисней не сумеет удалить все копии.

Недозволенное копирование — не новая проблема; она так же стара, как индустрия звукозаписи. Когда я учился в школе, у меня были магнитофонные записи музыки, которые были мне не по карману; и точно так же поступали все остальные мои знакомые. Тайвань и Таиланд долгое время были источником поддельных CD. Русская мафия развивает индустрию видеопиратства, а китайские триады все в большей степени вовлекаются в подделку программных средств. Ежегодные промышленные потери были оценены в 11 миллионов долларов в год, хотя это число, возможно, тоже основывается на ложных предположениях.

У электронных данных нет никакого волшебного иммунитета против подделок. Фактически они уникальны в том смысле, что их можно копировать совершенно точно. В отличие от моих кассет, качество незаконных цифровых видеодисков с *Королем Львом* или программных продуктов не ухудшается — это еще один оригинал. Препятствовать такому распространению — это все равно, что пытаться осушить воду, — попробуйте сами.

## Присвоение личности

Зачем красть у кого-то, когда вы просто можете стать этим человеком? Это существенно проще и может оказаться намного более выгодным: получить стопку кредитных карточек на чужое имя, наделать крупных долгов и затем исчезнуть. Это называется *присвоением личности* и является высокоразвитой сферой деятельности преступников. В Альбукерке (Нью-Мексико) преступники проникали в дома именно для того, чтобы забрать чековые книжки, свидетельства о кредитных кар¬

тах, квитанции и другие финансовые документы, выискивая номера социального обеспечения, даты рождения, места работы и номера счетов.

Такие вещи происходят все время. В США в течение одного только 1999 года сообщалось о тысячах случаев присвоения личности. Устранение последствий может стать тяжелой и изнурительной задачей.

Положение становится все хуже. По мере того как установление личности приобретает электронную форму, процедура присвоения упрощается. В то же время, поскольку все больше систем используют электронную идентификацию, присвоение становится все более выгодным и менее опасным. Зачем вламываться в чей-то дом, если вы можете оперативно собрать необходимую информацию о личности?

А жертвы беспомощны. Они сообщают засекреченную информацию любому, кто спрашивает, многие пишут номера водительских прав на чеках. Они выбрасывают счета, банковские квитанции и т. п. Они слишком доверчивы.

В течение долгого времени мы могли не опасаться за системы удаленной идентификации. Но теперь стало понятно, что «девичья фамилия матери» не может уже являться ее элементом (особенно сегодня, когда эта фамилия зафиксирована в исчерпывающей открытой базе данных на генеалогическом веб-сайте), хотя она и использовалась достаточно долго и за это время преступники неплохо погрели на ней руки. Сейчас это уже ушло в историю, и, смею надеяться, мы уже никогда не будем столь наивны.

## Кража фирменной марки

Идентификация в виртуальном пространстве жизненно важна и для фирм, и для отдельных индивидуумов. Разработка фирменного знака требует времени и денег. Такого рода идентификация — это больше чем надпись, лозунг и навязчивая мелодийка для рекламы. Это — продукция, кирпичные здания, работники службы по работе с покупателями, вещи, которые можно потрогать, люди, с которыми можно поговорить. Марка означает репутацию.

В Интернете препятствия минимальны. Любой может иметь веб-сайт — от Ситибанка до Фреда-храни-деньги-в-матрасе. И многие так и делают. Как пользователи узнают, какие сайты стоит посетить, на каких — сделать закладку, а с какими установить взаимоотношения? Тысячи компаний продают компьютеры в Сети. Какие из них действительно надежны, а какие не заслуживают доверия?

Снабжение товара торговой маркой — единственное, что может здесь помочь. Когда Веб был впервые представлен на публичное обозрение, специалисты утверждали, что он возвещает конец big brand[1]. Поскольку любой мог войти в Сеть и соревноваться с узнаваемыми брендами, торговые марки становились бессмысленными. Действительность доказала полную противоположность этому предположению: поскольку любой может войти в Сеть и соперничать с громкими именами, единственный способ отличить продукцию — ее фирменный знак. Пользователи запоминают знак и возвращаются на сайты, которым они доверяют. Знаки имеют реальную ценность, и их имеет смысл воровать.

[1] Торговая марка, пользующаяся широкой известностью. — *Примеч. ред.*

Например, малайзийская компания хотела продавать презервативы, используя торговую марку Visa. Малазийцы заявили, что не имеют ничего общего с компанией кредитных карт, но эта «игра слов» дает им своего рода пропуск. Провести настоящую Visa не удалось — компания предъявила иск. Она победила, и я считаю, что такой прецедент имеет важное значение для защиты права собственности на фирменный знак.

Киберпространство предоставляет широкие возможности для кражи фирменного знака. В 1998 году кто-то подделал запрос на передачу имени домена в Network Solution и украл Sex.com; настоящий владелец до сих пор еще пытается вернуть его назад[1]. В другом случае некий сантехник заменил телефонный номер на сайте другого такого же специалиста на свой. Преступные синдикаты в Лас-Вегасе сделали подобный трюк с телефонными номерами эскорт-услуг. Атаки такого рода не представляют собой ничего нового. Элмон Строугер был предпринимателем из Канзас-Сити. Он был убежден, что телефонисты перенаправляют звонки, поступающие к нему, в фирмы-конкуренты, поэтому в 1887 году изобрел телефонный диск, дабы исключить телефонистов.

Некоторые знатоки устраивают свой веб-сайт таким образом, чтобы красть трафик с других веб-сайтов; это известно как редирект, перенаправление *{page-jacking)*. В Сети есть и *майпсквоттеры,* «пираты» опечаток, регистрирующие имя домена, которое выглядит как имя оригинального веб-сайта, написанное с опечаткой. Так поступают создатели многих порносайтов. Даже большие компании не чуждаются такого рода приемов: когда стал популярен 1-800-COLLECT, компании MCI и AT&T создали свою службу телефонных разговоров, оплачиваемых абонентом, которому звонят. У AT&T ею стала 1-800-C0LLECT (нуль вместо буквы «О» - наиболее распространенная ошибка набора), а MCI опустился до того же, зарегистрировав 1-800-0PERATOR с тем же нулем вместо буквы «О», но в другом месте. Некоторые подобные приемы сегодня запрещены, и я надеюсь, что в будущем запретят и остальные.

## Судебное преследование

К сожалению, судебное преследование в киберпространстве может быть затруднено. Казалось бы, преступления точно такие же — ведь кража противозаконна: хоть аналоговая — хоть цифровая, хоть на линии — хоть без нее. Набор действий тоже одинаков: злоупотребление, подделка, рэкет, мошенничество — как и квалификация, и преследование по закону в соответствии с уголовным кодексом. Законы против таких поступков (вместе с инфраструктурой уголовного правосудия для того, чтобы претворять их в жизнь) уже есть. Были приняты и некоторые новые законы, специально для цифрового мира, но мы пока не знаем всех их тонкостей. Но темп работы судебной системы отличается от темпа развития Интернета, и США

[1] Предприимчивый Гари Кремен из Сан-Франциско зарегистрировал домен Sex.com в 1994 году. Еще более предприимчивый Стивен Майкл Коэн сфабриковал фальшивое письмо в VeriSign, в результате чего притягательное имя было передано во владение Коэна. Гари Кремен вышел из зала суда с победой, но отсуженных 65 миллионов долларов так и не получил. Тогда он пустился во все тяжкие и включил в следующий процесс нового оппонента — VeriSign. И выиграл бы, если бы в 1994 году была принята практика платить за регистрацию домена. В результате процесс вокруг секс-домена тянется до сих пор. — *Примеч. ред.*

может потребоваться десятилетие на то, чтобы упразднить плохо работающий закон или понять, как на самом деле нужно его применять.

Через какое-то время законы будут лучше отражать действительность цифрового мира. Несколько лет назад, когда поймали группу немецких хакеров, взломавших компьютерные системы США, у немецкого правительства вовсе не было уголовных законов, по которым их можно было бы осудить. Сегодня уже некоторые уголовные кодексы расценивают как преступление поступки вроде взлома удаленных компьютерных систем, тогда как старые законы о посягательстве не очень-то касались правонарушителей, удобно сидящих в своих спальнях, в то время как команды, поступающие по телефонной сети с их компьютера, совершают правонарушения. Постепенно законы о преследовании, вторжении в частную жизнь, авторском праве и подстрекательстве приспосабливаются к условии мира, в котором вещи работают не совсем так, как все привыкли.

В конечном счете люди осознают, что нет смысла писать законы, которые специфичны для какой-либо технологии. Мошенничество есть мошенничество вне зависимости от того, что является средством — почта, телефон или Интернет. Правонарушение не становится более или менее преступным, если в нем используется криптография. (Нью-Йоркские продавцы, которые в 1999 году при помощи Palm Pilot[1] копировали номера кредитных карт покупателей, не были бы менее виновны, если бы использовали ручку и бумагу.) И вымогательство не становится лучше или хуже, если его осуществляют с использованием компьютерного вируса, а не старомодных компрометирующих фотографий. Хорошие законы написаны так, чтобы не зависеть от технологии. В мире, где технологии движутся вперед намного быстрее, чем заседания Конгресса, только они и могут работать. Механизмы для более быстрого и чуткого законодательства, может быть, когда-нибудь и появятся.

## Вмешательство в частные дела

Вмешательство в частные дела не обязательно будет преступлением, но может им стать (оно, например, может быть прелюдией к присвоению личности). В США почти все такие вторжения законны. Людям не принадлежит информация о них самих. Если кредитное бюро или фирма, занимающаяся маркетинговыми исследованиями, собирают о вас информацию — о ваших личных привычках, схеме покупок, финансовом статусе, физическом здоровье, — они вправе продать сведения любому, кому захотят, без вашего ведома и согласия. Впрочем, это везде по-разному. Законы о неприкосновенности частной жизни в большей части Европы (охватывающей Европейский Союз), на Тайване, в Новой Зеландии и Канаде более суровы.

Другие типы вмешательства в частные дела тоже считаются законными. Когда нанимают частного детектива, чтобы собрать информацию о человеке или компании, это законно до тех пор, пока он не использует никаких противоправных методов. Любого рода вмешательство полиции в частную жизнь законно при наличии ордера, но многое законно и без него. (Знаете ли вы, что полиции США не нужен

[1] Карманный компьютер, созданный корпорацией 3Com. — *Примеч. перев.*

ордер, чтобы потребовать сделать копии с фотопленок, которые вы сдали на проявку?)

Существует два принципиально разных типа вторжений в личную жизнь: направленная атака и сбор информации. При направленной атаке злоумышленник хочет узнать все об Алисе. Если Алиса — человек, то атака называется преследованием. Если «Алиса» — компания, то это производственный шпионаж. Если «Алиса» — правительство, то такую атаку называют национальной разведкой или шпионажем. Любая из этих атак доведет вас до тюрьмы при использовании одних методов и не приведет к подобному результату при использовании других.

Компьютерные меры безопасности могут защитить Алису от направленной атаки, но только до определенного момента. Если злоумышленников достаточно хорошо финансируют, они всегда их обойдут. Например, установят жучок в офисе Алисы, перероют ее мусор или будут следить за ней при помощи телескопа. Информация есть информация, и компьютерные меры безопасности защищают ее, пока она в компьютере. Такие меры работают против бесконтактных атак. Они заставляют злоумышленников подбираться поближе к Алисе и делают вторжение в частную жизнь более опасным, дорогим и попадающим в сферу действия разных неприятных законов.

*Сбор информации* — это другой тип вторжений в личную жизнь. Такие атаки используют косвенные методы. Предположим, злоумышленнику нужен список всех вдов не моложе 70 лет, у которых в банке не менее миллиона долларов, которые за прошлый год участвовали более чем в 8 благотворительных акциях и подписываются на астрологический журнал. Или список всех жителей США, подписанных на AZT[1]. Или тех, кто посещал определенный социалистический веб-сайт. Аферисты сто лет собирали бы имена людей, которые могли бы поддаться определенному жульничеству, в то время как широкая распространенность баз данных в Интернете позволяет им автоматизировать и оптимизировать их поиск.

Хорошее шифрование и компьютерная безопасность в состоянии помочь защититься от атак по сбору информации (если считать, что это незаконно — просто купить информацию у того, кто владеет различными базами данных), делая затруднительной процедуру сбора. Сбор информации выгоден только потому, что его можно автоматизировать; нет никакого смысла перерывать всю мусорную корзину соседа, чтобы найти демографические данные. Если же все компьютерные данные защищены, злоумышленник просто не знает, где ему смотреть. Даже не самые высокие уровни криптографии могут полностью защитить от атак по сбору информации.

## Наблюдение

Сто лет назад у каждого была частная жизнь. Вы с приятелем могли выйти в чистое поле, убедиться, что больше никого поблизости нет, и общаться на таком уровне конфиденциальности, который теперь навсегда утрачен. Витфилд Диффи ска¬

[1] AZT — препарат, использующийся для ВИЧ-терапии. Был разработан в 1960-х годах как химиотерапевтическое средство для больных лейкемией. Подавался как панацея, но, похоже, не оправдал ожиданий. — *Примеч. ред.*

зал : «Ни о каком праве на конфиденциальные переговоры в Конституции не упоминается. Я не думаю, что это кому-либо приходит в голову, тогда как это право необходимо защищать». Возможность иметь конфиденциальный разговор, как возможность держать в голове собственные мысли и возможность упасть на землю, если толкнут, естественно проистекала из устройства мира.

Технология разрушила такое мировоззрение. Мощные направленные микрофоны могут улавливать разговоры на расстоянии сотен ярдов. После того как группа террористов MRTA захватила японское посольство в Перу (в 1997 году), в репортажах рассказывали о «жучках», спрятанных в пуговицах рубашек, которые позволили полиции установить местонахождение каждого. Устройства Ван Эйка могут читать текст на мониторе вашего компьютера, расположенного через квартал от них (пока еще это дорогая и сложная атака, но вы только подождите, пока распространятся беспроводные локальные сети). Камеры наблюдения, которые сейчас легко заказать по каталогам электроники, можно запрятать в самые маленькие щели; спутниковые камеры могут прочитать номерной знак вашего автомобиля прямо с орбиты. Министерство обороны проектирует миниатюрные летательные аппараты размером с небольшую птицу или бабочку, которые смогут разыскать вражеских снайперов, определить местонахождение заложников в захваченном здании или шпионить за кем-нибудь.

Возможность проследить за кем-нибудь издалека уже существует, но ее используют только в исключительных обстоятельствах (не считая телевидения). В 1993 году крупный колумбийский наркоделец Пабло Эскобар был обнаружен отчасти благодаря тому, что проследили его сотовый телефон: метод, известный как *точное определение местонахождения*. В 1996 году русские военные ликвидировали чеченского лидера Джохара Дудаева при помощи ракеты «воздух-земля» после того, как определили местонахождение, перехватив его разговоры по спутниковому телефону. Сотрудники ФБР обнаружили грузовик, который принадлежал людям, взорвавшим федеральное здание в Оклахома-Сити, собрав записи всех камер наблюдения в городе и сопоставив их показания по времени[1]. Невидимые идентификационные метки печатаются фактически на всей цветной ксерографической продукции любого производства. (Эти машины также обеспечивают меры защиты от подделки — такие, как дополнительный голубой тонер на банкнотах, если устройство определяет, что пытаются копировать американские деньги.)

Технология, которая позволяет автоматически определять наркодельцов, анализируя случайные телефонные разговоры, по подозрительному поведению спутниковой связи или с помощью уличной камеры по фотографиям преступников, находящихся в розыске, еще не стала обычным делом, но это только вопрос времени. Опознание лиц сможет выделить из толпы конкретного человека. Распознавание голоса позволит просканировать миллионы телефонных разговоров в поисках определенного субъекта; уже можно выискивать подозрительные слова и фразы и выделять один разговор из большого их количества. Закон Мура, который прогнозирует, что промышленность может удваивать вычислительную мощность процессоров каждые 18 месяцев, предсказывает кое-что еще: вычислительная техника следующего поколения будет меньше размерами, быстрее, намного дешевле и дос-

[1] Это заняло 1,5 часа. — *Примеч. ред.*

тупнее[1]. Как только технология распознавания научится идентифицировать людей, компьютеры смогут проводить их поиск.

Хранение данных тоже становится дешевле. Всего несколько поколений отделяет нас от возможности записать всю нашу жизнь — на аудио и видео — и сохранить информацию. О необходимости этого могут говорить как о защитном механизме, «на случай если вам когда-нибудь потребуется алиби», или как о полезном для общества механизме, поскольку «вы никогда не знаете заранее, что можете стать свидетелем преступления». Когда-нибудь, возможно, «неношение» устройства, записывающего вашу жизнь, будет казаться подозрительным.

Инфраструктура наблюдения устроена в Соединенных Штатах под видом «службы работы с клиентами». Кто не слышал повсеместного сообщения, что «этот разговор может прослушиваться или записываться в целях гарантии качества»? Некоторые отели заносят данные о предпочтениях клиентов в международную базу данных, так что клиенты будут чувствовать себя как дома, даже если это их первая остановка в данном городе. В ресторанах высокого класса сейчас есть видеокамеры в обеденном зале для того, чтобы изучать пристрастия в еде и следить за ходом обеда, и базы данных, содержащие сведения о предпочтениях клиентов. Amazon.com отслеживает поведение покупателей из различных демографических групп. Дэвид Смит, написавший вирус «Мелисса», был обнаружен, поскольку Microsoft Word автоматически вводит информацию об авторстве во все документы. Автоматические системы оплаты сохраняют сведения о том, какие машины проехали через шлагбаум. В 2000 году некоторые города стали измерять загруженность дороги, отслеживая разговоры водителей по сотовым телефонам. Грань между хорошими потребительскими услугами и преследованием очень тонка.

Иногда речь не идет о потребительских услугах: компании кредитных карт имеют подробные записи о покупках, поэтому могут предотвращать мошенничество. Компании контролируют посещение служащими веб-сайтов, чтобы ограничить злоупотребления и задолженность. Многие аэропорты фиксируют номерные знаки всех автомобилей, помещенных на стоянку, — международный аэропорт Денвера в целях безопасности записывает номера всех машин, въезжающих на территорию аэропорта.

GPS (спутниковая система определения координат) — это фантастическая технология наблюдения. По крайней мере две компании поставляют на рынок «умные» автомобильные локаторы на основе GPS. Некая компания продает автоматические системы складского учета, использующие GPS и передатчики, прикрепляемые к объекту. Передатчики сигнализируют о своем местоположении, а центральный компьютер следит, где что находится. Вероятно, у шпионов давно уже была возможность использовать такого рода игрушки, но сейчас это — предмет потребления, поэтому папа знает, где его сын берет машину.

Секретность личной жизни расползается по швам. Это происходит малозаметно, и никто не высказывает протеста. Но все меньше и меньше конфиденциальности доступно, и большинство людей совершенно не обращает на это внимания.

---

[1] Приведена нижняя граница. Верхняя — 2 года. Другая часть закона Мура говорит, что на разработку и организацию серийного выпуска процессоров удвоенной производительности компании вынуждены затрачивать на порядок большие денежные средства. — *Примеч. ред.*

Устройства слежения становятся дешевле, меньше размерами и распространяются повсеместно. Похоже, что скоро мы будем жить в мире, где конфиденциальность будет невозможна нигде и никогда.

## Базы данных

Исторически нарушение конфиденциальности касалось только наблюдения. Затем в 1960-е годы общество достигло критической черты. В бизнес пришли компьютеры с большими базами данных, и организации стали накапливать базы данных, содержащие информацию об отдельных людях. Недавно мы достигли второй критической границы: компьютеры, объединенные в сеть, позволяют совместно использовать, сравнивать и объединять отдельные базы данных. Значение подобных действий для скрытости частной жизни мы можем ощущать все время. Мы ухитрились успешно преодолеть Большого Брата только для того, чтобы затеряться в сетях Младших Братьев. Для начала можно за кем-нибудь незаметно понаблюдать.

В последнее время собирают и сохраняют все больше и больше данных — из-за того, что сбор информации стал дешевле, и благодаря тому, что люди оставляют больше электронных «отпечатков» в повседневной жизни. Большинство таких данных собирают и сравнивают друг с другом. Многое доступно в режиме онлайн. Отсюда вывод: собрать на кого-нибудь подробное досье совсем несложно.

Многие базы данных — коммерческие: огромные кредитные базы данных, принадлежащие Experian, TransUnion, Equifax; телефонные базы данных, фиксирующие отдельные звонки; базы данных кредитных карт, содержащие информацию о личных покупках. Информацию можно использовать по прямому назначению или продать для каких-то других целей. В этих случаях доступ к ней законен, но потенциальный доступ к информации имеется и у тех, кто достаточно умел, чтобы взломать компьютер. Можно сравнивать между собой базы данных: информацию о вашем здоровье, финансовые подробности, любые сведения об образе жизни, которые вы когда-либо предавали огласке. В 1999 году в прессе была небольшая перепалка из-за того, что некоторые общественные каналы телевидения продали список жертвователей Демократической партии. В 2000 году общественный протест вынудил DoubleClick отказаться от своих планов по сопоставлению записей веб-серфинга отдельных личностей.

Веб дает больше возможностей для вторжения в частную жизнь. Электронные магазины могут, теоретически, сохранять записи обо всем, что вы купили (Blockbuster, например, фиксирует в базе данных все видео, которые вы брали напрокат). Также можно сохранять записи обо всем, на что вы просто смотрели: любой предмет, по которому вы просили показать более полную информацию, любую тему, по которой вы искали данные, как долго вы изучали каждый вопрос, а также не только что вы купили, но и на что вы смотрели и не купили.

Оперативные базы данных оказывают огромную помощь полиции по обеспечению правопорядка — они действительно помогают автоматически получить оперативные сведения и фотографии прямо в патрульную машину, — но остается опасность нарушения конфиденциальности. Полицейские базы данных не более безопасны, чем любые другие коммерческие базы данных, но информация в них намного более секретная.

## Анализ трафика

Анализ трафика — это изучение параметров передаваемой информации. Не содержимого самого сообщения, а его характеристик. Кто с кем общается? Когда? Насколько длинные сообщения? Как быстро посылаются ответы, насколько они длинные? Какого рода связь возникает после получения определенного сообщения? Это все вопросы анализа трафика, и ответы на них могут быть очень информативны.

Например, то, что каждый раз, когда Алиса шлет длинное сообщение Бобу, Боб посылает короткий ответ Алисе и длинное сообщение пяти другим людям, указывает на цепь передачи команд. Алиса просто отсылает распоряжения Бобу, который передает их своим подчиненным. Если Алиса регулярно посылала Бобу короткие сообщения и внезапно посылает ряд длинных, то это говорит об изменении чего-то (чего?).

Часто даже системы передачи информации так же важны, как сама информация. Например, простой факт, что Алиса звонит каждую неделю известному террористу, более важен, чем детали их разговора. Нацисты использовали данные анализа телефонных счетов в оккупированной Франции, чтобы арестовывать друзей арестованных; их на самом деле не интересовало, о чем был разговор. Звонки из Белого дома Монике Левински довольно показательны даже без записи беседы. В часы, предшествующие американской бомбардировке Ирака в 1991 году, доставка пиццы в Пентагон стократно возросла. Любой, обративший на это внимание, точно знал: *что-то* произошло (довольно интересно, что ЦРУ заказывало такое же количество пиццы, как обычно). Некоторые исследования показывают, что даже если вы зашифруете свой веб-трафик, анализа, основанного на размере зашифрованных веб-страниц, более чем достаточно, чтобы понять, что вы просматриваете.

Хотя военные используют анализ трафика уже десятки лет, он до сих пор — новая область исследований в академическом мире. Мы в действительности не знаем, насколько коммуникации, особенно наши интернет-связи, уязвимы для анализа трафика и что можно сделать для снижения риска. Ожидается, что в будущем это будет важной научной областью.

## Широкомасштабное электронное наблюдение

ECHELON — это кодовое обозначение автоматизированной глобальной системы перехвата, управляемой службами безопасности США, Великобритании, Канады, Австралии и Новой Зеландии, возглавляет которую Агентство национальной безопасности. По оценке специалистов, ECHELON перехватывает ежедневно до 3 миллиардов сообщений, включая телефонные звонки, сообщения электронной почты и информацию Интернета, спутниковые передачи и т. п. Система собирает все эти передачи без разбора, а затем сортирует и очищает информацию при помощи программ искусственного интеллекта. В некоторых источниках заявляется, что ECHELON анализирует 90% данных интернет-трафика, хотя это кажется сомнительным[1].

[1] База США в Германии Bad Aibling, выполнявшая с 1947 года радиопрослушивание стран Восточной Европы, а в последние годы работавшая на систему ECHELON, расформировывается в 2002 году вследствие недоверия немцев. Высказывались опасения, что Bad Aibling в числе прочего занимается промышленным шпионажем. — *Примеч. ред.*

Такие попытки широкомасштабного наблюдения устрашают и порождают некоторые уникальные проблемы. Информация, полученная таким путем, полезна, если только придать ей вид, в котором люди могли бы ее понять и действовать в соответствии с результатами ее анализа. Соединенные Штаты перехватили сообщение для японского посла в Вашингтоне, в котором обсуждалась бомбардировка Пирл Харбора, но эта информация приобрела смысл только в ретроспективе и не могла обрести его раньше из-за низкого уровня квалификации служащих, к которым она попала. Но трудно не только анализировать данные, но и принять решение, что нужно записывать.

Потенциально перехват системой ECHELON — нескончаемый поток данных, который никогда не смогли бы обработать сколь угодно много аналитиков. Перехватывающее устройство должно решить в реальном времени, стоит или нет записывать часть информации для дальнейшего анализа. И система не может позволить себе слишком много «дальнейших анализов» — она должна продолжать запись информации. Я уверен, что наиболее ценная часть записанной информации никогда не будет изучаться людьми подробно.

Чтобы построить систему, аналогичную этой, вам следовало бы инвестировать деньги в две технологии: возможности первичной обработки и анализ трафика.

Оборудование перехвата должно получить возможность быстро характеризовать фрагменты информации: кто отправитель и получатель, тему сообщения, как ее классифицировать. (Если вы считаете, что трудно проделать такую работу с информацией Интернета и электронной почты, подумайте, как сложно это сделать при речевом общении.) Во многом эта технология похожа на ту, что заложена в поисковых системах.

Анализ трафика еще более важен. Схемы трафика проясняют многое о любой организации, и их намного проще собирать и анализировать, чем реальные коммерческие данные. Эти схемы также поставляют дополнительную информацию для первичной обработки. Тщательно разработанные базы данных по схемам трафика, несомненно, являются сердцем любой подобной ECHELON системы.

Последний аккорд: в мире, где большинство сообщений не зашифровано, зашифрованные сообщения, возможно, записываются регулярно. Простого знака, что собеседники не хотят, чтобы их подслушали, будет достаточно, чтобы бить тревогу.

## Атаки ради рекламы

Концепция атак ради известности очень проста: «Как мне атаковать систему, чтобы мое имя появилось в газетах?» Такой тип атаки является относительно новым в цифровом мире: несколько лет назад взлом компьютерной системы не считали достойным освещения в прессе, и я не могу найти в истории никакой другой технологии, которую люди пытались бы испортить только для того, чтобы их упомянули в газете. В физическом мире, однако, этот тип атаки древен: человек, который сжег Храм Артемиды в Древней Греции, сделал это только потому, что хотел, чтобы его имя осталось в веках. (Кстати, его звали Герострат.) Из более позднего упомяну искавших того же детей, которые расстреляли школу в Колумбине.

Большинство злоумышленников такого типа — это хакеры: умельцы, много знающие о системах и их безопасности. Часто у хакеров есть доступ к значительным ресурсам — в лице студентов большого университета или служащих крупных компаний. У них обычно нет больших денег, но иногда случается избыток времени. Кроме того, они чаще всего не делают того, что может привести их в тюрьму: основная их цель — огласка, а не заключение под стражу.

Канонический пример такой атаки: два аспиранта университета Беркли в 1995 году взломали схемы кодирования Netscape Navigator. Обнаруженные в системе слабые места аспиранты не использовали для получения незаконной прибыли, а просто позвонили в *Нью-Йорк Таймс*. Реакция Netscape была примерно такая: «Мы провели ряд вычислений и полагаем, что такая атака потребовала бы безумных затрат; мы не думаем, что взлом системы стоил бы чьих-то усилий». Они правы: взлом не стоит усилий... для того, кого интересуют деньги. У аспирантов были все навыки, доступ ко всем свободным компьютерам в университете и никакой общественной жизни[1].

Что важно осознать создателям систем — так это то, что люди, желающие огласки, не укладываются в ту же модель, что и преступники. Преступники будут атаковать систему, только если от этого можно получить выгоду; люди, которым необходима огласка, будут нападать на систему, если высока вероятность, что это станет освещаться прессой. Для них атаки, направленные на крупномасштабные системы и широко распространенные программные продукты, — лучше всего.

Иногда такие атаки мотивируют тем, что требуется привлечь внимание к проблеме. Многие компании игнорируют уязвимость своей системы безопасности, если не привлечь к ней внимание общественности. Если исследователь объявляет атаку, компания-жертва будет суетиться, чтобы исправить ситуацию. Таким образом, атаки повышают безопасность систем.

Атаки ради известности могут дорого обходиться. Клиенты могут после такой атаки отказаться от одной системы в пользу другой, как уже произошло после нескольких атак на банковские системы. Инвесторы могут отказать жертве в фондах. Это уже произошло с индустрией цифровой сотовой связи после подобных атак, показавших недостатки защитных мер по отношению к конфиденциальности и краже. Ситибанк лишился многих значительных вкладов после взлома хакером из Санкт-Петербурга. Взлом системы безопасности цифровых видеодисков

[1] Это утверждение больше напоминает проблему отцов и детей. Внештатный эксперт PC Magazine и автор многих книг по информационным технологиям Джефф Просис провел собственное расследование по следам раскрытия алгоритма защиты информации Netscape. В опубликованной им статье говорится, что Netscape разразилась подобным заявлением, но только в ответ на сообщение аспиранта из Франции Долигеса, который, используя мощности 120 рабочих станций и двух суперкомпьютеров, за 8 дней «в лоб» рассчитал значение 40-битового ключа SSL-сообщения. Эту задачу ранее пытались решить многие с целью убедить правительство США снять экспортные ограничения на средства шифрования (для работы американских компаний за пределами США). Долигес «сделал это», но за месяц до того, как действительно два студента Беркли решили другую задачу — обнаружили, что ключ задается на базе ненадежного случайного числа. Им удалось сузить диапазон возможных ключей, и они определили, что уровень защиты шифрования Netscape Navigator равносилен ключу длиной 47 разрядов. Вот тогда Netscape засуетилась и срочно начала работы по пересмотру алгоритма генерации ключей. Кстати, Голдберг и Вагнер использовали всего один персональный компьютер. А вот в чем соглашается Просис со Шнайером, так это в том, что прочность любой защиты определяется прочностью ее наименее стойкого звена. — *Примеч. ред.*

(DVD) приостановил поступление на рынок продуктов Sony после рождества 1999 года. В 2000 году CD Universe потеряли множество клиентов в результате того, что хакеры украли с веб-сайта компании 300 000 номеров кредитных карт. Иногда дурная слава обходится дороже, чем реальная кража.

Атаки ради известности грозят и другими опасностями. Одна из них — в том, что о таких атаках узнают преступники и будут использовать успешную методику. Другая — в том, что доверие общества к системам разъедается оглаской подобных фактов. В частности, это может быть основной проблемой для электронных коммерческих систем. Банки вынуждены сохранять в тайне успешные атаки преступников на их системы с тем, чтобы не волновать общественность. Но хакеров и образованных людей гораздо труднее урезонить, и они продолжают держать в поле зрения коммерческие системы. Если где-то в системе безопасности есть прорехи, кто-нибудь найдет их и созовет пресс-конференцию. Может быть, не первый, кто обнаружит прорехи, но кто-нибудь огласит все. Компании должны быть готовы к этому.

«Дефейс» — подмена чьих-нибудь веб-страниц — одна из форм атак ради известности. Это обычно становится важной новостью в средствах массовой информации. В 1996 году таковой стал взлом веб-сайта Министерства юстиции. Тот же эффект имел в 1997 году взлом сайта AirTran, а в 1998 — главной страницы сайта газеты *New York Times*.

В те дни настроения были таковы, что на некоторых сайтах вовсе не ожидали нападений. Студия Метро-Голдуин-Майер/Юниверсал была шокирована; когда в 1995 году взломали веб-сайт их фильма «Хакеры». А в 1997 студия Юниверсал Пикчерз сама взломала свой веб-сайт фильма «Парк юрского периода» в рекламных целях. (Она пыталась сделать вид, что это дело рук хакеров, но имитированный сайт выглядел слишком профессионально, а взломанная страница была загружена на сайт заранее, за три дня.)

В наши дни подобные атаки происходят настолько часто, что их только вскользь упоминают в новостях. Возможно, все основные веб-сайты американского правительства были взломаны в 1999 году, как были взломаны веб-сайты множества местных (муниципальных) и иностранных правительств. Я привел в первой главе список из 65 повреждений веб-сайтов в первую неделю марта 2000 года. Системные администраторы уже привыкли к этой проблеме.

## Атаки, приводящие к отказам в обслуживании

Совсем недавно атаки типа «отказ в обслуживании» стали де-юре-атаками, направленными на огласку. Это произошло только вследствие их широкого освещения в прессе/и, будем надеяться, они вскоре перестанут быть новостью. Идея в том, чтобы просто остановить работу чего-нибудь. И, как вам скажет любой, кому приходилось иметь дело с бастующими рабочими — водителями автобусов, диспетчерами воздушного движения, рабочими на ферме и т. п., — такие атаки весьма эффективны.

В физическом мире есть и другие атаки, приводящие к отказам в обслуживании, например блокады и бойкоты. Все эти атаки имеют свои аналоги в киберпро¬странстве. Имея достаточные возможности телефонной связи, можно заблокиро¬

вать все соединения модема локального поставщика услуг Интернета. Аналоговая сотовая телефонная сеть испытывает затруднения с соединением, если перемещающийся пользователь переходит от одной ячейки к другой; можно сидеть на холме с направленной антенной и, медленно поворачивая ее туда-сюда, заблокировать все каналы ближайших ячеек.

Нападения, приводящие к отказам в обслуживании, работают, поскольку компьютерные сети являются сетями связи. Некоторые простые атаки, вроде передачи слова «привет», можно автоматизировать до такой степени, что они становятся атаками, приводящими к отказам в обслуживании. Так в основном устроены атаки, рассчитанные на «затопление» сети, которым подверглись некоторые провайдеры в 1996 году.

Вот другая атака, повлекшая отказ в обслуживании: в середине 1980-х политическая организация Джерри Фолвелла установила бесплатный телефон для разных целей. Один парень запрограммировал свой компьютер на то, чтобы непрерывно набирать этот номер, а затем вешать трубку. Таким образом он осуществил две вещи: во первых, занял линию, и люди, которым действительно нужно было позвонить по этому номеру, не могли дозвониться, а во-вторых, организации Фолвелла приходилось платить деньги за каждое соединение. Прелестный пример атаки, приводящей к отказу в обслуживании.

Подобные действия могут быть прелюдией к преступной атаке. Взломщики подходят к складу в 1:00 ночи и перерезают провод охранной сигнализации, ведущий к полицейскому участку. Сигнализация срабатывает и предупреждает полицию, что провод поврежден. Взломщики отступают на безопасное расстояние и ждут, когда приедет полиция. Полиция приезжает и ничего не обнаруживает. (Если преступники изобретательны, они перережут провод в незаметном месте.) Полиция решает, что проблема в системе, а владельцы склада решают, что разберутся с ней утром. Полиция уезжает. Взломщики возвращаются и уносят все, что хотят.

Вариант этого сценария, который, как подтверждается несколькими случаями, дает гарантированный результат, — это атака телефонного узла, через который проходит сигнал тревоги. Многие типы сигнализации включают в себя устройство, которое посылает в полицейский участок сигнал о нарушении защиты. При атаке телефонного узла этому сигналу не пробиться, и полиция не узнает, что сигнализация сработала.

Вот другой пример: военная база, окруженная забором, снабженным сенсорами движения. Злоумышленники берут кролика и перекидывают его через забор, а затем убегают. Срабатывает сенсор движения. Охрана приходит, ничего не обнаруживает и возвращается на пост. Злоумышленники повторяют всю процедуру, охрана вновь реагирует. После нескольких таких ночей охрана отключает сенсор движения. И злоумышленники на джипе протаранивают забор. Такого рода атаки неоднократно использовались против русских военных баз в Афганистане и, в качестве проверки охраны, на некоторых военных базах США. Они удивительно удачны.

Похожая атака, предположительно, была предпринята против советского посольства в Вашингтоне, округ Колумбия. Американцы подожгли сласти (по существу, кусок сахара) у окна посольства. От треска включилась сигнализация, но сахарный шарик испарился, и не было заметно, из-за чего возникла тревога. Затем

следующий шарик. Чпок. Тревога. Ничего. В конечном итоге сигнализацию изменили так, чтобы резкий звук за окном не включал бы ее. (Я не знаю, произошло ли в результате этой атаки реальное проникновение или это был просто способ подействовать на нервы охране советского посольства.)

Более близкий пример: широко используемый метод кражи автомобилей заключается в том, чтобы включать сигнализацию в 2 часа ночи, 2:10, 2:20, 2:30... пока владелец машины не отключит ее, чтобы успокоить разгневанных соседей. Утром машины нет.

Во время войны атаки, приводящие к отказам в обслуживании, используют все время. Каждая из сторон пытается заблокировать вражеские системы радиолокации и управления стрельбой, нарушить системы коммуникации, взорвать мосты. Одно из свойств подобных атак в том, что для них часто нужен низкий, а не высокий технический уровень: взорвать компьютерный центр намного проще, чем использовать уязвимые места Windows 2000.

Атаки, приводящие к отказам в обслуживании в Интернете, детально обсуждаются в главе 11.

## «Законные» атаки

В 1994 году в Великобритании один человек обнаружил, что его банковский счет пуст. Когда он подал жалобу на то, что последние шесть процедур снятия денег со счета он не производил, его арестовали и обвинили в краже. Британский банк заявил, что система безопасности кредитных карт непогрешима, а подсудимый, несомненно, виновен. Когда адвокат изучил доказательства, он обнаружил следующее:

1. У банка нет службы безопасности и гарантии качества программного обеспечения.
2. В нем никогда не проводились проверки безопасности.
3. Обсуждаемые изъятия денег никто не проверял.

Фактически программисты банка заявили, что поскольку код написан на ассемблере, в нем не может быть проблем (дескать, если бы там были ошибки, то они привели бы к поломке системы). Так или иначе, этого человека осудили. При рассмотрении апелляции банк представил суду гору документов, подготовленных аудиторской фирмой и подтверждающих безопасность системы. Когда защита потребовала неограниченного доступа к системе для того, чтобы самостоятельно оценить степень ее безопасности, банк отказал, но обвинение было снято.

От атак, использующих несовершенство законодательства, защититься труднее всего. Цель таких атак не в том, чтобы использовать недостатки системы. И даже не в том, чтобы обнаружить эти недостатки. Цель состоит в том, чтобы убедить судью и присяжных (которые, возможно, не сильны в технике), что в системе *могут быть* недостатки. В том, чтобы дискредитировать систему, заронить в умы судьи и присяжных сомнения в совершенстве системы и в результате доказать невиновность клиента.

Вот гипотетический пример. В большом деле о наркотиках полиция использует данные сотовой телефонной сети, по которым устанавливает местонахождение

обвиняемого с телефоном в определенном месте в определенное время. Адвокат находит какого-нибудь специалиста-хакера, который дает показания, что такие данные легко могли быть сфабрикованы, что они ненадежны и не могут служить доказательством. У прокурора есть другая группа экспертов, которые утверждают противоположное, и единственный возможный выход состоит в том, что они уравновесят друг друга: суд продолжится без учета данных сотовой связи.

То же самое может произойти, если данные экспертизы используют для осуждения кого-то, взломавшего компьютерную систему, или если данные о подписании документов используют для навязывания контракта. «Я этого никогда не подписывал, — говорит ответчик. — Компьютер попросил меня ввести пароль и нажать эту кнопку. Что я и сделал». Присяжные, возможно, в такой же степени несведущи в технологии, в какой представляет себя ответчик, и, скорее всего, сочувствуют ему.

У всякой монеты есть и оборотная сторона. Полиция может использовать показания экспертов, чтобы убедить присяжных в том, что расшифрованное сообщение является изобличающей уликой, хотя расшифровка не стопроцентно точна, или что установление факта компьютерного вмешательства несомненно, и поэтому подсудимый виновен.

Использование «законных» атак в полной мере предоставляет огромные возможности. Многие преступники в высшей степени искусны — в некоторых случаях они могли бы стать лучшими экспертами по вопросам безопасности — и хорошо обеспеченными. Они могут использовать процесс исследования системы, чтобы узнать все необходимые детали. Им даже не нужно проводить атаку; злоумышленнику достаточно найти доказательства существования слабых мест в системе безопасности. Подобные действия можно считать атаками ради престижа, подкрепленными денежными средствами и с почти гарантированной победой.

# Глава 4. Противники

Так кто же все-таки угрожает цифровому миру? Хакеры? Преступники? Распространители порнографии? Правительства? Противники — те же самые, что и в обычном мире: уголовные преступники, жаждущие обогащения; промышленные шпионы, охотящиеся за секретами, способными обеспечить конкурентоспособность товаров; хакеры, ищущие тайные ходы; разведка, добывающая военные сведения. Они не изменились, просто киберпространство стало новым полем их деятельности.

Мы можем разделить противников на категории несколькими способами, приняв за основу классификации цели, доступ, ресурсы, квалификацию и риск.

Цели противников могут быть различны: причинение ущерба, финансовая выгода, информация и т. д. Это важно. Цели промышленного шпиона отличаются от целей синдиката организованной преступности, и контрмеры, которые способны остановить первого, могут даже не побеспокоить второй. Понимание целей вероятных противников — это первый шаг к выяснению, какие контрмеры могут быть эффективными.

Противники имеют различный уровень доступа: возможности члена какой-либо организации, например, намного больше, чем любого одиночки. Противники также сильно различаются по своим финансовым возможностям: некоторые хорошо финансируются, другие висят на волоске. Одни имеют достаточную техническую квалификацию, у других ее нет.

Различные противники по-разному относятся к риску. Террористы часто бывают счастливы умереть за свои убеждения. Преступники смиряются с риском оказаться в тюрьме, но, вероятно, не захотят иметь неприятности сверх тех, которыми может обернуться грабеж банка. Ищущие славы вовсе не хотят попасть в тюрьму.

Состоятельный противник наиболее гибок в решениях, так как он может использовать свои средства для различных вещей. Он может получить доступ, подкупив посвященных лиц, и повысить свой технический уровень, купив технологию или наняв экспертов (возможно, посвятив их в свои намерения, возможно, нанимая их под ложными предлогами). Он может также использовать деньги для снижения риска, совершая более подготовленные и поэтому более дорогостоящие атаки.

Рациональный противник (не все из нападающих в здравом уме, однако большинство достаточно обдуманно подходят к делу) выбирает нападение, которое с лихвой окупит понесенные расходы с учетом всех издержек: квалификации, доступа, трудовых ресурсов, времени и риска. Некоторые нападения требуют хорошей квалификации, но не требуют никакого специального доступа: взлом алгоритма кодирования, например. Каждый противник старается использовать набор приемлемых для него видов атак, отбросив те, которые ему не подходят. Конечно же, он выберет такое нападение, которое уменьшает затраты и увеличивает выгоды.

# Хакеры

Слово «хакер» имеет широкий спектр значений — от системного администратора, достаточно хорошо представляющего, как в действительности работают компьютеры, до подростка-преступника, который кудахчет от восторга, когда громит вашу сеть. Слово было подхвачено средствами массовой информации, и его первичное значение изменилось. Оно скорей используется как комплимент, нежели как оскорбление. В последнее время люди используют слово «крекер» (взломщик программной защиты) для плохих парней и «хакер» — для хороших. Я определяю хакера как индивидуума, который экспериментирует с недостатками системы ради интеллектуального любопытства или собственного удовольствия; это слово описывает человека со специфическим набором навыков и неспецифической моралью. Есть хорошие хакеры и плохие хакеры, аналогично хорошим водопроводчикам и плохим водопроводчикам. (Есть также «хорошие плохие» хакеры и «плохие хорошие» хакеры... но не берите это в голову.)

Хакеры стары как любопытство, хотя сам по себе этот термин современен. Галилео Галилей был хакером. Мадам Кюри тоже. Аристотель не был. (Аристотель приводил некие теоретические доказательства, что у женщины меньшее количество зубов, чем у мужчины. Хакер просто посчитал бы зубы своей жены. *Хороший* хакер посчитал бы зубы своей жены без ее ведома, в то время когда она спала бы. Хороший *плохой* хакер мог бы удалить некоторые из них, только бы доказать свое теоретическое предположение.)

Когда я учился в колледже, я знал людей, подобных хакерам, назовем их страстными коллекционерами ключей. Они хотели иметь доступ всюду, и их цель была в том, чтобы владеть ключом от каждого замка в университетском городке. Они изучали новые системы запоров, карты коммуникаций, запоминали их расположение и обменивались друг с другом копиями ключей. Запертая дверь была вызовом, личным оскорблением. Эти люди не собирались причинить кому-либо ущерб — воровство не было их целью — хотя, конечно, они могли бы использовать для этого свои знания. Их страстью было получать доступ всюду, куда бы им захотелось попасть.

Помните телефонных взломщиков, которые могли болтать по таксофонам и делать бесплатные телефонные звонки? Несомненно, они терроризировали телефонную службу. Но им это было нужно не для того, чтобы делать восьмичасовые звонки в Манилу или Мак-Мурдо. Они хотели знать систему лучше, чем проектировщики, и уметь изменять ее по своему желанию. Понимание того, как работает телефонная система, служило им наградой. Другой пример раннего хакерства — фанаты-радиолюбители.

Ричард Фейнман был хакером, почитайте любую из его книг[1].

Компьютерные хакеры унаследовали эти черты. Скорее даже они принадлежат к тому же племени, только действуют в новых условиях. Компьютеры и сети, в частности, — это новый ландшафт, который можно исследовать. Сети представляют

[1] Американский физик-теоретик, лауреат Нобелевской премии за работы по квантовой электродинамике. Работы Фейнмана посвящены квантовой электродинамике, квантовой механике, статистической физике. Автор способа объяснения возможных превращений частиц, количественной теории слабых взаимодействий, теории квантованных вихрей в сверхтекучем гелии, модели нуклона и др. Доктор философии. Умер в 1988 году. Автор сатирико-юмористической книги-бестселлера «Вы, конечно, шутите, мистер Фейнман!». Похоже, что Шнайер апеллирует именно к ней. — *Примеч. ред.*

сложнейшее переплетение многочисленных связей, где новая хакерская технология становится ключом, который может открывать компьютер за компьютером. А за этим — знание, понимание. Как получить доступ? Что и как работает? Почему это вообще работает? Ответы где-то рядом, они ждут, чтобы их обнаружили.

Сегодняшние компьютерные хакеры, как правило, молоды (около двадцати), мужского пола и социально — на задворках общества. Они имеют свою собственную культуру: хакерские имена-прозвища, язык, правила. И, что характерно для любой субкультуры, только маленький процент ее представителей действительно что-то собой представляет. Настоящие хакеры понимают технологию на базовом уровне и ими движет желание расширить свое понимание. Остальные же — бездарные позеры, полностью неспособные ни к чему, или преступники. Иногда их называют ламерами.

Хакеры могут иметь достаточную квалификацию, часто более высокую, чем сами проектировщики системы. Я прослушал большое количество лекций по безопасности, и большинство ораторов, в чьих лекциях есть здравый смысл, — хакеры. Это их страсть. Хакеры смотрят на систему с внешней стороны, с позиции нападающего, а не с внутренней — с позиции проектировщика. Они смотрят на систему, как на организм, как на единое целое. И часто понимают атаки лучше, чем люди, которые разрабатывают системы. Таковы настоящие хакеры.

У хакеров обычно много времени, но мало финансовых средств. Некоторые из них питают отвращение к риску и в высшей степени осторожно балансируют на грани закона, у других нет страха перед наказанием, и они занимаются незаконной деятельностью без мысли о связанном с ней риске.

Имеются хакерские телеконференции, хакерские веб-сайты и хакерские соглашения. Хакеры часто продают способы и автоматизированные средства атак друг другу. Есть различные группы хакеров (или шайки, если вам так больше нравится), но нет никакой иерархии. Вы не сможете нацелить сообщество хакеров на что-то определенное: они пойдут так далеко, как смогут. Часто они будут взламывать что-то лишь потому, что это широко известно, интересно, или потому, что цель «заслуживает» этого.

К сожалению, большинство хакеров совершают незаконные действия. Я не говорю о тех немногих, кто работает в исследовательской среде, кто лабораторно оценивает параметры защиты систем и кто публикует анализ исследований программных продуктов и систем. Я говорю о тех хакерах, которые врываются в пользовательскую сеть, стирают веб-страницы, вызывают аварийные отказы компьютеров, распространяют вирусы и пишут автоматические программы, которые позволяют другим делать то же самое. Эти люди — преступники, и общество должно относиться к ним как к таковым.

Я не куплю систему, защиту которой хакер взломал только затем, чтобы к ней присмотреться, не причинив никаких повреждений. Некоторые системы довольно хрупкие, и даже тот, кто просто присматривается, может нечаянно что-нибудь испортить. А после того как посторонний побывал внутри системы, вы более не можете быть уверены в ее целостности. Вы не знаете, прикасался или нет непрошеный гость к чему бы то ни было.

Вообразите, что вы приходите домой и на двери своего холодильника находите записку: «Привет. Я заметил, что у вас паршивая блокировка входной двери, так что я вломился. Я ничего не касался. Вы действительно должны улучшить свою систему защиты». Как бы вы чувствовали себя?

Проблемы начинаются с хакеров, которые создают инструменты взлома. Это программы — иногда их называют эксплоитами (exploits), — которые автоматизируют процесс вторжения в системы. Пример — Trin00 — инструмент, создающий отказы в обслуживании. Тысячи серверов были выведены из строя после нападения с помощью этой программы, что потребовало от компаний-владельцев миллионов долларов, уйму времени и усилий для восстановления. Trin00 — одна из причин исследований уязвимости Интернета для такого типа атак и повод для написания научного труда о защите от них. А эти занятия гораздо сложнее, нежели составление программы, автоматизирующей атаку.

Программа Trin00 не предназначена ни для каких других целей, кроме нападения на системы. Владельцы оружия могут рассуждать о самообороне, но интернет-серверы не врываются в наш дом по ночам. А эта программа намного страшнее, потому что однажды написанная, она стала доступной, и любой, кто захочет быть хакером, сможет загрузить ее и напасть на компьютеры в Интернете. При этом ему даже не обязательно знать, как все работает. Атаки с помощью Trin00 были популярны в начале 2000 года, потому что эта программа была доступна. Без этой доступности — даже при наличии описаний уязвимых мест систем — ни один ламер не был бы способен воспользоваться наличием последних.

Конечно, те, кто использует Trin00, чтобы напасть на системы, — преступники. Я также верю, что тот, кто ее написал, тоже преступник. Существует тонкая грань между написанием кода для демонстрации исследовательской работы и публикацией инструментальных средств нападения; между невинным хакерством и хакерством как преступной деятельностью. Я возвращусь к этому в главе 22.

Большинство организаций с оправданной осторожностью относится к найму хакеров. Есть и исключения — Агентство национальной безопасности, предлагающее стипендию хакерам, желающим работать в Форт Мид; израильская разведка, нанимающая еврейских хакеров из Соединенных Штатов; Вашингтон, предлагающий создать ассоциацию защиты. Некоторые хакеры работают в частных компаниях и профессионально занимаются безопасностью. Недавно ряд консалтинговых компаний попытался оправдать хакеров и представить их в более выгодном свете. Иногда такой подход справедлив, но многим людям бывает тяжело понять отличие этики хакеров от этики преступников.

## Преступники-одиночки

В апреле 1993 года маленькая группа преступников привезла автоматизированную модель банкомата Fujitsu 7020 на аллею Бэклэнд Хилл в Хартфорде (штат Коннектикут). Машина была специально запрограммирована, чтобы принимать кредитные карты (ATM-карты) от клиентов, записывать номера их счетов и личные идентификационные номера (PIN), а затем сообщать неудачливым клиентам, что выдать деньги невозможно. Несколькими днями позже банда скопировала похищенные номера счетов и личные идентификационные номера на поддельные карты ATM и стала получать по ним наличные деньги в центре Манхэттена. В конечном счете преступники были пойманы, когда банк сопоставил использование поддельных карт с ежедневными записями видеонаблюдения.

Это было умное и практичное нападение, технически более высокоорганизованное, чем большинство банковских преступлений. Один технически продвинутый преступник из Нью-Джерси прикреплял поддельный уличный депозитный ящик к стене банка и убирал его ранним утром. Но более интересное произошло в другом месте. Несколько лет назад автоматизированный кассовый аппарат был украден в Южной Африке... из полицейского управления средь бела дня.

Преступники-одиночки совершают большую часть связанных с компьютером преступлений. Иногда, являясь людьми осведомленными, они замечают недостаток в системе и принимают решение использовать его, иногда же они нападают «снаружи». Обычно у них немного денег, отсутствует доступ и недостаточно хорошо организована экспертиза, и они часто попадаются из-за глупых ошибок. Кто-то может быть достаточно сообразительным, чтобы устанавливать поддельные автоматизированные кассовые аппараты, собирать номера счетов и личные идентификационные номера, но если уж он хвастается своим умом в баре и дает арестовать себя прежде, чем вычистит все регистрационные записи... ну, в общем, сложно испытывать к нему симпатию. Посмотрите на два вызывающих нападения на Интернет в начале 2000 года. Кто-то организовал доступ к десяти тысячам номеров кредитных карт с именами и адресами. Лучшее, что он мог выдумать, — это вымогательство. Кто-то другой установил контроль над большим числом компьютеров, готовых подчиниться его указаниям. Но не смог придумать ничего лучше, чем донимать администраторов этих веб-сайтов.

Преступники-одиночки будут нацеливаться на торговые системы, потому что там — деньги. Их методы могут страдать недостатком изящества, но они будут красть деньги, и еще большие деньги понадобятся для их поимки и доказательства вины.

## Злонамеренные посвященные лица

Злонамеренный член организации — опасный внутренний враг. Он всегда внутри системы, поэтому, когда он хочет атаковать, его не беспокоят заграждения, приготовленные для незваных гостей. Возможно, он имеет самый высокий уровень доступа и рассматривается системой как заслуживающий доверия, в то время как он ее атакует. Помните русского шпиона Олдриджа Эймса? Он занимал удобнейшую позицию в Центральном разведывательном управлении, чтобы продавать КГБ имена американских разведчиков, находящихся на территории Восточной Европы: его имени доверяли. Подумайте теперь о программисте, который имеет возможность написать код для базы данных платежных ведомостей, дабы обеспечить себе повышение зарплаты каждые шесть месяцев. Или о службе охраны банка, которая сообщает своим дружкам-грабителям предположительное время закрытия кассы. Злонамеренных сотрудников практически невозможно остановить, поскольку они именно те люди, которые пользуются доверием.

Вот канонический пример атаки посвященного лица. В 1978 году Стэнли Марк Рифкин был консультантом в центральном отделении банка. Он использовал свои знания и доступ к системе пересылки денег, чтобы переместить несколько миллионов долларов на счет в швейцарском банке, а затем обратить эти деньги в алмазы. Он также запрограммировал компьютерную систему на автоматическое стирание записанной на пленку резервной копии данных, поскольку в ней содержалось сви¬

детельство его преступления (он ушел бы с этими деньгами, если бы не похвастался своему адвокату, который и рассказал обо всем).

Посвященные лица не всегда нападают на систему, иногда они просто используют ее в преступных целях. В 1991 году служащий Чарльз Шваб из Сан-Франциско использовал электронную почту компании для покупки и продажи кокаина. Некто, осужденный за насилие над детьми, работал в больнице в одном из районов Бостона. Украв пароль сослуживца, он изучал больничные карты пациентов, которым звонил с непристойными предложениями.

Посвященные лица — это необязательно служащие. Они могут быть консультантами и подрядчиками. Во время паники в связи с «проблемой 2000 года» многие компании нанимали программистов из Китая и Индии, чтобы обновить старое программное обеспечение. Оставив ксенофобию в стороне, замечу, что любой из тех программистов мог напасть на системы, будучи посвященным лицом.

Большинство компьютерных мер защиты — аппаратно-программные средства сетевой защиты (брандмауэры), системы обнаружения вторжения и т. д. — имеют дело с внешними нападающими, но в значительной степени бессильны против посвященных лиц. Атака на систему со стороны посвященных лиц менее вероятна, чем со стороны посторонних, но системы гораздо более уязвимы перед ними.

Посвященное лицо в курсе, как системы работают и где их слабые места. Такой человек знает структуру организации и то, как будет вестись любое расследование его действий. Он всегда пользуется доверием системы, которую собирается атаковать. Посвященное лицо может использовать собственные ресурсы системы против нее самой. В убийственных с точки зрения безопасности случаях посвященное лицо имеет высокую квалификацию, и еще хуже, если оно участвовало в проектировании системы.

Мотивация нападений посвященных лиц может быть различна: месть, финансовая выгода, изменение существующих порядков или даже реклама. Вообще говоря, она совпадает с мотивацией хакера, преступника-одиночки или агента национальной разведки. Степень риска, на который готовы идти злонамеренные посвященные лица, зависит от того, движимы они «высокой целью» или простой жадностью.

Конечно, атаки посвященных лиц не новы, и проблема эта возникла не в кибер¬пространстве. Если бы не было системы электронной почты, служащий Чарльз Шваб мог бы использовать телефонную систему, факсимильные аппараты или, может быть, даже бумажную почту.

## Промышленный шпионаж

Бизнес — это война. Вернее, это некое подобие войны, в которой есть судьи. Судьи устанавливают правила — что является законным, а что нет — и стараются проводить их в жизнь. Иногда, если в бизнесе задействовано достаточное количество денег и влияния, контролирующие его люди могут подать ходатайство судье и добиться изменения правил. Но обычно они делают ходы только в пределах установленных правил.

Черта, где исследовательские методы перестают быть законными, пролегает там, где заканчивается сбор сведений о конкуренте и начинается промышленный шпионаж. Черта эта устанавливается в соответствии с местными законами, но между

последними всегда есть много общего. Вторжение в офис конкурента и кража файлов всегда незаконны (даже для Ричарда Никсона); просмотр их в базе данных новостей всегда законен. Подкуп старших инженеров незаконен; оплата их услуг законна. Платить за то, что они добудут копию исходного кода конкурентов, незаконно. Притвориться, что вы хотите заплатить старшим инженерам конкурентов всего лишь за консультацию... это законно, достаточно тонко и действительно умно.

Промышленный шпионаж имеет четкую мотивацию: получить преимущество в конкурентной борьбе, завладев торговыми секретами конкурентов. Общеизвестный пример: Borland обвинял Symantec в передаче торговых секретов через внедренного в компанию руководителя. В другом случае Cadence Design Systems[1] предъявила иск своему конкуренту — компании Avant!, — среди прочего обвинив ее в краже исходного кода. В 1999 году Alibris, продающая книги через Интернет, обвинила Amazon.com в просматривании электронной почты корпорации. Компании из Китая, Франции, России, Израиля, Соединенных Штатов, как, впрочем, и отовсюду, воруют секреты технологий у иностранных конкурентов.

Промышленный шпионаж может хорошо финансироваться: аморальная, но разумная компания выделит на промышленный шпионаж достаточно средств, которые с лихвой окупятся. Даже если завладение технологией конкурента обойдется вам в полмиллиона долларов, это может быть только десятой частью расходов на самостоятельное создание такой технологии. (Когда-нибудь задайтесь вопросом, почему возвращаемый на Землю космический корабль русских так похож на все американские корабли серии «Шаттл»?) Противник такого типа не готов рисковать по-крупному, потому что репутации компании (неосязаемый, но ценный элемент) будет нанесен значительный урон, если конкуренты уличат ее в шпионаже, — он действует украдкой.

## Пресса

Пресса напоминает промышленного шпиона, но действующего с иными побуждениями. Прессу не интересует победа в конкурентной борьбе, ее интересует «заслуживающая освещения в печати» история. Это могут быть вашингтонская City Pages, публикующая видеозаписи судьи Борка (что привело к появлению закона о защите видеоданных от 1988 года), британские бульварные газеты, выносящие на всеобщее обозрение частные телефонные разговоры принца Чарльза с Камиллой Паркер Боулз, или газеты, делающие разоблачения какой-то компании или какой-либо правительственной акции.

Поднять тираж газеты помогают публикации изображения кандидата в президенты, например Гарри Харта, с чужой женой на коленях. Даже не слишком компрометирующие фотографии принцессы Дианы стоили более полумиллиона долларов. Некоторые репортеры утверждают, что они не стали бы думать дважды, предавать ли гласности секреты национальной безопасности, так как, по их мнению, право публики знать правду важнее.

[1] Компания-разработчик систем автоматизированного проектирования микроэлектронных компонентов. — *Примеч. ред.*

Во многих странах свобода прессы воспринимается как преступление. В таких странах пресса обычно плохо финансируется и вообще больше напоминает жертву, нежели агрессора. Журналисты, обладающие достаточной смелостью, чтобы выступать против правительства, попадают в тюрьмы, их пытают и даже убивают. Это — не тот случай, когда можно говорить о прессе как о нападающей стороне.

В промышленно развитых странах с разумными свободами пресса может выделить достаточные средства для нападения на отдельную систему или цель. Она может хорошо финансировать, может нанимать экспертов и получать доступ к информации. И если журналисты полагают, что их дело правое, они могут рисковать. (Конечно, журналисты, которые устроили Уотергейтский скандал, попадают в эту категорию.) Журналисты в Соединенных Штатах и других странах оказывались в тюрьме, защищая то, что им представлялось правильным. Некоторые даже умерли ради этого.

## Организованная преступность

Организованная преступность — это гораздо большее, нежели итальянские мафиозные «семьи» из фильмов Фрэнсиса Форда Копполы. Это — глобальный бизнес. Русские преступные синдикаты действуют как в России, так и в Соединенных Штатах. Азиатские преступные синдикаты действуют как дома, так и за границей. Колумбийские наркокартели также интернациональны. Нигерийские и другие западноафриканские синдикаты захватили 70% чикагского героинового рынка. Польские гангстеры занимаются угоном дорогих машин в Соединенных Штатах и на кораблях переправляют их в Польшу. Конечно, случаются войны между соперничающими группировками, но хорошо развито и международное сотрудничество.

Объекты бизнеса организованной преступности не изменились за многие столетия: наркотики, проституция, ростовщичество, вымогательство, мошенничество, азартные игры. Использование современной технологии идет двумя путями. Во-первых, это — принципиально новая сфера криминальной деятельности: преступники применяют средства взлома, чтобы ворваться в компьютеры банка и украсть деньги; перехватывают идентификационные коды сотовых телефонов и перепродают их; они занимаются компьютерным мошенничеством. А во-вторых, это — присвоение личности, развивающаяся область; здесь лидируют китайские банды. Конечно, электронное воровство более выгодно: один крупный чикагский банк в 1996 году потерял 60 тысяч долларов из-за грабителей и 60 миллионов долларов из-за мошенничества с чеками.

Воровские шайки используют компьютеры и в своем основном бизнесе. Легко организовать незаконные азартные игры: сотовый телефон позволяет букмекерам работать повсюду, а быстродействующие компьютеры могут стереть все следы в считанные секунды. И отмывание денег становится все более и более тесно связано с компьютерами и электронными платежами: перемещение денег с одного счета через другой на третий, изменение реквизитов, маскировка происхождения денег — перемещение их через страны почти не оставляет следов.

В отношении к риску организованная преступность — это то, что получается при объединении преступников-одиночек в организацию, обладающую большими деньгами. Эти парни знают, что для того, чтобы сделать деньги, нужно их немного потратить, и вкладывают капитал в сулящее прибыль нападение на финансовую систему. У них минимальная квалификация, но они могут купить ее. У них минимальный доступ, но они могут купить и его. Они готовы пойти на больший риск, нежели преступники-одиночки; иерархия преступного синдиката часто вынуждает тех, кто стоит ниже, брать на себя самый большой риск, и защита, предоставляемая синдикатом, делает этот риск более терпимым.

# Полиция

В интересующем нас аспекте полицию можно рассматривать как разновидность национальных разведывательных организаций, за исключением того, что она хуже финансируется, хуже оснащена технически и сосредоточена на борьбе с преступлениями. Тем не менее надо понимать, что от того, насколько благополучна страна, проводятся ли в ней демократические выборы, «борьба с преступностью» может включать в себя целый ряд вещей, обычно не связанных с установлением правопорядка. Возможно, полиция подобна прессе, но имеет лучшее финансирование и читателей, которых интересуют только истинные истории преступления. Или можно думать о полиции как о промышленном конкуренте организованной преступности.

В любом случае полиция обладает достаточными финансированием и квалификацией. Она, в общем, не склонна к риску — никакой полицейский не хочет умереть за свои убеждения, — но так как законы на ее стороне, то вещи, которые являются рискованными для некоторых других групп, могут быть менее опасны для полиции. (Наличие ордера, например, превращает подслушивание из опасного нападения в допустимый инструмент сбора улик.) Первичная цель полиции — сбор информации, которая может быть использована в суде.

Но полиция не должна нарушать закон. Фундаментальное предположение заключается в том, что мы доверяем государству защиту нашей частной жизни и надеемся только на мудрое использование власти. В то же время истина в том, что по большей части злоупотребления регулярны и бывают значительными. Поток незаконных прослушиваний ФБР во Флориде и их утаивание получили некоторое освещение в прессе в 1992 году; было еще 150 или около того незаконных прослушиваний Лос-Анджелесским полицейским управлением. (Конечно, не обошлось без наркотиков: один человек сказал, что война против наркотиков, кажется, является основным паролем к американской Конституции.) Джон Эдгар Гувер регулярно использовал незаконное прослушивание для сбора сведений о своих врагах. А 25 лет назад действующий президент использовал незаконные подслушивания в попытке остаться у власти.

Дела, кажется, изменяются к лучшему со времен Гувера и Никсона, и у меня есть много причин надеяться, что возврата к старому не будет. Но риск остается. Технология развивается медленно, а намерения меняются быстро. Даже если сегодня у нас есть уверенность, что полиция будет придерживаться законодательства, вести подслушивание, только когда необходимо, получать все необходимые

ордера и вообще вести себя, как положено государственной службе, — мы ничего не знаем о завтрашнем дне. Кризис, подобный тому, который привел к преследованию подозреваемых коммунистов в эпоху Маккарти, может снова наступить. Данные переписи в соответствии с законом не предназначены для использования в любых других целях. Даже в том случае, когда они использовались американцами для обнаружения японцев, проживающих в Америке, и помещения их в концентрационные лагеря во время Второй мировой войны. Организация с устрашающим названием «Комиссия суверенитета Миссисипи» шпионила за тысячами активистов движения за гражданские права в 1960-е годы.

ФБР использовало незаконное прослушивание, чтобы шпионить за Мартином Лютером Кингом-младшим. Национальная инфраструктура «открытого ключа» может предшествовать национальной регистрации шифрования. Как только появится новая технология, всегда будет искушение использовать ее. И едва ли гражданская активность создаст механизм удержания полиции в предписанных ее статусом рамках.

## Террористы

Эта категория охватывает широкий диапазон идеологических групп и индивидуумов — как внутренних, так и международных. Здесь нет места для рассуждений на тему морали: террорист — это истребитель свободы других людей. Террористические группы обычно мотивируют свои действия геополитикой или (что еще хуже) этнорелигией — «Хезболлах», «Красные бригады», «Светлый путь», «Тигры Тамила и Ламы», IRA, ETA, FLNC, PKK, UCK, — но они могут быть движимы и моральными или этическими убеждениями, вроде таких, как Earth First и группы радикалов, ратующих за запрещение абортов.

Эти группы вообще больше сосредоточены на причинении вреда, чем на сборе информации, так что их действия по большей части приводят к дестабилизации и полному разрушению. В то время как их долгосрочные цели — обычно нечто невразумительное, вроде восстановления материка Гондвана или возвращения всех коров в дикое состояние, их ближайшие цели — это месть, хаос и кровавая реклама. Хотя больше всего им нравятся бомбы, они не брезгуют и похищениями людей. Когда с неба падает самолет или оказывается разнесенной в пыль клиника по прерыванию беременности, возникает большой международный ажиотаж. Но в конечном счете эти парни поймут, что гораздо большего результата можно достичь, если научиться заставлять диспетчеров аэропорта О'Хара направлять самолеты друг на друга. Или что если они смогут взломать систему бронирования авиабилетов, чтобы выяснить, каким рейсом вылетит на юг Франции делегация Конгресса этим летом, то их террор будет намного эффективней.

Настоящих террористов на самом деле очень немного. Их нападения весьма напоминают военные действия, и они, вероятно, должны попадать в категорию «информационный воин». А так как террористы вообще считают себя лично вовлеченными в состояние войны, они готовы идти на самый большой риск.

Если нет богатого идеалиста, финансирующего их действия, большинство террористов работают на мизерном бюджете. Большая часть из них обладают низкой

квалификацией: «Вам туда. Несите эту сумку. Идите по середине этого оживленного рынка. Нажмите на эту кнопку. Увидимся в чудесной загробной жизни». Имеются исключения (некоторые организации изначально хорошо продуманы, хорошо обучены и имеют хорошую поддержку — например, предполагают, что продажа в Ирландии поддельных телевизионных дешифраторов помогла финансировать IRA), но большинство групп не имеют хорошей организации или доступа. И им присуща тенденция к глупым ошибкам.

## Национальные разведывательные организации

Это — большие мальчики: ЦРУ, Агентство национальной безопасности и Разведывательное управление Министерства обороны США — в Соединенных Штатах (есть и множество других); КГБ (ныне ФАПСИ для контрразведки и ФСБ для иностранной разведки) и ГРУ (военная разведка) — в России; MI5 (контрразведка), MI6 (аналогично ЦРУ) и GCHQ (аналогично Агентству национальной безопасности) — в Великобритании; DGSE — во Франции; BND — в Германии; Министерство национальной безопасности — в Китае (также называемое «Техническим отделом»); Моссад — в Израиле; CSE — в Канаде. Для большинства других противников все это игра: взломать веб-сайт, получить некоторые коммерческие секреты, украсть кое-какие деньги, устроить небольшой погром — все равно что. Для этих парней, однако, это — работа.

Главная национальная разведывательная организация — это наиболее грозный из окружающих противников. Она чрезвычайно хорошо финансируется, так как обычно считается военным подразделением. (Хотя точная сумма является тайной, объединенные бюджеты ЦРУ, Спецслужбы безопасности, Агентства национальной безопасности, Национальной разведывательной службы и других федеральных разведывательных агентств по оценке печати со ссылкой на «источники в Конгрессе» в 1997 году составляли 33,5 миллиарда долларов.) Это преданный делу и искусный противник с финансированием, достаточным, чтобы оплатить целый комплекс исследований, оборудование, экспертизы и обзавестись опытной и квалифицированной рабочей силой.

С другой стороны, главная национальная разведывательная организация обычно совершенно не склонна к риску. Национальные разведывательные организации не любят видеть свои названия на титульном листе «Нью-Йорк Таймс» и вообще не вовлекаются в опасную деятельность (исключения, конечно, существуют: это то, о чем вы все же читаете на титульном листе «Нью-Йорк Таймс»).

Выставление операций на всеобщее обозрение создает несколько проблем. Первая состоит в огласке. Национальная разведка занимается сбором информации, которую страна знать не должна. Это подслушивание позиций сторон на переговорах, сбор сведений о новой системе вооружения, достижение превосходства в осведомленности над противником. Если противник узнает, что известно разведывательной организации, часть выгоды от этого знания будет потеряна.

Второе и, вероятно, более важное: раскрытые операции разоблачают методы, возможности и источники. Много лет АНБ отслеживало по советским автомобиль-

ным телефонам, как Политбюро разъезжало по Москве. Кто-то пропустил информацию о здоровье Хрущева в газеты, и внезапно автомобильные телефоны были зашифрованы. В газетах не писалось что-либо об автомобильных телефонах, но КГБ не был настолько глупым. Утечка информации здесь была не в том, что мы узнали о здоровье Хрущева, но в том, что мы прослушивали их переговоры. То же самое случилось после того, как террористы взорвали берлинскую дискотеку в 1986 году. Рейган объявил, что мы располагаем доказательством причастности к этому Ливии — тем самым он разгласил, что у нас была возможность прослушивать переговоры их посольства с Триполи. В течение Второй мировой войны союзники не могли использовать многие сведения, полученные из расшифровки немецкой системы «Энигма» из опасения, что немцы изменят коды.

Разведывательные цели включают множество вещей, о которых вы знаете: это военная информация, проектировка оружия, дипломатическая информация — и многое другое, о чем вы даже не догадываетесь. Телефонная система — это золотая жила разведки; то же самое относится и к Интернету. Несколько национальных разведывательных организаций активно занимаются промышленным шпионажем (по подсчетам ФБР, до 20 из них шпионят за американскими компаниями) и передают информацию конкурирующим компаниям в своих странах. Китай является самым большим правонарушителем в мире, Франция и Япония тоже хороши, но есть и другие.

Соединенные Штаты не исключение. В 1999 году Европейское сообщество (EU) обнародовало несколько примеров:

- В 1994 году правительство Бразилии предоставило контракт на 1,4 миллиарда долларов корпорации Raytheon в обход предложений двух французских компаний. По общему мнению, Raytheon изменил цену своего предложения, когда узнал детали французских предложений.
- В 1994 году корпорация Douglas McDonell выиграла контракт на строительство самолетов для Саудовской Аравии; возможно, этот выигрыш был основан на внутренней информации, пришедшей от американской разведки.

Прежний директор ЦРУ Р. Джеймс Вусли признал факт использования информации, полученной системой ECHELON об иностранных компаниях, прибегающих к взяткам для заключения международных контрактов: чтобы «выровнять шансы на игровом поле», информация передавалась в американские компании, а иностранные правительства подвергались давлению с целью пресечь взятки. Хотя это и не доказано. Конечно, любая компания, которая теряет предложение, готова найти причины своей неудачи в чем угодно, и ни одна из жертв не скажет чего-либо публично. Однако сама возможность получения информации таким образом вызывает беспокойство.

И этот вид воровства в киберпространстве становится все распространеннее. ECHELON — не единственная программа, которая использует Интернет как поле деятельности. Сингапур и Китай прослушивают поток информации, проходящий по сети Интернет через эти страны. (Китай использует свою национальную сетевую защиту, которая называется Великой стеной.) Интернет-провайдеры в России помогают преемникам КГБ читать частные электронные сообщения и прочую информацию в Интернете, что является частью внутренней программы шпионажа, называемой СОРМ-2.

Национальным разведывательным организациям не чуждо использование хакерского инструментария или даже самих хакеров для выполнения своей работы. У израильского и японского правительств есть программы для привлечения хакеров своих стран — прикармливая их, привлечь к работе по сбору данных. Другие правительства провоцируют хакеров, насмехаясь над ними, с тем чтобы заставить их работать бесплатно. «Если вы настолько хороши, у вас будет пароль к этому правительственному компьютеру» — такие слова сильно действуют на чувство собственного достоинства талантливого подростка. Книга «Яйцо кукушки» Клиффорда Столла повествует об эксплуатации трех хакеров, которые работали на КГБ за деньги и кокаин.

Методы агентств национальной безопасности изменчивы и, с учетом возможностей целой нации, могут быть очень эффективны. Компании, обеспечивавшие безопасность британских коммуникаций, долгое время страдали от слухов о наличии неких дефектов в их программах кодирования — все это по просьбе британской разведки. В 1997 году директор ЦРУ Джордж Тенет упомянул (мимоходом, без деталей) об использовании хакерских инструментов и методов для пресечения международных переводов денег и других финансовых операций арабских бизнесменов, поддерживающих террористов. Возможности бесконечны.

## Информационные войны

Да, это слухи. Но это также и реальность. Инфовоин — это военизированный противник, который старается подорвать способность своей мишени вести войну, атакуя информационную или сетевую инфраструктуру. Атаки этого рода варьируются от неуловимого изменения систем так, чтобы они не работали (или не работали корректно), до полного их разрушения. Нападения могут быть скрытыми, в этом случае они имеют сходство с нападениями террористов (хотя хорошего инфовоина огласка заботит в меньшей степени, чем результаты). Нападения, совершаемые через Интернет, могут иметь иностранное происхождение, и их обнаружение и наказание за них становятся намного более сложным делом.

Этот противник имеет те же ресурсы, что и национальная разведывательная организация, но с различиями в двух важных аспектах. Во-первых, он сосредоточен почти исключительно на краткосрочной цели — пресечении способности своей мишени вести войну. И во-вторых, он готов пойти на риск, который неприемлем для долгосрочных интересов разведки. Его задачи — военное преимущество и, что еще важнее, хаос. Мишенями, которые могут интересовать инфовоина, являются армия и средства управления, телесвязь, тыл и снабжение, а также инфраструктура (читайте «коммерческие информационные системы») и транспортные маршруты (читайте «коммерческая авиация»). Эти виды мишеней называются *критической инфраструктурой*.

В 1999 году НАТО нанесло удар по электростанциям Белграда; это имело далеко идущие последствия для его компьютерных ресурсов. В качестве возмездия сербские хакеры атаковали сотни американских и натовских сайтов. Китайские хакеры вывели из строя компьютеры в Министерстве внутренних дел, Министерстве энергетики и в американском посольстве в Пекине в качестве мести за случайную

бомбежку их посольства в Белграде. Китай и Тайвань участвовали в небольшой кибервойне на протяжении почти всего 1999 года, нападая на компьютеры друг друга с использованием Интернета (хотя это, вероятно, не планировалось правительствами ни одной из сторон).

В прошлом военные и гражданские системы были отдельными и несхожими: различная аппаратура, различные протоколы связи — все различное. За последнее десятилетие все изменилось; прогресс в технологии произошел слишком быстро для традиционного у военных многолетнего цикла перестройки. Все чаще и чаще коммерческие компьютерные системы используются военными. Это означает, что все нападения, которые проводятся против коммерческих компьютеров, могут проводиться и против военных — те и другие имеют одни и те же уязвимые стороны. И обе стороны конфликта могут использовать одинаковое оборудование и протоколы: TCP/IP, операционные системы Windows, спутниковые приемники GPS. Внешние сети командования американскими Стратегическими воздушными силами (SAC) недавно были переведены на Windows NT.

Военные боролись с инфраструктурой противника с тех пор, как начали воевать. Средневековые рыцари убивали крепостных, наполеоновские армии жгли посевы, бомбардировщики союзников целенаправленно бомбили немецкие фабрики в течение Второй мировой войны. Сегодня информация — это инфраструктура. В ходе операции «Буря в пустыне» американцы систематически подрывали иракскую инфраструктуру командования и управления. Системы связи были заглушены, кабели индивидуальной связи являлись целью бомбежек. Без командования и управления наземные отряды были почти бесполезны. Ажиотаж в средствах массовой информации вокруг информационной войны смущает, но военные говорят об этом серьезно. Вот цитата из китайской армейской газеты «Жефанг¬ жунбао» — резюме речей, произнесенных в мае 1996 года:

«После войны в Заливе, когда все ожидали вечного мира, возникла новая революция в военном деле. Эта революция, по существу, есть переход от механизированной войны в индустриальную эпоху к информационной войне в информационную эпоху. Информационная война — это война решений и контроля, война знаний и война интеллектов. Цель информационной войны постепенно изменится от „сохранения себя и уничтожения врага" к „сохранению себя и управлению противником". Информационная война включает радиоэлектронную войну, тактический обман, стратегическое сдерживание путем устрашения, противостояние пропаганды, психологическую войну, войну в сетях и структурный саботаж. В сегодняшних технологических условиях всепобеждающий принцип, сформулированный Сан Цзу больше двух тысячелетий назад, — „победить врага без борьбы" и подчинить врага „мягким ударом"— может наконец быть понят правильно».

Война — это не обязательно глобальный конфликт, подобно Второй мировой войне или противостоянию Соединенных Штатов и СССР, которое могло привести к концу света. Более вероятно, что это — «тлеющий конфликт»: «Буря в пустыне», аргентинское вторжение на Фолклендские острова, гражданская война в Руанде. В «Трансформации войны» Мартин ван Кревелд указывает, что так называемые тлеющие конфликты стали после Второй мировой войны доминирующей формой столкновений, уничтожившей более 20 миллионов людей во всем мире. Это изменение — результат двух главных тенденций. Первая состоит в том, что

небольшим группам стало гораздо легче прибрать к рукам оружие массового поражения: химическое оружие, биологическое оружие, ракеты дальнего радиуса действия и т. д. Вторая заключается в том, что все больше межнациональных группировок стали способны вести войну. Фактически различие между государствами и межнациональными группировками размыто. Организованные преступные группы в таких странах, как Мексика, Колумбия и Россия, объединяются с правительством на различных уровнях. Не все инфовоины работают в интересах главных индустриально развитых стран. Все больше и больше они работают для второстепенных политических сил.

# Глава 5. Потребность в секретности

В каких же видах секретности мы нуждаемся? Прежде чем обсудить (а может быть, и отвергнуть) специфические контрмеры против нападений, уже названных нами, остановимся и поговорим о том, что же нам нужно. Какая степень секретности необходима в сегодняшнем компьютеризированном, интернациональном, взаимосвязанном и взаимозависимом мире?

## Секретность

Обыватели имеют общее представление о секретности. Когда же за нее просят платить, они не хотят этого делать. Предприниматели также имеют общее представление о секретности. Они нуждаются в ней, отлично понимая, что в случае ее отсутствия все их грязное белье будут полоскать газеты, — и они даже готовы платить за нее: за замки, сигнализацию, брандмауэры и корпоративные службы безопасности. Но когда обстоятельства подгоняют и работа должна быть быстро завершена, секретность оказывается первой вещью, которой пренебрегают. Правительства чувствуют себя комфортно только в условиях секретности: они понимают важность того обстоятельства, что их военные тайны могут попасть в руки врагов. Они нуждаются в ней и готовы заплатить за нее дорого. И они несут бремя секретности. Они готовы поступиться деталями, но сохранить основную идею.

Почти никто не понимает точно, как важна секретность в его жизни. Верховный суд утверждает, что это право, гарантируемое в соответствии с Конституцией. Демократия построена на понятии секретности: без этого вы не можете быть уверены в секретности избирательного бюллетеня. Предприниматели не могут работать, не имея никакого понятия о секретности; многочисленные работники компании должны владеть приватной информацией, которую люди вне компании иметь не должны. Люди хотят иметь гарантию секретности своих бесед и бумаг.

В Соединенных Штатах частные лица не имеют монопольного права собственности на любую информацию о себе. Списки клиентов принадлежат бизнесменам, которые их собирают. Личные записи в базе данных принадлежат владельцу базы данных. Только в редких случаях отдельным личностям обеспечены какие-либо права или защита от сбора частной информации.

Большинство стран имеют законы, защищающие частную жизнь. В Европейском Союзе, например, действует Закон от 1998 года о защите данных (Data Protection Act). Организации, которые собирают персональные данные, должны заре¬

гистрироваться в правительстве и обеспечить меры против злоупотребления ими. Они не имеют права собирать, использовать и распространять сведения частного характера без согласия того, к кому они относятся.

Организации также должны сообщать частным лицам о причинах сбора информации, обеспечивать доступ к ней, исправлять неточности и охранять эту информацию от доступа неправомочных сторон. Люди имеют право видеть собранные о них персональные данные и исправлять в них погрешности. Они также имеют право знать, для чего собираются эти сведения, и убеждаться, что информация не будет продана для других целей. И они также имеют право «уклоняться» от любого сбора сведений, когда они этого не хотят. Сборщики данных должны отвечать за защиту индивидуальных данных в разумно высокой степени и не делиться данными с тем, кто нетвердо придерживается этих правил.

Последний пункт стал причиной осложнений между ЕС и Соединенными Штатами, так как Соединенные Штаты не ведут никакого контроля за сбором персональных данных и позволяют компаниям покупать и продавать последние по желанию. Соединенные Штаты и ЕС в порядке эксперимента приняли соглашение относительно условий секретности для американских компаний, что должно было обеспечить «адекватный» уровень секретности к июлю 2001 года. Некоторые члены Конгресса несколько раз пробовали изменить законодательство о секретности, но под давлением промышленников эти попытки были блокированы. Группа лоббирования NetCoalition.com, которая включает в себя AOL, Amazon.com, Yahoo!, eBay и DoubleClick, верит в саморегулирование, которое является эквивалентом отсутствия секретности. К сожалению, большая часть деловых людей считают, что секретность плоха для бизнеса и что нарушение права на защиту частной информации — иногда единственный путь заработать деньги.

О деловой секретности. Предприниматели вообще не нуждаются в долгосрочной секретности. (Торговые секреты — формула кока-колы, например, — являются исключениями.) Клиентские базы данных должны оставаться конфиденциальными в течение многих лет. Данные развития производства — только несколько лет, а в бизнесе, связанном с компьютерами, намного меньше этого. Информация об общем финансовом здоровье, деловых переговорах и тактических маневрах — от недель до месяцев. Маркетинг и планы производства, стратегии, долгосрочные переговоры — от месяцев до нескольких лет. Подробная финансовая информация, возможно, нуждается в секретности в течение нескольких лет, но, вероятно, не больше. Даже общие пятилетние планы после девяти месяцев устаревают. Мы живем в мире, где информация распространяется быстро. Деловые тайны прошлой недели к этой неделе вытеснились новыми. А деловые тайны этой недели — это заголовки Wall Street Journal следующей недели.

Правительства также нуждаются в краткосрочной секретности. Интересы любой страны вынуждают ее следить за интересами других государств, и правительства заинтересованы в сохранении определенной информации в тайне от других государств. К несчастью, страны намного больше, чем компании. Невозможно сообщить каждому гражданину США секретную информацию так, чтобы при этом не произошла ее утечка к китайскому правительству. Таким образом, если Соединенные Штаты хотят сохранить секрет от китайцев, они должны также тщательно охранять его от большинства американцев.

Эти секреты обычно по своей природе — военные: стратегии и тактики, возможности оружия, разработки и обеспечение, мощь войск и их передвижение, научные исследования и разработки. Военные секреты часто перерастают в государственные: позиции на переговорах о соглашениях и т. п. Часто они пересекаются с корпоративными секретами: военными контрактами, положением на торгах, импортными и экспортными сделками и т. д.

Исключения для краткосрочной секретности связаны с затруднениями: личными, политическими или деловыми. Правительства не хотят, чтобы информация об их политических трудностях просочилась в прессу. (Вспомните Уотергейт. Вспомните мятеж в Иране. Вспомните любой политический скандал, раскрытый средствами массовой информации.) Люди не хотят обнародования своего личного прошлого. (Вспомните Билла Клинтона. Вспомните Боба Ливингстона, конгрессмена и спикера, который ушел в отставку в 1999 году после того, как было обнародовано дело двадцатилетней давности. Вспомните Артура Аша, чье заболевание СПИДом было обнаружено прессой.) Приблизительно через два десятилетия у нас будут выборы, в которых кандидаты окажутся перед необходимостью объяснять послания по электронной почте, которые они написали, когда были подростками.

Немногие случаи, которые требуют очень продолжительной секретности — из тех, что я знаю, — связаны с правительством. Американские данные переписи — имеются в виду оригиналы, а не что-либо, подвергшееся обработке, — должны остаться тайной в течение 72 лет. Мандаты ЦРУ, которые идентичны шпионским, остаются секретными, пока не переживут и шпиона, и его детей. Канадские данные переписи остаются секретными навсегда.

## Многоуровневая секретность

Военные обладают большим количеством информации, которая должна храниться в тайне; разные части этой информации имеют различную степень секретности. Местоположение морских судов может быть интересно врагу, но коды запуска ракет на этих судах намного важнее. Количество шинелей в поставках крайне интересно, но количество винтовок более важно.

Имея дело с подобными вещами, военные заинтересованы в многоуровневой классификации секретности. У военных США данные могут являться несекретными, конфиденциальными, секретными и совершенно секретными. Правила устанавливают, какие данные к какому уровню классификации относятся, и обусловливают различные правила хранения, распространения и т. д. Например, требуются сейфы различной прочности для хранения данных разного уровня секретности. Данные высшей степени секретности могут храниться только в надежно охраняемом, лишенном окон помещении без фотокопировальных устройств; за них должны расписываться.

Люди, работающие с этими данными, должны проходить проверки, соответствующие информации самой высокой степени секретности из всех, с которыми они работают. Кто-нибудь с секретным допуском, например, может видеть информацию и несекретную, и конфиденциальную, и секретную. Кто-то с допуском к конфиденциальной работе может видеть только несекретные и конфиденциальные

данные. (Конечно, допуск не является гарантией доверия. Глава российского отдела контрразведки ЦРУ Олдридж Эймс имел высший допуск секретности, но при этом он был российским шпионом.)

Данные на высшем уровне секретности иногда подразделяются по темам или по разделам, имеющим к ним отношение. На эти документы ставится гриф «TS/SCI» (высшая секретность/специальный раздел сведений). Каждый раздел имеет ключевое слово. TALENT и KETHOLE, например, являются ключевыми словами, связанными со спутниками-шпионами KH-11. SILVER, RUFF, TEAPOT, UMBRA и ZARF — с другими. (UMBRA применяется к информации о коммуникациях, RUFF соотносится с изображениями.)

Разделы — это важные барьеры доступа: кто-нибудь, имеющий доступ высшей степени секретности с дополнительным доступом KEYHOLE (иногда называемым «билетом»), не уполномочен видеть данные высшей степени секретности COBRA.

Дробление на разделы формально отражает понятие «потребности знать». Наличие у кого-либо некоторого уровня доступа не означает, что он автоматически получает возможность видеть все данные на этом уровне доступа. Он может получать для просмотра только те данные, которые должен знать, чтобы выполнять свою работу. Имеются и другие обозначения, которые уточняют классификацию: NOFORN — «Не для иностранных подданных», WNINTEL — «Внимание: источники и методы разведки», LIMDIS — «Ограниченное распространение».

В других странах также существуют подобные правила. В Великобритании один дополнительный уровень классификации — ограниченный, который находится между несекретным и конфиденциальным. Соединенные Штаты имеют нечто подобное, называемое FOUO (For Official Use Only) — только для служебного пользования — что означает «несекретно, но не подлежит сообщению посторонним лицам».

Здесь существенны два момента. Во-первых, все это намного легче осуществить на бумаге, чем на компьютере. В главе 8 говорится о некоторых из многоуровневых систем секретности, которые были построены и использовались, но ни одна из них никогда не работала в крупном масштабе. Во-вторых, эти системы в значительной степени не соответствуют враждебному внешнему окружению. В них не признаются ни корпоративные тайны, ни личные секреты. Секретность в реальном мире не вписывается в узкие иерархические рамки.

## Анонимность

Нуждаемся ли мы в анонимности? Хорошая ли это штука? В Интернете горячо обсуждалась целая концепция анонимности, взвешивались самые разные мнения людей по этому спорному вопросу.

Любой, кто работает на кризисной телефонной линии — будь то звонки о самоубийстве или насилии — знает силу анонимности. Тысячи людей в Интернете обсуждают свою личную жизнь в конференциях для оставшихся в живых жертв злоупотребления наркотиками, больных СПИДом и т. д.; они желают делать это только через анонимную пересылку. Это — социальная анонимность и она жизненно важна

для здоровья мира, потому что позволяет людям обсуждать такие вещи, говоря о которых нет желания указывать свое имя. Например, послания по почте некоторых людей в alt.religion.scientolgy[1] сделаны анонимно, иначе они не стали бы писать.

Политическая анонимность также важна. Не бывает и не может быть того, чтобы все политические выступления были подписаны. Так же как кто-то может совершить массовую политическую рассылку по почте без обратного адреса, может сделать то же самое через Интернет. Это играет важную роль в некоторых случаях: в 1999 году онлайновая анонимность позволила сербам, представителям Косово и другим прекратить Балканскую войну; они посылали новости о происходящем конфликте в остальные части света без непосредственного риска для жизни из-за раскрытия своей личности.

С другой стороны, люди используют анонимность Интернета, чтобы рассылать угрожающие сообщения по электронной почте, печатать речи, полные ненависти и оскорблений, распространять компьютерные вирусы и иным способом досаждать нормальным гражданам киберпространства.

Есть два различных вида анонимности. Первый — полная анонимность: письмо без обратного адреса, сообщение в бутылке, обращение по телефону без автоответчика или телефонной идентификации. О человеке, создающем полностью анонимный контакт, никто не может выяснять, кем он является, и если, что еще более важно, этот человек еще раз вступает в контакт, контактирующий не знает, что он имеет дело с тем же самым человеком.

Второй тип анонимности связан с использованием псевдонима. Подумайте о счете в швейцарском банке (хотя это фактически прекращено в 1990 году), о почтовом ящике, арендованном за наличные под вымышленным именем (хотя это больше невозможно в Соединенных Штатах без поддельного удостоверения), о встречах анонимных алкоголиков, где вы известны только как Боб. Эта анонимность заключается в том, что никто не знает, кто вы, но есть возможность идентифицировать вас по этому псевдониму. Это то, что нужно швейцарским банкам: их не заботит, кто вы, — только то, что вы являетесь тем же самым человеком, который внес деньги на прошлой неделе. Торговец не должен знать ваше имя, но должен знать, что вы законно купили товар, который теперь пытаетесь возвратить.

Оба типа анонимности сложно осуществить в киберпространстве, потому что многое в его структуре требует установления личности. Новые микропроцессоры класса Intel Pentium III имеют уникальные серийные номера, которые могут быть отождествлены так же, как и сетевые карты в компьютерах локальных вычислительных сетей. Документы Microsoft Office автоматически сохраняют информацию, указывающую на автора. Веб-серверы прослеживают людей в Сети; даже по анонимным обращениям по электронной почте теоретически можно вычислить реального человека, если отследить IP-адрес. Много недостатков было найдено в различных программах, которые обещали анонимную работу. Поверхностная анонимность проста, но истинная анонимность, вероятно, в сегодняшнем Интернете невозможна.

---

[1] Конференция в USENET. — *Примеч. ред.*

## Коммерческая анонимность

Понятие псевдонима приносит нам приятную анонимность в финансовых сделках. Но с другой стороны, ее же используют и недобросовестные продавцы, не несущие за товар никакой ответственности: это — ничей бизнес, не принадлежащий ни правительству, ни оптовым, ни мелким торговцам, — что бы люди не покупали, будь то порнофильмы или сюрпризы к дню рождения. К сожалению, существует еще и большая группа негласных сторонников финансовой анонимности: торговцы наркотиками и другие темные элементы. Можно ли примирить две эти стороны?

Очевидно, можно, потому что существуют наличные деньги. Реально вопрос состоит в том, хотим ли мы когда-нибудь получить электронную версию наличных денег. Я не верю, что хотим, может быть, исключая лишь небольшие сделки.

Анонимность дорого стоит, потому что с ней связаны дополнительные риски. (Правительственное регулирование также влияет на это.) Банки не глупы, они предпочитают менее опасную систему. И анонимная система обходится дороже, чем система, основанная на учетных записях и отношениях. Банки могут вложить дополнительные затраты в систему, но клиенты не желают за это платить. Если вы — торговец, то проведите эксперимент. Поднимите цены в своем магазине со словами: «Пять процентов скидки, если вы дадите нам свое имя и адрес и позволите проследить ваши привычки в покупках». Посмотрите, сколько клиентов предпочитает анонимность. Люди скажут, что они не хотят попасть в мегабазы данных, прослеживающие каждую их денежную трату, но желают получить такую симпатичную карту постоянного клиента, которая предоставляет все возможности выиграть бесплатное путешествие на Гавайи (одно на тысячу карт). Если Макдоналдс предлагает три бесплатных Биг-Мака за образец дезоксирибонуклеиновой кислоты, то значит, что на это есть причина.

С другой стороны, поднимите цену, сказав: «Пять процентов скидки, если вы сообщите нам название и адрес школы вашего ребенка», и вы, вероятно, увидите другую реакцию. Есть некоторые вещи, которые большинство людей хотят сохранить в секрете, и есть люди, которые хотят держать в секрете большинство вещей. Всегда будет существовать стиль швейцарского банка — анонимной платежной системы для богатых, которые готовы нести расходы за сохранение секретности. Но средний потребитель далек от богатых. Среди средних потребителей есть некоторые исключения, но вообще-то они не заботятся об анонимности. А у банков нет никакой причины предоставлять им ее, особенно если правительство не заставляет их этого делать.

## Медицинская анонимность

А еще существуют медицинские базы данных. С одной стороны, медицинские данные только полезны, если использовать их по назначению. Доктора должны знать историю болезни своих пациентов, а общие медицинские данные нужны для всех видов исследований. С другой стороны, медицинская информация рассказывает о пациенте без прикрас: генетическая предрасположенность к болезням, аборты и репродуктивное здоровье, эмоциональное и психическое здоровье, злоупотребление наркотиками, сексуальные реакции, болезни, переданные половым путем, ВИЧ-

статус, физические отклонения. Люди имеют право хранить медицинские данные о себе в секрете. После того как личная медицинская информация обнародована, людей могут беспокоить, угрожать им и даже обстреливать.

Эту информацию получить нетрудно. Медицинские записи о Николь Браун Симпсон были опубликованы через неделю после ее убийства в 1994 году. А в 1995-м Лондонская *Sunday Times* сообщила, что цена каждой медицинской записи в Англии составляет 200 фунтов стерлингов. И если так обстоят дела в богатых странах, то вы только вообразите, какие злоупотребления могут стать возможными в странах типа Индии и Мексики, где сумма в 10 долларов способна соблазнить даже наиболее добродетельного государственного служащего.

Компьютеризированные данные пациентов плохи с точки зрения секретности. Но они хороши для всего остального, так что их сбор неизбежен. В Законе о мобильном и общедоступном страховании здоровья (HIPAA) теперь есть стандарты для компьютеризированных медицинских записей. Они делают информацию легко доступной там, где в этом есть необходимость, что удобно для населения, которое реже пользуется услугами семейного доктора и чаще ездит по всей стране, посещая различных докторов и клиники в случае необходимости. Специалисты могут легко получать нужные данные. Так же поступают и страховые компании, потому что такие данные более содержательны, более стандартизированы и более дешевы для обработки: если все данные электронные, то проверять клиентов будет дешевле. Это лучше и для исследователей, потому что позволяет им эффективнее использовать доступную информацию: впервые они могут смотреть на все в стандартной форме.

Это большое мероприятие, вероятно, столь же важное, как ранее упомянутые финансовые и кредитные базы данных. Как общество мы окажемся перед необходимостью сбалансировать потребность в доступе (который более очевиден для медицинской, чем для финансовой информации) с потребностью в секретности. Так или иначе, компьютеризация приходит в медицину. Мы должны быть уверены, что все сделано правильно.

## Секретность и правительство

Правительство и ФБР, в частности, любят изображать частную секретность чудовищным инструментом четырех всадников Апокалипсиса: террористов, торговцев наркотиками, тех, кто отмывает деньги, и тех, кто занимается детской порнографией. В 1994 году ФБР протаскивало через Конгресс законопроект о цифровой телефонной связи, согласно которому телефонные компании должны были бы установить на своих коммутаторах оборудование, позволяющее легко подключаться к любой линии. После таранов Центра международной торговли оно продвигало законопроект о всесторонней борьбе с терроризмом, который дал бы ему полномочия проводить прослушивание телефонных переговоров и наделил бы Президента властью единолично объявлять политические группы террористическими организациями. К счастью, это не прошло. После падения из-за взрыва топливного бака самолета в 1996 году ФБР играло на слухах о том, что это было попадание ракеты, и принимало ряд мер, которые разрушали частную секретность. Оно продолжает

лоббировать предоставление правительственного доступа ко всем шифровальным ключам или ослабление защиты до того уровня, когда она уже не будет играть никакой роли.

В течение нескольких последних десятилетий развитие компьютерной секретности в Соединенных Штатах было ограничено тем, что называется *экспортными законами*. Экспортные законы определяют те виды шифрования, которые компании США могут экспортировать. Так как большинство программных продуктов распространено по всему миру, такие законы значительно ограничивали эффективность шифрования в массовых программах, подобных браузерам и операционным системам[1].

В 1993 году американское правительство отстояло так называемый «Клиппер-Чип», который будет обсуждаться подробно в главе 16. Это система, которая дает полицейским доступ к вашим ключам кодирования.

Дебаты продолжаются. ФБР стремится к узакониванию прав, нарушающих секретность: прослушиванию широкополосных телефонных сетей, установке подслушивающих устройств в компьютеры пользователей без ордера везде, где возможно. Во время написания книги у нас появились новые экспортные правила для программного обеспечения, представленного на массовом рынке, разнообразие законопроектов либерализации шифрования находится в Конгрессе, и несколько дел о контроле над экспортом направлены в Верховный суд. Изменения происходят постоянно; что-либо сказанное мной здесь может устареть к моменту издания книги.

Также интересны (и бесконечны) философские проблемы. Первое — правильно ли думает правительство, когда оно предполагает, что социальные беды от секретности перевешивают социальные преимущества? В предыдущем разделе я приводил доводы, что преимущества анонимности перетягивают связанные с ней проблемы. То же самое с секретностью. Она чаще всего применяется там, где нужно, и положительных сторон ее использования намного больше, чем отрицательных.

Второе — может ли правительство закупать технологию, которая совершенно очевидно приносит большие социальные выгоды, но, с другой стороны, некоторым образом препятствует законному принуждению, так что, по идее, необходимо ограничивать ее использование? Ключевой козырь ФБР — это шифрование, являющееся большой помехой для расследования уголовных дел, так как ФБР име¬

[1] Из США до самого последнего времени был запрещен экспорт алгоритмов шифрования с длиной ключа более 40 бит. Такая криптостойкость уже не считается надежной при современных вычислительных мощностях и даже на персональном компьютере. Бюрократия вносит свой вклад в политику администрации. Оригинальные версии популярного архиватора ARJ, начиная с 2.60, оказались недоступны российским пользователям только потому, что используемый в них криптостойкий российский алгоритм ГОСТ также запрещен к экспорту. Шифрование же в более ранних версиях (RC4) крайне ненадежно. Декабрь 1998 напугал многих. В результате борьбы с ограничениями экспорта из США 33 страны-участницы Вассенаарского соглашения (включая Россию) согласились взять под контроль программное обеспечение, содержащее средства шифрования с ключами длиной в 64 битов и более, то есть ограничить свободный доступ к ним. Но в 2000 году ограничения были смягчены: длина ключа увеличена до 56 бит (при этом поставщики остаются владельцами дешифрующего ключа на всякий случай). Компаниям с филиалами за рубежом разрешено было применять один и тот же алгоритм шифрования. — *Примеч. ред.*

ет те же самые возможности подслушивания, что и десять лет назад. Однако подслушивание не обеспечивает доказательств, и история убедительно показывает, что перехват — это нерентабельный метод борьбы с преступностью. Широко распространенное шифрование может быть шагом назад в осуществлении механизмов законного принуждения, но не в обвинении преступников.

Я не знаю ответов. Существует равновесие между секретностью и безопасностью. Законы, которые регламентируют розыскную деятельность и соблюдаются должным образом, препятствуют юридическому принуждению, и это может кончиться тем, что некоторые преступники получат свободу. С другой стороны, эти законы защищают граждан от злоупотреблений полиции. Мы, как общество, должны решить, какое равновесие является правильным для нас, и затем создать условия для законного проведения его в жизнь. Но я громогласно возражаю против ФБР, старающегося насадить выгодное ему решение без общественного обсуждения и без общественного понимания.

В любом случае будущее не оптимистично. Право на секретность — это первое, чем пренебрегают в случае кризиса, и ФБР попробует сфабриковать кризисы, чтобы попытаться захватить большее количество полномочий для вторжения в секретность. Война, террористические нападения, полицейские акции... наверняка вызовут большие изменения в точках зрения. И даже сейчас, в обстоятельствах, наиболее способствующих аргументированным дебатам о секретности, мы теряем ее все больше и больше.

# Аутентификация

Секретность и анонимность могут быть важны для нашего общественного и делового благосостояния, но аутентификация необходима для выживания. Аутентификация, давая информацию о том, кому можно и кому нельзя доверять, служит непрерывному возобновлению отношений, придающих смысл сложному миру. Даже животные нуждаются в аутентификации запаха, звука, касания. Возможно, сама жизнь — это распознавание молекулярного состава ферментов, антител и т. д.

Люди аутентифицируют себя огромное количество раз в день. При входе в компьютерную систему, вы подтверждаете свою подлинность компьютеру. В 1997 году управление социального обеспечения пробовало ввести данные людей в сеть; они прекратили это после жалоб на то, что номер социального обеспечения и девичья фамилия матери не являются достаточно хорошими опознавательными средствами и что у людей будет возможность читать данные других людей. Компьютер также должен подтвердить свою подлинность вам; в противном случае как вы узнаете, что это ваш компьютер, а не какой-нибудь самозванец?

Посмотрите на среднего человека на улице, собирающегося купить пирог. Он рассматривает витрину за витриной, ища тот магазин, в котором продают пироги. Или, возможно, он уже знает свою любимую булочную, но еще только идет туда. В любом случае, когда он добирается до магазина, он подтверждает подлинность того, что это — правильный магазин. Аутентификация сенсорная: он видит пироги в меню, чувствует их запах в воздухе, магазин выглядит точно так же, как тогда, когда он последний раз был в нем.

Человек говорит с продавцом магазина и спрашивает о пироге. В некоторой степени оба аутентифицируют друг друга. Продавец хочет знать, способен ли клиент заплатить. Если клиент одет в тряпье, продавец может попросить, чтобы он ушел (или, по крайней мере, заплатил сразу). Если клиент носит лыжную маску и размахивает АК-47, продавец, скорее всего, убежит сам.

Клиент также аутентифицирует подлинность торговца. Он на самом деле продавец? Он продаст мне пирог или даст мне только кучу опилок в булочке? Что сказать о булочной? В наличии есть некое свидетельство о чистоте, подписанное местным санитарным инспектором, оно висит где-то на стене, если вдруг клиент захочет проверить. Но чаще клиент доверяет своим инстинктам. Мы все уходим из булочной, если нам не понравилось «ощущение» этого места.

Торговец вручит пирог, а клиент заплатит 5 долларов по счету. Еще большая аутентификация. Действительно ли этот счет подлинный? Выглядит ли этот пирог съедобным? У нас настолько хорошо развита зрительная (и обонятельная) аутентификация, что мы не задумываемся об этом, но поступаем так все время. Клиент получит сдачу, посмотрит на чек, чтобы удостовериться, что он пробит законным предпринимателем, и положит его в карман.

Если бы клиент платил по кредитной карте, за этим последовало бы еще больше аутентификации. Торговец прогнал бы карту через считывающее контрольное устройство, которое связалось бы с центральным сервером, и убедился бы, что счет действителен и кредита на нем достаточно для покупки. Торговец исследовал бы карту, чтобы убедиться, что это не подделка, и проверил бы подпись на ее оборотной стороне. (Большинство торговцев, правда, не будут так беспокоиться, особенно если сделка незначительна по сумме.)

Если бы клиент платил чеком, был бы другой опознавательный ритуал. Торговец посмотрел бы на чек и, возможно, спросил бы клиента о некоторых идентификационных данных. Он мог бы записать номер водительского удостоверения клиента и номер его телефона на обороте чека или, допустим, номер кредитной карточки клиента. Ни одно из этих ухищрений фактически не позволяет сделать вывод, что чек действителен, но помогает проследить за клиентом в случае, если возникнут проблемы.

Подделка аутентификации может быть очень выгодна. В 1988 году Томпсон Сандерс был осужден за обман Чикагского управления торговли. Он изображал торговца — полного, в парике, с бородой и поддельными документами. Этот поддельный торговец разместил большие рискованные заказы, затем заявил свои права на те, которые были выгодны, а от тех, с кем сделки оказались убыточными, просто скрылся. Брокеры, участвовавшие в этих сделках с другой стороны, были не способны определить, кто участвовал в торге, и понесли ответственность за ущерб.

Вернемся к нашему торговцу. Приходит другая клиентка. Она и торговец — старые друзья. Каждый из них знает другого в лицо. Это — здравая система подтверждения: они узнают друг друга даже при том, что у нее новая прическа, а он носит новый парик и очки. Супергерои понимают это и носят маски, чтобы сохранить свою личность в секрете. Это больше подходит для комиксов, чем для реальной жизни, потому что аутентификация — это не только распознавание лица (иначе слепой никогда никого не узнал бы). Люди помнят голос друг друга, фигуру,

особенности и т. д. Если торговец говорит со своим другом по телефону, они могут подтвердить личности друг друга вообще без визуального контакта. Специальный уполномоченный Гордон должен был понять, что Брюс Уэйн — на самом деле Бэтман, просто потому, что они так часто разговаривали по телефону.

В любом случае наш клиент, купивший пирог, закончил его есть. Он произносит «до свидания», будучи уверенным в том, что говорит это тому же самому продавцу, который обслуживал его. Он выходит через ту же самую дверь, в которую вошел, и идет домой.

Все достаточно просто, потому что каждый бывал в таких магазинчиках. Платон не доверял написанному, потому что не мог определить, что является правдой, если человек не находится прямо перед ним. Что бы он сказал о Всемирной Сети: никакого голоса, никаких лиц... только биты.

Тот же самый клиент, купивший пирог, теперь возымел желание купить что-нибудь менее скоропортящееся — рецепт приготовления пирога, например. Для этих поисков он заходит в Сеть, пользуется своим испытанным поисковым сервером и находит несколько веб-сайтов, где продаются рецепты пирогов. Все они принимают к оплате кредитные карточки через Интернет или позволяют сделать заказ по почте. Все они обещают доставку в три-четыре дня. Что теперь?

Как бедный клиент узнает, можно ли им доверять? Потребуется немного усилий, чтобы просмотреть предложения: в Сети любой может сделать это в течение нескольких минут. Но какие из продавцов честны, а какие занимаются жульничеством? URL — указатель информационного ресурса (строка символов, указывающая на местонахождение документа в Интернете) мог бы быть именем продавца рецептов, которому можно доверять, но где гарантия, что сайт действительно соответствует тому самому доверенному имени? У Северо-Западных авиалиний есть веб-сайты, где можно купить авиабилеты: www.nwa.com. До недавнего времени у туристических агентов был веб-сайт www.northwest-airlines.com. Сколько людей купили билеты у последних, думая, что покупают их у первых? (Многие компании не знают доменных имен своих тезок.) Некоторые компании помещают имена своих конкурентов в описание своих веб-сайтов (обычно скрытые) в надежде обмануть поисковые серверы. Internic.net, где вы собираетесь зарегистрировать доменное имя, — не то же самое, что Internic.com. Последний возник как надувательство, сформировавшись внутри Internic Software, и в настоящее время якобы регистрирует имена доменов. Хозяева, вероятно, сделают значительный бизнес за счет создания путаницы. Есть даже более мрачное предположение: кто может сказать, что некий незаконный хакер не убедил программу просмотра отображать один URL вместо другого?

Клиент находит веб-сайт, который выглядит подходящим, и выбирает рецепт пирога. Теперь он должен заплатить торговцу. Если он покупает что-либо ценное, то в этом случае нужна серьезная аутентификация. (Если он тратит 25 центов на виртуальную газету, все немного проще.) Действительны ли эти электронные деньги? Действительна ли эта кредитная карточка и есть ли у клиента право выписывать электронный чек? Некоторые торговцы, работающие непосредственно с клиентом, просят показать водительские права перед тем, как принять чек; а что же может проверить цифровой торговец перед принятием электронного чека?

Наиболее важной проблемой безопасности является аутентификация через цифровые сети. Здесь может быть так же много различных решений, как и различных требований. Некоторые решения должны быть сильными и весомыми, чтобы защитить миллионы долларов. Для других это не обязательно: например, аутентификация дисконтной карты торговца. Некоторые решения подразумевают анонимность — наличные деньги или карту, которая пускает вас в специфическую область сети, не требующую обязательного раскрытия вашего имени, — в то время как другие нуждаются в строгой системе аутентификации. Большинство будут стремиться к интернациональности: сетевой паспорт, системы, используемые для международной торговли, цифровые подписи или международные контракты и соглашения.

Часто аутентификация осуществляется невидимо для пользователя. Когда вы используете свой телефон (или платный телевизионный канал), то аутентифицируете себя в сети так, чтобы было известно, кому выставлять счет. Военная авиация имеет системы IFF (позволяющие узнать — друг перед вами или враг) для опознавания своих собственных и союзных самолетов системами ПВО. Тахографы, применяемые повсюду в грузовиках в Европе, чтобы заставить водителей соблюдать правила — такие, как принудительный отдых, — используют методы аутентификации, чтобы предотвратить мошенничество. Предоплата электричества в Великобритании — другой пример.

Когда думаете об аутентификации, держите в уме два ее различных типа. Они могут выглядеть похожими, но техника их использования очень различна. Первый — это *аутентификация сеанса:* беседа лицом к лицу или по телефону, или через IRC (международную линию передачи документальной информации). Сеансами также могут быть разовые посещения интернет-магазина. Метод аутентификации здесь — это сравнение отдельных диалогов: является ли лицо, сказавшее что-либо сейчас, тем же самым лицом, сказавшим что-либо ранее? (Это легко сделать при переговорах по телефону или при личной встрече — если голос и внешность те же самые, то, вероятно, это один и тот же человек. В Сети значительно сложнее.)

Второй — это *аутентификация транзакции:* покупки с использованием кредитной карты, частично денежное обращение. Аутентифицируется здесь действительность сделки: признают ее стороны или же будет вызвана полиция. Споры при обсуждении этой стороны вопроса одни и те же, осуществляется сделка через Сеть, по телефону или же при личном контакте. Неважно, идет ли речь о проверке чека на 100 долларов торговцем, который должен удостовериться, что чек не поддельный, или о сопоставлении подписи на кредитной карточке с подписью в регистрационной карточке продаж.

# Целостность

Когда мы говорим об аутентификации, на самом деле имеем в виду целостность. Две эти концепции различаются, но иногда они переплетены. Аутентификация имеет дело с источником данных: кто подписал лицензию на медицинскую практику, кто выпустил в обращение валюту, кто санкционировал закупочный ордер на 200 фунтов удобрений и 5 галлонов дизельного топлива? Целостность имеет

отношение к действительности данных. Верен ли номер этой платежной ведомости? Были ли данные исследования окружающей среды изменены, с тех пор как я последний раз видел их? Целостность не имеет отношения к источнику данных, к тому, кто создал их, когда и как, но определяется тем, были ли данные изменены с момента их создания.

Целостность — это не то же самое, что точность. Точность характеризует степень соответствия данной величины ее истинному значению; целостность описывает отношение данной величины к самой себе через какое-то время. Часто они тесно взаимосвязаны.

В любом обществе, где компьютеризированные данные используются для принятия решений, целостность важна. Иногда это может иметь значение для общества в целом: если статистика о детях с уровнем жизни за чертой бедности является признанным фактом, ситуация может быть изменена путем выделения федеральных пособий. Для любого, кто зависим от акций NASDAQ (компьютеризированной системы котировки ценных бумаг), путаница может быть убийственной. Иногда это важно для отдельной личности: вы на самом деле можете создать беспорядок, внеся изменения в записи о водительских правах и отметив чью-либо лицензию как приостановленную. (Это было случайно сделано в 1985 году в Анкоридже, штат Аляска, по отношению к 400 людям, по крайней мере один человек из которых провел ночь в тюрьме. Подумайте об удовольствии, которое кто-нибудь может получить, сделав это специально.)

Было несколько инцидентов, связанных с целостностью и имевших отношение к акциям. В 1997 году у компании Swisher, которая производит дезодоранты для унитазов, сильно увеличилась стоимость акций из-за того, что информационные службы некоторое время путали символику ее акций с символикой акций другой компании с названием Swisher, которая производит сигары. Компания Swisher1 была намного меньше, чем Swisher2, поэтому когда вы, будучи введены в заблуждение, просматривали ее годовой отчет, то находили невероятную недооценку акций. Некие парни вычислили, что же на самом деле произошло, и быстро продали акции Swisher1, просчитав, что цена снова упадет, как только инвесторы поймут ошибку.

В 1999 году служащий PairGain Technologies отправил по почте поддельные объявления о слиянии компаний, оформленные так, как выглядят объявления информационной службы Блумберг, и получил контроль над 30 % акций до тех пор, пока обман не был раскрыт.

Эти случаи не имеют отношения к аутентификации. Не имеет значения, кто собрал данные переписи, кто составил конечные цены акций или кто ввел записи о регистрации автомашин — они касаются целостности. Но есть много других баз данных, где целостность важна: телефонные книги, медицинские записи, финансовые записи и т. д.

Когда один мой знакомый писатель-мистик появляется перед аудиторией, я всегда думаю, что хладнокровный способ убить кого-нибудь состоит в том, чтобы изменить базу данных дозировки препарата в больнице. Если врач недостаточно внимателен — например, он утомлен, препарат незнакомый, его отвлекает некий Мак-Гуффин, — он может прописать только то, что ему сообщает компьютер. Сегодня это может не сработать — рядом есть бумажный первоисточник, напри¬

мер настольный справочник врача или фармакологические стандарты препарата — но кто поручится за завтра?.. Миллионы людей получают медицинскую информацию по Сети. Например, drugemporium.com делает запрос другому сайту — drkoop.com, чтобы получить информацию о возможной несочетаемости препаратов, которые вам назначены. Пользователей обычно предупреждают, чтобы они не полагались на информацию, взятую всего из одного источника, но большинство из них все равно будут поступать так, а не иначе. Кто-нибудь захочет поиграть с целостностью этих данных и причинит много вреда.

И даже если нет никакого преступного намерения, в любой сетевой системе, которая имеет дело с рецептами и лечением, лучше проводить проверку целостности, чтобы застраховать себя от случайных ошибок: никто не хочет, чтобы случайно измененный байт привел в итоге к смертельному случаю в больнице — ни пациент, ни компания, занимающаяся поставками программного обеспечения, которой придется иметь дело с судебными процессами.

В физическом мире люди используют материальную копию объекта как доказательство целостности. Мы доверяем телефонной книге, настольным справочникам врача и «Американским статистическим отчетам», поскольку это книги, которые выглядят настоящими. Если они фальшивы, значит, кто-то тратит много денег, чтобы они выглядели настоящими. Когда вы снимете с полки роман Диккенса и начнете читать его, вы не усомнитесь в его реальности. Точно так же с вырезкой из *Business Week* — это всего лишь клочок бумаги, но он выглядит и воспринимается как страница журнала. Если вы получаете фотокопию журнальной вырезки, то она только напоминает страницу журнала. Если кто-нибудь перепечатает статью (или загрузит ее из LEXIS-NEXIS) и пошлет ее по электронной почте вам, тогда... кто знает.

1 августа 1997 года я получил электронную почту от друга; в ней была копия речи Курта Воннегута в день присуждения университетских степеней в Массачусетском технологическом институте. По крайней мере, я так предполагал. Мой друг переслал ее мне с честными намерениями. Но это не было речью Курта Воннегута на присуждении университетских степеней в 1997 году. В 1997 году Воннегут не выступал там. Он не писал этой речи и нигде не выступал с ней. Она была написана Мари Шмич и опубликована в ее колонке в *Chicago Tribune* 1 июля 1997 года.

Я сопоставил этот случай сомнительного авторства Воннегута с письмом, полученным мной приблизительно 15 лет назад, еще до появления Всемирной паутины, даже до того, как у меня появился адрес электронной почты (но уже во времена Интернета). Речь идет об эссе с названием «Мечта о будущем (не исключая омаров)»; друг отправил мне фотокопию по электронной почте. Копия была сделана непосредственно с публикации. Да, она могла быть сфальсифицирована, но это потребовало бы уйму работы. Это было до эры настольных издательских систем, и придать чему-либо вид фотокопии журнала *Esquire* было сложно и дорого. Сегодня отличить реальную вещь от «утки» уже сложно.

Я получал переданные по электронной почте статьи из журналов и газет. Кто может дать гарантию, что те статьи на самом деле из газет и журналов, хотя утверждается, что это так. Как я узнаю, что они не были искусно изменены: слово здесь, предложение там. Что если я сделаю эту книгу доступной интерактивно, и некие хакеры возьмут и изменят мои слова? Возможно, вы читаете эту книгу в Сети;

остановитесь ли вы, чтобы подумать, что прочитанное вами может не быть написано мною, что вы верите серверу, с которого загрузили книгу. Является ли вера механизмом, который применим для проверки, что это мои слова? По прошествии достаточного количества лет многие люди будут читать переделанную версию книги — иную, чем мои настоящие слова. Думаете, кто-нибудь заметит это? Насколько задолго до измененной версии была создана «настоящая» версия? Когда протест Воннегута будет забыт и его речь на присуждении ученых степеней войдет в историю?

Соблазн подделывать или изменять данные велик. Покрытый рунами камень, найденный в Миннесоте, предположительно описывает визит викингов, и ничего, что он содержит слова, возникшие только в современном шведском языке. Поль Шлиман (внук Генриха Шлимана) претендовал на открытие секрета Атлантиды в старинных свитках майя, которые он прочел в Британском музее. Ничего, что никто не может прочесть письмена майя и что старинные свитки хранятся в Мадриде. Переписанная Бисмарком телеграмма Эмса в 1870 году развязала Франко-Прусскую войну. В 1996 году, когда Дэвид Селборн пытался протолкнуть свой перевод описания посещения Китая итальянским путешественником (обогнавшим Марко Поло на три года), он использовал «владельца манускрипта, согласившегося на перевод только в случае, если будет соблюдена строжайшая его анонимность», в качестве уловки, чтобы скрыть подделку.

Цифровой мир позволяет с легкостью осуществлять подобные вещи, потому что подделку настолько же просто произвести, насколько сложно выяснить истину. В мае 1997 года 13-летняя жительница Бруклина выиграла национальный конкурс по орфографии. Когда в *New-York Post* была напечатана фотография *Associated Press,* на которой девочка прыгает от радости, со снимка убрали название ее спонсора, нью-йоркской *Daily News.* Так же на видео: когда Си-Би-Эс показывала празднование Нового, 2000 года она добавила свою собственную эмблему к эмблеме корпорации NBC (30 на 40 футов). А поддельные эссе и речи, подобные речи Воннегута, путешествуют по Интернету постоянно.

Изображения способны оказывать мощные воздействия на людей. Они могут изменять мнения и оказывать влияние на внешнюю политику. Картины «Бури в пустыне» — загнанные в ловушку иракцы, ставшие жертвами снарядов военно-воздушных сил коалиции, сыграли большую роль в быстром прекращении огня: американцы не любят видеть резню. А помните Сомали? Кадры были взяты из тридцатисекундного видеоклипа: мертвую Марине (Marine) проволокли по улицам Могадишу для того, чтобы отбить у американцев желание воевать. Информация — это сила. В некоторых случаях видеоклип может быть и обманом.

Это звучит жутко, но, несмотря на внимание к этой проблеме, мы теряем способность отличать настоящую вещь от фальсификации. На протяжении всей истории человечества мы использовали контекст для проверки целостности; в электронном мире контекста нет. Для кинофильма «Афера» Ньюман и Редфорд нанимали дюжины актеров на пробы и устраивали реалистичную имитацию тотализатора конных бегов, чтобы детально изучить поведение каждого кандидата. Во время съемок более современного кинофильма «Испанский заключенный» было то же самое. Вовлечение в такие глобальные игры с целью детального изучения реакций было популярно во времена депрессии; впрочем, я знаю, что подобные вещи дела¬

ются и сегодня. Данный способ оценки надежен, потому что человек не может предположить, что все, что он видит — комнаты, люди, движение, — в действительности является только представлением, разыгранным исключительно для него. В Сети это сделать просто. В мире, в котором нет возможности потрогать, людям нужен новый способ проверки целостности того, что они видят.

## Аудит

Двойная запись в бухгалтерском учете была придумана в 1497 году Лукой Пацио¬ли из Борго Сан-Сеполкро, хотя само это понятие на 200 лет старше. Основная идея в том, что каждая операция будет влиять на два или более счета. Один счет дебетируется на ту же самую сумму, на которую кредитуется другой счет. Таким образом, все операции всегда проходят по двум счетам и, поскольку они всегда показывают увеличение на одном счете и уменьшение на другом, суммарный итог по всем счетам всегда будет нулевым.

У этой системы есть две главные цели. Две книги хранятся у двух разных клерков, уменьшая возможность обмана. Но еще более важно, что две книги будут сбалансированы друг с другом в установленном порядке (бизнесмены должны подсчитывать баланс каждый месяц, банки каждый день). Процесс подсчета баланса и является аудитом: если один клерк пытается совершить злостную фальсификацию или просто сделает ошибку, это будет раскрыто, потому что кто-нибудь другой будет проверять его работу, а не он сам. В дополнение к этому могут пригласить ревизоров со стороны, когда придут другие бухгалтеры и проверят книги снова... только чтобы удостовериться.

Аудит жизненно необходим, если к безопасности относиться серьезно. Двойная запись бухгалтерского учета — это только начало; банки имеют комплексные и исчерпывающие требования к аудиту. То же касается тюрем, стартовых шахт ядерных ракет и бакалейных магазинов. В тюрьмах должны хранить записи на каждого, кто поступает и выбывает, и регулярно составлять баланс, чтобы быть уверенными, что никто незамеченным не убыл (или случайно не остался). В ракетных шахтах могут дополнительно подвергнуть ревизии каждый убывающий и прибывающий контейнер и упаковку, сравнивая отгрузочные и приемные записи с действительностью. Бакалейный магазин хранит кассовую ленту всего пробитого товара и сравнивает количество денег, находящихся в ящике кассы фактически, с тем, что пробито на кассовой ленте.

Это не профилактические меры безопасности (хотя они могут предотвратить нападения); аудит предназначен для того, чтобы помочь судам. Суть аудита состоит в том, чтобы вы смогли обнаружить успешное нападение, выяснить, что случилось после него, и затем доказывать наличие нападения в суде. Специфические потребности системы в аудите зависят от сферы его приложения и масштабов. Например, вам не нужно многое из контрольных функций аудита, применяемых к системе кредитных карт для обслуживания фотокопировальных машин университета. И есть потребность в гораздо более жестком контроле, если кредитные карты собираются использовать для проведения больших закупок, которые могут быть конвертированы в наличные деньги.

Аудит с трудом может быть применим к компьютерам. Регистрационная лента хорошо подходит для ревизии, потому что клерк не в силах изменить записи: операции последовательно напечатаны на едином рулоне бумаги, и невозможно добавить или удалить операцию, не вызвав подозрения. (Правда, есть некоторые способы: блокирование записи, имитация того, что закончились чернила, блокировка записи для отдельной операции, подделывание целой ленты и т. д.) С другой стороны, компьютерные файлы могут быть легко стерты или изменены; это делает аудиторскую проверку записей более сложной. Большинство проектировщиков систем не думают о ревизии, когда занимаются их разработкой. Вспомните заложенное изначально контрольно-ревизионное свойство двойной записи счетов бухгалтерского учета. Эта контрольная способность обречена на неудачу, когда обе книги хранятся в одной и той же компьютерной системе и один и тот же человек имеет доступ к обеим книгам. Но таким образом работают все бухгалтерские компьютерные программы.

## Электронные деньги

Вернемся в старые времена (год 1995 или около того). В те дни каждый думал, что мы должны создать новый вид денег для обращения в электронной торговле. Много компаний прекратили свое существование в попытках выдумать новые деньги. Некоторые компании старались создать электронный эквивалент наличных денег, другие — электронный эквивалент чеков и кредитных карт. Одной из последних таких попыток стал объединенный протокол Visa/Master card, предназначенный для использования существующих кредитных карт совместно с особой интернет-системой, позволяющей сделать кредитные карты надежными для электронной коммерции.

Они как-то изворачиваются, но не в этом дело. Кредитные карты прекрасно подходят для Интернета, и очень многие с готовностью пользуются ими для покупки книг, одежды и всего прочего. Однако наличие таких брешей в защите, которые позволили осуществить кражи серийных номеров кредитных карт в 2000 году, впечатляет. Будет ли когда-нибудь создана специфическая для Интернета форма оплаты?

В большей степени это вопрос регулирования, чем безопасности. Для электронной торговли система безопасности должна быть разработана на основе синтеза всех рассмотренных выше требований: подтверждения подлинности, секретности, целостности, безотказности, аудита. Потребности достаточно просты: нам нужна возможность перемещать денежные массы по компьютерной сети. При пристальном рассмотрении обнаруживаются несколько путей для достижения этого. Мы можем взять любой из имеющихся вариантов оплаты: наличные деньги, чеки, дебеторские и кредиторские карты, кредитные бумаги и перенести их в киберпро¬странство. Различные платежные средства подчиняются различным правилам и требованиям.

Некоторые требования зависят от того, какую ответственность кто несет. Торговцы и компании, обслуживающие операции по кредитным картам, несут ответственность по большинству долгов украденных кредитных карт и мошенническим

сделкам с их использованием. По этой причине электронные версии для данных систем разрабатываются таким образом, чтобы облегчить жизнь именно им, а не потребителям.

Различные физические реализации также предполагают различные требования. Эта система сетевая или автономная? Все намного проще, если вы можете рассчитывать на сетевое соединение с банком (каковое требует банкомат). Если вы создаете торговую систему для использования в той части света, где недостаточно телефонных линий (как, например, в отдельных районах Африки), вы не можете принять этот вариант. Будет ли система работать в программной среде или мы можем рассчитывать на надежные аппаратные средства, подобные смарт-карте? И будет ли эта система предполагать анонимность, как в случае использования наличных денег, или включать опознавание, подобно системе кредитных карт? И наконец, какое правительственное регулирование будет осуществляться по отношению к этой системе? Это зависит не только от выбранных платежных средств, но также от постановлений правительства или правительств, имеющих власть над системой.

Мы уже можем наблюдать кое-что из этого. У нас пока нет цифровых наличных денег, но уже появляются альтернативные системы, которые выполняют ту же роль, что и деньги. Flooz.com создал специализированную валюту для оплаты подарков. На нем выдаются подарочные сертификаты, которые могут быть использованы в качестве денег. Beenz.com предпринимает нечто подобное; «beenz» не являются настоящими деньгами, но они могут использоваться и обращаться, как настоящие деньги. Другие компании тоже участвуют в этом процессе.

Я ожидаю, что это станет большим делом и, возможно, опасным. Причина в том, что псевдовалюты не могут играть той регулирующей роли в процессе товарооборота, которую играют реальные деньги.

## Упреждающие меры

Традиционно предотвращение мошенничества было упреждающим. Криминальные элементы находят изъян в торговой системе и пользуются им. Они продолжают идти вперед, в то время как проектировщики системы пытаются понять, как устранить недостатки или хотя бы свести к минимуму ущерб. Преступники изучают ситуации, когда их атаки не достигают цели, и продолжают атаковать другими способами. И процесс продолжается.

Вы можете проследить это на примере кредитных карт. Изначально подтверждение кредитных карт не осуществлялось через сеть. Торговцам предоставляли книги с недействительными номерами кредитных карт каждую неделю, и они должны были вручную проверять номер по книге. Сейчас подтверждение карты происходит по сети в режиме реального времени. Плохие люди воровали новые карты из почтовых ящиков; из-за этого компании, обслуживающие кредитные карты, стали требовать, чтобы вы звонили для активации своей карты. Сейчас карты и извещения об активации отправляются из различных точек. У компаний также есть разведывательные программы для контроля непредвиденных расходов. («Доброе утро, сэр, извините за беспокойство. Многие годы вы были хорошим клиентом.

**Мы хотим** проверить, действительно ли вы внезапно переехали в Гонконг и полностью исчерпали свой кредит».)

Когда банкоматы впервые были введены Citicorp в 1971 году, клиент должен был помещать кредитную карту в прорезь и набирать свой идентификационный номер[1]. Машина проверяла его и выбрасывала карту обратно клиенту. После этого он мог закончить операцию. Предприимчивые нью-йоркские преступники переодевались в костюмы обслуживающего персонала и ждали недалеко от этой машины. После подтверждения идентификационного номера клиента они подходили и говорили, что банкомат сломан, проходит тестирование или в нем просто нет денег, и просили использовать соседний рядом. В конце концов, людям в таких костюмах можно доверять — так думали клиенты. После того как клиент уходил, они заканчивали первую операцию и клали в карман наличные деньги.

Карта должна была удерживаться до конца сделки, но это требовало реконструкции аппаратных средств. Банкам нужно было действовать быстро, и они нашли временное решение, которое могло быть введено в действие в банкоматах: было сделано так, чтобы расположенные рядом машины имели связь между собой. Поскольку банки применили это повсюду, то могли наблюдать, как преступники перемещались по всему городу в поисках машин, где уловка все еще срабатывала. Тогда они настроили банкомат так, чтобы он удерживал карту до конца сделки. Долгосрочное решение состояло в том, чтобы создать сеть с обратной связью, дающую уверенность в том, что в любой момент времени проводится только одна транзакция с использованием данной карты. Это было сделано, так что теперь не имеет значения, сколько времени карта удерживается машиной. Теперь многие банкоматы попросят вас просто предъявить свою карту, но раньше было очень много мошенничества, пока проблема не была определена.

Подобные способы фиксации недостатков в системах безопасности после того, как уже было осуществлено нападение, не подходят для Интернета. Атаки могут быть автоматизированными, они могут легко и быстро повторяться низкоквалифицированными нападающими. Нападение на банкоматы, адаптированное к Интернету, может разрушить банковскую систему. Недостаточно противодействовать мошенничеству после того, как оно было продемонстрировано в работе; мы должны быть предусмотрительны и бороться с обманом до того, как он произойдет.

[1] Первый банкомат в Нью-Йорке и в США (компания Docutel) был установлен банком Chemical Bank в Лонг-Айленде в 1969 году. В Лондоне это произошло двумя годами раньше (первый в мире). Речь идет о конкретной модели банкоматов. — *Примеч. ред.*

# Часть II

# Технологии

Система безопасности, как луковица, состоит из слоев. На внешнем слое находятся пользователи, по-разному использующие систему, по-разному всем доверяющие и по-разному же реагирующие на баги системы. Внутри «луковицы» находятся связи, обеспечивающие безопасность взаимодействия пользователя с системой и контактов различных систем. Еще ближе к сердцевине расположены программные средства, наверняка содержащие ошибки; поэтому естественно ожидать, что для них у нас есть какие-то элементы защиты. Эти программы работают в сетях и на отдельных компьютерах. Двигаясь глубже, мы обнаружим теоретически идеальные протоколы обмена данными. И в самой сердцевине (иногда) располагается криптография: математические уравнения, описывающие условия безопасности.

Защита — это процесс, а не продукт. Он включает в себя большое количество компонентов. Как и в любом процессе, одни из них — более сильные, надежные, гибкие и безопасные, чем остальные. Кроме того, компоненты должны работать совместно. Чем лучше они совместимы, тем лучше идет весь процесс. Часто наименьшей надежностью обладают именно связи между компонентами.

Защита также похожа на цепь. Она состоит из многих звеньев, и для прочности цепи важно каждое из них. И, подобно цепи, надежность всей системы безопасности определяется надежностью самого слабого ее звена. В этой части книги мы коснемся различных технологий защиты, из которых состоит эта цепь, постепенно продвигаясь от сердцевины «луковицы» к внешним слоям.

И мы постараемся не злоупотреблять смешением плохо согласующихся между собой метафор.

# Глава 6. Криптография

Криптография весьма загадочна. С одной стороны — это набор сложных математических выражений. Шифровальщики вечно изобретают сложные математические преобразования, а им вечно противостоят криптоаналитики, находя все более оригинальные способы нарушить работу этой математики. У шифрования длинная и славная история: с его помощью наперсники, любовники, тайные общества и правительства на века сохраняли свои секреты.

С другой стороны, криптография — это одна из основных технологий в кибер¬пространстве. Она позволяет нам взять все те деловые и социальные структуры, с которыми мы привыкли иметь дело в физическом мире, и переместить их в ки¬берпространство. Криптография — это технология, позволяющая нам обеспечить безопасность в киберпространстве и бороться с теми атаками и злоумышленниками, о которых шла речь в части I. Без криптографии никогда бы не смогла распространиться электронная торговля. Криптография не являет собой панацею, для полной надежности вам потребуется еще много всего другого, но она, несомненно, важна.

Для того чтобы понять, как обеспечивается безопасность в киберпространстве, вы должны уяснить, как устроена криптография. Вам не обязательно разбираться в математике, но придется освоить некоторые приемы ее применения. Следует иметь представление о том, что может криптография и, что еще важнее, чего она не может. Вы должны уметь рассматривать ее в контексте компьютерной и сетевой безопасности. Две следующие главы не превратят вас в шифровальщиков, а только научат грамотно пользоваться криптографией.

С точки зрения пользователя, криптография представляет собой туманный объект, выполняющий функции защиты, — вроде Бэтмана — нечто грозное, но справедливое и наделенное мистической силой. Если пользователь уделит ей немного внимания, криптография предстанет перед ним целым собранием акронимов, которые способны обеспечить решение самых различных задач безопасности. Например, IPsec защищает трафик в Интернете. С его помощью обеспечивается безопасность виртуальных частных сетей (VPN). Протокол SSL (Secure Sockets Layer) отвечает за безопасность соединений Всемирной Сети. Системы PGP (Pretty Good Privacy) и S/MIME гарантируют надежность электронной почты; они не дают прочитать сообщения никому, кроме адресата, и не допускают фальсификации авторства. Протокол SET защищает операции, проводящиеся в Интернете с использованием кредитных карт. Все вышеперечисленное — это протоколы. Существуют протоколы для защиты цифровой информации (музыки, фильмов и т. п.), для аутентификации сотовых телефонов (чтобы предотвратить мошенничество), для электронной торговли и для многого другого. Чтобы создать эти протоколы, шиф¬

ровальщики используют различные алгоритмы: алгоритмы шифрования, цифровой подписи и т. д.

## Симметричное шифрование

Исторически криптография использовалась с одной единственной целью: сохранить секрет. Даже сама письменность была своего рода шифрованием (в Древнем Китае только высшие слои общества могли обучаться чтению и письму), а первый опыт применения криптографии в Египте относится примерно к 1900 году до н. э.: автор надписи пользовался необычными иероглифами. Есть и другие примеры: дощечки из Месопотамии, на которых зашифрована формула изготовления керамической глазури (1500 год до н. э.), еврейский шифр ATBASH (500-600 годы до н. э.), греческое «небесное письмо» (486 год до н. э.) и шифр простой подстановки Юлия Цезаря (50-60 год до н. э.). Кама Сутра Ватсьяяны даже помещает искусство тайнописи на 44-е, а искусство секретного разговора на 45-е место в списке 64 искусств (йог), которыми должны владеть мужчины и женщины.

Основная идея, лежащая в основе криптографии, такова: группа людей может секретным способом записывать послания так, что они будут непонятны всем остальным. Пусть имеются сообщения (их еще называют открытым текстом), которые кто-то хочет сохранить в секрете. Представим, что кто-то (назовем ее Алиса) хочет послать сообщение кому-то другому (например, Бобу); а может, она хочет сама перечитать его через несколько дней. Но она точно не хочет, чтобы кто-нибудь другой, кроме Боба, смог этот текст прочесть.

Поэтому Алиса зашифровывает сообщение. Она придумывает какие-нибудь преобразования (их называют алгоритмом), превращающие открытый текст в зашифрованный. Такое зашифрованное сообщение кажется абсолютной абракадаброй, поэтому перехватчица (назовем ее Евой), в чьи руки оно попало, не может превратить его опять в открытый текст, а значит, не сумеет понять смысла сообщения. А Боб знает, как произвести обратное преобразование и превратить шифрограмму в открытый текст.

Такая схема более или менее работает. Алиса может при помощи изобретенного ею алгоритма зашифровать свой секрет глазури для керамики. Алиса и Боб могут договориться об алгоритме, чтобы поделиться друг с другом мыслями о Кама Сутре. Целый общественный класс — китайская знать (хотя, скорее всего, никого из них Бобом не звали) — использовал письменность, чтобы утаить от крестьян государственные тайны.

Но возникают определенные сложности. Во-первых, алгоритм должен быть надежным. Не надейтесь, что Ева посмотрит на зашифрованное сообщение, пожмет плечами и отступится. Она твердо намерена прочесть открытый текст. Если Ева — это правительство Великобритании времен Второй мировой войны, она наймет лучших математиков, лингвистов и шахматистов страны, запрет их и еще 10 000 человек в тайной резиденции в Блетчли Парк и создаст компьютер — только так она сможет взломать алгоритм и восстановить текст сообщения. Агентство национальной безопасности — единственный в своем роде крупнейший потребитель компьютерного оборудования и работодатель математиков. Алисе нужно быть очень

ловким шифровальщиком, если ей предстоит перехитрить Еву такого уровня. Я еще расскажу об этом позже.

Во-вторых, сложно включать и исключать людей из группы избранных, которым известен алгоритм. Чтобы обмениваться тайными посланиями с китайским аристократом, вам придется научиться китайской грамоте. Это потребует времени. Если через какое-то время вы утратите расположение правительства, у последнего будет только один способ помешать вам читать все сообщения. Вы знаете, как осуществляется шифрование, и если правительство не хочет, чтобы вы читали его переписку и дальше, ему придется вас убить. (Во время Второй мировой войны американская армия использовала в качестве шифра язык навахо. Те, кто говорил на этом языке, сохранили секрет от японцев, но вся система могла бы рухнуть, если бы хоть один навахо перешел на вражескую сторону.)

Останься эти две проблемы неразрешенными, в наше время криптография могла бы стать почти бесполезной. Представьте, что вы, один из скольких-то там миллионов пользователей Интернета, хотите посекретничать с сотней своих лучших друзей. Вы хотите пользоваться не одним и тем же тайным языком для всех ста друзей, а использовать сто разных секретных алгоритмов. (Вам необходима попарная секретность. И того же желают остальные несколько миллионов пользователей Интернета.) Это означает, что вам придется создать эти 100 различных алгоритмов шифрования, обменяться ими с каждым из друзей, самостоятельно запрограммировать все алгоритмы в своем компьютере (вы ведь никому этого не доверите) и надеяться на то, что вы умнее тех, кто, возможно, попытается понять ваш алгоритм.

Очень маловероятно.

В этом прелесть ключа. Замок на вашей двери — это серийный продукт какой-то безликой компании, которую нисколько не заботит ценность вашей коллекции марочных вин, но вы и не должны им доверять. Они не говорят вам: «Помните, любой, у кого есть замок такой же марки, может открыть ваш замок». У вас есть ключ. Набор шпеньков в замке, соответствующий вашему ключу, отличает ваш замок от всех остальных замков в округе, несмотря на то что у них может быть та же модель. (На самом деле пример упрощен. Вы все же должны принять на веру, что они правильно собрали замок и не припасли запасных ключей. Но мы пока что не принимаем это во внимание.)

Дверной замок служит примером такой же модели защиты, какую в 1466 году привнес в криптографию Леон Баттиста Альберти, известный итальянский архитектор эпохи Возрождения, создав криптографический ключ. У кого угодно могут быть замки одинаковых моделей, но ключи у всех разные. Конструкция замка не является уникальной — у слесарей есть книги с подробными схемами, а большинство хороших моделей описаны в общедоступных патентах, — но ключ является тайной. У вас есть ключ, значит, вы можете отпереть дверь. Если вы одолжите ключ другу, то он сможет войти в дом. А тот, у кого ключа нет, останется на улице. (Слесарями в данном случае будут криптоаналитики; до них мы доберемся позднее.)

Применение такой модели в криптографии решает обе упомянутые выше проблемы. Алгоритмы, так же как и замки, можно стандартизировать. Стандарт шифрования данных (Data Encryption Standard, DES) — это общепринятый криптографический алгоритм, широко распространенный с 1977 года. Его использовали

в тысячах различных программ для любых приложений. Самые сокровенные подробности устройства DES были опубликованы с первого дня его существования; их опубликовали даже до того, как этот алгоритм приняли в качестве стандарта. Доступность этого алгоритма не сказывается на безопасности, поскольку различные группы пользователей выбирают себе разные секретные ключи. Если Алиса и Боб пользуются одним ключом, значит, они могут общаться. Ева ключа не знает, следовательно, она не сможет прочесть их сообщения — даже если у нее есть точно такие же шифровальные программы, как у Алисы и Боба.

С помощью ключей решается задача включения и исключения людей из группы избранных. Если Алиса и Боб договорились о совместном использовании одного ключа и хотят, чтобы Ким Филби смог присоединиться к их общению, они просто дадут ему копию ключа. Если позже они решат, что Филби передает секреты Советскому Союзу, им нужно просто сменить ключ и не сообщать об этом Филби. С этого момента он исключен из системы и больше не сможет читать сообщения, зашифрованные новым ключом. (Увы, он все же может прочитать старые.)

Современная криптография традиционно устроена именно таким образом. Алгоритмы заменяют традиционные ручку и бумагу — они оперируют битами вместо символов алфавита, компьютеры оснащены эффективными микропроцессорами и интегральными схемами, — но философия остается прежней. Алгоритм доступен для всех, а общающиеся стороны договариваются о секретном ключе, который они применят для этого алгоритма.

Такие алгоритмы называют *симметричными,* потому что отправитель и получатель используют один и тот же ключ. Ключ представляет собой случайную строку битов некоторой длины: в 2000 году хорошей длиной ключа считалась длина в 128 бит. У различных симметричных алгоритмов — разная длина ключей.

Симметричные алгоритмы можно обнаружить в шифровальных системах всего компьютеризованного мира. Общепринятыми алгоритмами считают DES и тройной DES, RC4 и RC5, IDEA и Blowfish. Улучшенный стандарт шифрования (Advanced Encryption Standard, AES) вскоре станет стандартным алгоритмом шифрования правительства США[1]. При помощи этих алгоритмов обеспечивается безопасность частных сообщений электронной почты, индивидуальных файлов, электронных банковских операций и кодов запуска ракет с ядерными боеголовками. Эти алгоритмы препятствуют нарушению конфиденциальности.

---

[1] 40- и 48-битовые ключи RC5 были вскрыты (за 3,5 и 316 часов) на конкурсе, объявленном Bell Labs, в 1997 году, на сети из 250 компьютеров университета Беркли. (40 бит — экспортное ограничение США на ключи шифрования.) Через полгода пал DES (56-битовый ключ вскрывался 140 дней). В 1998 году он уже был вскрыт за 39 дней. Именно тогда национальный институт стандартизации США (NIST) объявил конкурс на утверждение нового стандарта AES (Advanced Encryption Standard), призванного защищать конфиденциальную, но не секретную информацию. На конкурс было подано 15 заявок, первый этап отбора прошли только 5 (включая RC6). В числе финалистов был и TwoFish — разработанный компанией Шнайера Counterpane Labs (впитавший много из Blowfish). (Бизнес Шнайера после этого резко пошел в гору, была основана новая компания, о TwoFish много писали в прессе.) Победителем стал шифр Rijndael (быстрый блочный шифр, реализованный на математическом аппарате теории конечных полей). В 2001 году Rijndael был принят в качестве американского стандарта криптографической защиты AES и заменил отживший свое DES — обратите внимание при чтении следующей главы в разделе «Выбор алгоритма или протокола». Правда, из заявленного множества длин ключей (128, 192, 256) остался только 128-битовый. — *Примеч. ред.*

Но они несовершенны.

Проблема в распределении ключей. Для того чтобы описанная система работала, Алиса и Боб должны договориться о секретном ключе перед тем, как получат возможность обменяться тайными сообщениями. Если они достаточно сообразительны, то будут регулярно менять ключ, скажем, раз в день. Им необходимо както тайно уславливаться об этих ежедневно изменяющихся ключах, так как, перехватив ключ, кто угодно сможет читать все сообщения, зашифрованные с его помощью. Кроме того, поскольку необходима попарная секретность, количество ключей будет возрастать пропорционально квадрату количества пользователей. Двум пользователям нужен только один ключ, но сеть из десяти пользователей требует 45 ключей[1], чтобы предоставить каждой паре возможность секретного общения. А сеть из 100 пользователей потребует 4950 различных ключей. В 80-х годах корабли Военно-Морского флота США отправлялись в плавание с полным набором ключей, предоставляемых Агентством национальной безопасности — каждый ключ был напечатан на бумажной ленте, перфокарте или на чем-нибудь еще; этого было достаточно для всех сеансов связи на все время выполнения задания.

Но недостаточно только распространять ключи секретно: их нужно секретно хранить, секретно использовать, а затем секретно уничтожать. Алиса и Боб обязаны хранить свои ключи в тайне в течение всего времени, пока им необходимо общаться друг с другом и быть уверенными, что ни у кого больше нет их ключей. Секретность должна соблюдаться как до и во время использования ключа, так и впоследствии.

Это означает, что уничтожение ключей имеет большое значение. Алиса и Боб не могут просто выбросить свой ключ в корзину в конце работы и надеяться, что его никто не найдет. Перехватчики не пренебрегают хранением зашифрованных посланий, которых не способны прочесть, в надежде, что найдут к ним ключ через несколько дней. Агентство национальной безопасности получило возможность расшифровать русский трафик VENONA (вспомните эту историю; это действительно поучительно) только благодаря тому, что СССР повторно использовал ключ, который, должно быть, ранее выбросил, и тому, что АНБ хранило все эти советские шифрованные сообщения более 10 лет[2].

Есть множество исторических примеров, когда недостаточное внимание к обращению с ключами приводило к разгадке безупречного шифрования. Джон Уокер служил в военно-морском флоте США и был офицером безопасности, ответственным за обеспечение безопасности ключей, но одновременно он делал и другую карьеру, фотографируя ключи флота и отсылая фотографии русским. Японская сек¬

[1] В случае попарных комбинаторных сочетаний их количество подчиняется зависимости, близкой к квадратичной. — *Примеч. ред.*

[2] Venona, тайная операция спецслужб США периода Второй мировой войны, получила свое название по имени матери Гайаваты из поэмы Г. Лонгфелло. В результате многократного использования одноразового шифра при обмене донесениями между Центром и резидентурой была расшифрована секретная переписка советской разведки, что привело к разоблачению и аресту множества агентов, работавших в 1939-1957 годах. Официально операция завершена в 1980 году. Американский исследователь истории криптографии Роберт Луис Бенсон, офицер АНБ, в 1996 году подготовил двухтомный сборник, названный «VENONA. Soviet Espionage and American Responce», куда вошли и материалы по Venona. Их можно посмотреть на сайте АНБ (http://www.nsa.gov/docs/venona). — *Примеч. ред.*

та поклонников смерти Аум Синрике зашифровала свои компьютерные записи, но она была настолько неосторожна, что оставили копии ключей на дискете, обнаруженной полицией. Это произошло в 1995 году: не кажется ли вам, что приверженцы культа смерти могли бы к тому времени кое-что знать о ключах.

## Типы криптографических атак

Что означает «взломать» алгоритм? На первый взгляд очевидно — кто-то сможет прочитать сообщение, не имея ключа. Но на самом деле все сложнее.

Если злоумышленник просто берет зашифрованное сообщение и восстанавливает открытый текст, то такую атаку называют *атакой с использованием только шифрованного текста (ciphertext-only attack)*. Их, как правило, больше не используют — современные алгоритмы слишком хороши, чтобы их одолеть атаками такого рода.

С большей вероятностью можно встретить *атаки с использованием известного открытого текста (known-playntext attack):* у дешифровщика есть образец открытого и шифрованного текста, что позволяет ему восстановить ключ. На первый взгляд может показаться, что это бесполезно, но на самом деле смысл есть. Если при помощи этого же ключа будут зашифрованы другие тексты, злоумышленник сумеет взять ключ и прочесть остальные тексты. Например, почти у всех компьютерных файлов есть заголовок. Все файлы Microsoft Word, например, начинаются с нескольких сотен одинаковых байтов. (Это «невидимые» символы — служебные байты программы, и они не отображаются в окне.) Если криптоаналитик может, воспользовавшись имеющимся открытым текстом, воспроизвести ключ, то сумеет прочитать документ Word полностью. Атаки, при которых известен открытый текст, с большим успехом применяли против немецкой «Энигмы». Аналитики получали в руки единственный незашифрованный текст, например прогноз погоды, а через некоторое время один из отдаленных немецких постов в Норвегии начинал пунктуально присылать ежедневные одинаковые сообщения: «Нечего сообщать». (Открытый текст можно еще называть *шпаргалкой.)* Обычно таким образом узнавали ключ текущего дня, а затем с помощью ключа читали зашифрованные сообщения.

Еще более действенной является *атака с помощью избранного открытого текста (chosen-plaintext attack).* При такой атаке криптоаналитик имеет возможность выбрать сообщение, которое после этого зашифровывают. Затем он получает зашифрованное сообщение и восстанавливает ключ. Атаки этого рода срабатывали против немецких шифров: союзники сознательно допускали утечку определенной информации для того, чтобы получить зашифрованный текст, или провоцировали сообщения о событиях в городах с уникальными названиями, служащие особенно хорошими шпаргалками. Эти атаки хорошо срабатывают и против некоторых систем, использующих смарт-карты, в результате чего злоумышленники получают возможность помещать на карту нужную им информацию.

Все эти атаки объединяет то, что аналитикам известны детали алгоритма. (Единственным исключением, которое мне известно, сегодня является японский код PURPLE.) Это не просто ученый ярлык, а очень хороший механизм. Если в программах будут использовать какой-то алгоритм, то его «раскрутят» и в обратную

сторону. Среди секретных когда-то алгоритмов, которые уже «раскрутили» — RC4, все алгоритмы шифрования цифровой сотовой связи, DVD- и DIVX-алгоритмы шифрования видеоизображений и алгоритм Firewire. Захватят и расшифруют даже алгоритмы, глубоко запрятанные в военном оборудовании. Примерами могут служить «Энигма» во Второй мировой войне[1] или почти все алгоритмы НАТО и Варшавского договора во времена холодной войны. (Мы их не знаем, но весьма компетентные военные занимались их расшифровкой.) Полезно предполагать, что врагу известны подробности вашего алгоритма, потому что в конечном итоге они все равно станут ему известны. Август Керхкофф первым сформулировал это положение в 1883 году: «В алгоритме нет никакой тайны, вся тайна в ключе».

# Распознавание открытого текста

В разговоре об атаках всегда возникает один вопрос: как криптоаналитик распознает открытый текст? Ответ прост: его легко узнать, потому что он выглядит как открытый текст. Это сообщение на английском языке или файл компьютерного приложения, изображение в формате JPEG или база данных в каком-нибудь приемлемом формате. Когда вы смотрите на расшифрованный файл, он похож на что-нибудь вам известное. Когда вы смотрите на зашифрованный файл или файл, расшифрованный с применением неправильного ключа, он выглядит как полная тарабарщина. Человек или компьютер могут понимать эту разницу.

В 1940-х годах Клод Шеннон ввел понятие *расстояния уникальности (unicity distance)*. Среди прочего, расстояние уникальности измеряет количество необходимого зашифрованного текста, позволяющее однозначно воспроизвести открытый текст. Это значение зависит и от свойств открытого текста, и от длины ключа, характерной для такого алгоритма шифрования.

Например, алгоритм RC4 зашифровывает данные в байтах. Представьте себе одну единственную букву в ASCII-кодировке в качестве открытого текста. На 26 букв приходится 256 возможных вариантов кодирования. Любой случайный ключ, если использовать его для расшифровки этого текста (буквы), с вероятностью 26/256 даст верный открытый текст. У аналитика нет никакого средства, позволяющего отличить ошибочный открытый текст от правильного.

Представьте теперь сообщение электронной почты размером 1 Кбайт. Аналитик пытается применять случайные ключи, и в конечном счете возникает открытый текст, который выглядит как сообщение электронной почты: слова, фразы,

[1] «Энигма» — одна из первых роторных машин, осуществляющая шифрование (многоалфавитную подстановку) посредством взаимодействия вращающихся роторов. Разработана в 1917 году Эдвардом Хеберном и усовершенствована Артуром Кирхом. Роторные машины активно использовались во время Второй мировой войны. Для того времени это было последнее слово докомпьютерной криптографии. До появления ЭВМ шифры роторных машин считались наиболее стойкими. После Второй мировой войны США продавало немецкую «Энигму» в страны Третьего мира. Факт, что шифр уже взломан, долгие годы при этом оставался засекреченным. До недавнего времени шифр «Энигмы» использовался отдельными UNIX-системами для шифрования файлов. Алгоритмы «Энигмы» были опубликованы в 60-х годах, как и связанные с ними результаты по решению уравнений в подстановках. Японское устройство Purple (пурпурный, багровый, царский) также является роторной машиной. — *Примеч. ред.*

предложения, грамматика. Вероятность того, что это неправильный открытый текст, бесконечно мала.

Для стандартного англоязычного сообщения расстояние уникальности равно K/6,8, где K — это длина ключа в битах. (6,8 — степень естественной избыточности английского языка. Для других открытых текстов она будет больше или меньше, но незначительно.) Для ASCII-кода, применяемого согласно стандарту DES, расстояние уникальности составляет 8,2 байт. Для 128-битового шифра это примерно 19 байт. Таким образом, для англоязычных сообщений, длина которых превышает 19 байт, расшифрованный текст, похожий на английский, с большой вероятностью будет истинным открытым текстом. Почти такое же значение расстояния единственности имеют файлы электронных таблиц, текстовых процессоров и баз данных. (На самом деле оно может быть намного меньше, потому что форматы файлов предполагают стандартное начало файла.) Для сжатых файлов расстояние уникальности могло бы быть в два-три раза больше (но опять-таки, стандартное начало может его существенно снизить).

Отсюда мораль: «Распознать открытый текст просто, и для этого не требуется большого количества информации».

## Коды аутентификации сообщений

Коды аутентификации сообщений (Message authentication codes или MACs) — это следующий базисный элемент, о котором мы поговорим. Они не обеспечивают секретность, но гарантируют аутентификацию и целостность. Они дают уверенность, что сообщение пришло именно от того человека, который обозначен как автор (это аутентификация), и что сообщение по пути не изменилось (а это целостность).

Вы можете рассматривать MAC как защищающую от вскрытия оболочку сообщения. Кто угодно может прочесть сообщение — оболочка не обеспечивает секретность. Но кто-то, кто знает ключ MAC, может удостовериться, что сообщение не было изменено. Конкретнее, MAC — это номер, который прикреплен к цифровому сообщению.

Для MAC применяют секретные ключи совместного использования, типа симметричных алгоритмов шифрования. Сначала Алиса договаривается о ключе с Бобом. Затем, когда она хочет послать Бобу сообщение, она вычисляет MAC сообщения (применяя секретный ключ) и присваивает его сообщению. У каждого сообщения есть уникальный MAC для любого возможного ключа.

Когда Боб получает сообщение, он вычисляет его MAC (опять-таки используя все тот же совместный ключ) и сравнивает его с тем значением MAC, которое прислала Алиса. Если они совпадают, то он может быть уверен в двух вещах: сообщение действительно пришло от Алисы (или от кого-то, кто знает секрет общего ключа) — потому что только применяя этот ключ, можно вычислить MAC, и это сообщение цельное и не измененное — так как MAC можно вычислить только по полному и точному сообщению. Если бы Ева (помните нашу перехватчицу?) прослушивала связь, она смогла бы прочитать сообщение. Однако если бы она попыталась изменить текст сообщения или MAC, то вычисленный Бобом MAC не был бы равен тому значению, которое он получил. Еве пришлось бы изменить сообще¬

ние, а затем изменить MAC, чтобы он был правильным для нового сообщения, но она не могла бы этого сделать, так как не знает ключа. Банки используют такую простую систему аутентификации уже несколько десятилетий.

Алиса может прибегнуть к той же уловке, чтобы установить подлинность информации, содержащейся в базе данных. Добавляя информацию в базу данных, она вычисляет MAC и хранит его вместе с информацией. Когда она извлекает информацию, то снова вычисляет MAC и сравнивает его с тем значением, которое хранилось в базе данных. Если они совпадают, то она приобретает уверенность, что никто не изменил информацию.

MAC постоянно используются в Интернете. Их применяют, например, в протоколе IPsec, чтобы гарантировать, что IP-пакеты не были изменены в промежутке между отправлением и прибытием на место назначения. Их используют во всевозможных протоколах межбанковских переводов для установления подлинности сообщений. Большинство MAC сконструированы с применением симметричных алгоритмов или односторонних хэш-функций. Например, в CBC-MAC применяется симметричный алгоритм, а в HMAC и NMAC — хэш-функции.

## Односторонние хэш-функции

Односторонние (однонаправленные) хэш-функции напоминают цифровые отпечатки пальцев: небольшие фрагменты данных, которые могут служить для идентификации достаточно больших цифровых объектов. Это общедоступные функции, у них нет никаких секретных ключей.

Они названы односторонними из-за своей математической природы. Любой может вычислить одностороннее хэш-значение чего угодно (например, текста этой книги). Однако если имеется хэш-значение этой книги, исходя из вычислений невозможно создать другую книгу с таким же значением хэш-функции или получить подлинный текст книги.

Хэш-функция также может обеспечивать аутентификацию и целостность. Если бы вы загрузили эту книгу из Интернета, у вас не было бы никакого способа узнать, написал все это я или кто-то другой все же частично изменил мои слова. Однако, если бы я дал вам в руки хэш-значение для этой книги (типичный 20-байтовый код), вы смогли бы сравнить расчетный результат с тем значением, которое дал я. Если они совпадают, то это моя книга, без изменений.

Хэш-функции широко применяются в криптографии и компьютерной безопасности. Они используются почти во всех протоколах Интернета, чтобы обрабатывать ключи, связывать последовательность событий или аутентифицировать события. Они также важны для алгоритмов цифровой подписи (подробнее об этом — позднее). Они, возможно, — наиболее полезный инструмент в коллекции шифровальщика.

В настоящее время используется целый набор односторонних хэш-функций. Стандарт на хэш-функцию SHA-1 принят правительством США. Для алгоритма безопасности хэширования (Secure Hash Algorithm) есть акронимы, и они приведены в соответствующем стандарте (Secure Hash Standard, SHS). RIPEMD-160 — это европейский алгоритм. MD4 выходит из употребления (хотя вы все еще може¬

те его неожиданно встретить), а MD5 демонстрирует существенные недостатки, и его больше не используют для создания чего-либо нового.

## Шифрование открытым ключом

Помните проблему распределения ключей, о которой я упоминал в разговоре о симметричном шифровании? Как два человека могут убедиться, что у них один и тот же ключ и что они могут пользоваться алгоритмом симметричного шифрования или функцией MAC? *Шифрование открытым ключом* (или *асимметричное шифрование)* решает эту проблему. Оно позволяет вам посылать секретное сообщение людям, которых вы никогда раньше не встречали и с которыми вы не договаривались о секретном ключе. Оно допускает возможность двум людям обмениваться данными у всех на виду и в результате этого обмена получить секретные данные, которые не сможет получить кто-то, подслушивавший переговоры. Говоря в терминах физического мира, такое шифрование позволяет вам и вашему приятелю прокричать друг другу числа в кафе, битком набитом математиками, — так что, когда вы закончите, вы и ваш приятель получите одно и то же число, и никто, кроме вас двоих, совсем ничего не поймет.

Звучит нелепо? Это кажется невозможным. Если бы вы спросили шифровальщиков со всего света в 1975 году, все они сказали бы, что это невозможно. Так что можете себе представить всеобщее изумление, когда в 1976 году Витфилд Диффи и Мартин Хеллман объяснили, как это сделать. Или удивление британской разведки, когда Джеймс Эллис, Клиффорд Кок и М. Д. Уильямсон осуществили то же самое на несколько лет раньше.

Основная идея в том, чтобы использовать математическую функцию, которую просто вычислять в одном направлении и тяжело — в другом. Одна из таких функций — разложение целых чисел на множители. Если даны два числа, их легко перемножить и найти произведение. Но если дано только произведение, практически невозможно разложить число на множители и определить исходные числа. Как раз такого плана математику можно применять для создания шифрования с открытым ключом: в нее входят арифметические операции над абсолютными значениями чисел, возведение в степень и большие многоразрядные (до нескольких тысяч битов) исходные числа. Сегодня существует добрые полдюжины алгоритмов с названиями вроде RSA, Эль-Гамаль и алгоритм эллиптических кривых. (Алгоритмы, в основе которых лежит так называемая «задача о ранце», конкурировали с ними на ранних стадиях, но по прошествии 20 лет их так или иначе взломали.) Математика для каждого алгоритма своя, но концептуально они все одинаковы.

Вместо единственного ключа совместного пользования у Алисы и Боба есть два ключа: один для шифрования, а другой для расшифровки. Ключи различны, и невозможно, зная один ключ, вычислить другой. То есть если у вас есть ключ для шифрования, вы не сумеете найти ключ для расшифровки.

Вот в этом-то и есть самое интересное. Боб может создать пару таких ключей. Он может взять и обнародовать ключ для шифрования. Он может послать его друзьям, опубликовать на своем веб-сайте или поместить в телефонной книге. Алиса может найти этот ключ. Она может с его помощью зашифровать сообщение для

Боба. Затем она может послать ему сообщение. Боб, используя свой ключ расшифровки (который он предусмотрительно не размещал на веб-сайте), сможет расшифровать и прочитать послание Алисы. Заметим, что Алисе не приходится встречаться с Бобом в какой-нибудь темной аллее и договариваться об общем секрете. Бобу даже не обязательно знать Алису. И, как ни странно, даже Алисе не обязательно знать Боба. Если Алиса сможет найти ключ, который Боб обнародовал, она сможет послать ему тайное сообщение, которое никто, кроме Боба, не сможет прочитать. Такое постоянно происходит с пользователями PGP; один из их ключей находится на каком-либо сервере, и тогда совершенно посторонний человек может отправить им зашифрованные сообщения. Даже если вы что-то смыслите в математике, это не менее удивительно.

Детали этого процесса содержат в себе целую кучу хитростей. Например, я не рассказал, как Боб создал открытый и закрытый ключи и как он сделал свой личный ключ секретным. (Он не может его помнить — ведь ключ состоит из более чем тысячи случайных цифр.) И я пропущу здесь рассказ о невероятно сложной задаче — как Алиса узнает, что она получила именно ключ Боба, а не какой-то старый, или неправильный, или ключ какого-либо злоумышленника. Мы вернемся к этому позднее.

А сейчас я хочу обратить ваше внимание на то, что никто не применяет шифрование с открытым ключом для кодирования сообщений. Все операционные системы используют гибридные технологии, в которых задействованы оба типа криптографии. Причина интереса к этому подходу в его эффективности. На самом деле, когда Алиса хочет послать сообщение Бобу, она зашифровывает сообщение при помощи симметричного алгоритма, используя произвольный ключ, который создает «из воздуха» (так называемый *сеансовый ключ)*. Она зашифровывает этот произвольный ключ при помощи открытого ключа Боба, а затем отправляет вместе зашифрованный ключ и зашифрованное сообщение для Боба. Когда Боб их получает, он производит обратную операцию. При помощи личного ключа он расшифровывает произвольный симметричный ключ, а затем использует его для расшифровки сообщения.

Это может показаться сверхъестественным, но все совершенно нормально. Повторюсь, никто не использует криптографию с открытым ключом непосредственно для шифрования сообщений. Все применяют гибридные технологии. Так устроены все программы, обеспечивающие безопасность электронной почты, — PGP, PEM, S/MIME и любые другие. Так обеспечивается защита сообщений Веб, TCP/IP, телефонной связи и всего остального.

## Схемы цифровой подписи

Шифрование с открытым ключом — вещь довольно удивительная, но цифровые подписи (сигнатуры) — еще более интересный и важный инструмент. Цифровые подписи обеспечивают тот же уровень аутентификации сообщений, что и MAC. А в современном бизнесе аутентификация намного важнее секретности,

Как и шифрование с открытым ключом, цифровые подписи используют пару ключей: открытый и закрытый. Вы также не можете установить по одному ключу другой. Но в этом случае ключи меняются местами.

У Алисы есть открытый текст сообщения. Применяя свой закрытый ключ, она сообщение зашифровывает. Поскольку это ее личный ключ, то только им можно зашифровать сообщение абсолютно тем же способом. Таким образом, зашифрованное сообщение становится Алисиной *подписью* на сообщении. Открытый ключ Алисы общедоступен. Кто угодно способен достать этот ключ и расшифровать сообщение, удостоверившись таким образом, что его *подписала* (то есть зашифровала) Алиса. Подпись является функцией сообщения, поэтому она уникальна для сообщений: злостный фальсификатор не может снять подпись Алисы с одного документа и поместить ее на другой. Подпись — это функция личного ключа Алисы, то есть она уникальна для нее.

Конечно, реальные системы более сложны. Так же как Алиса не зашифровывает сами сообщения при помощи алгоритмов шифрования с открытым ключом (она зашифровывает только ключ сообщения), она и не подписывает непосредственно сообщение. Вместо этого она вычисляет одностороннюю хэш-функцию сообщения и затем ее подписывает. Опять же, подписывание хэш-значения на несколько порядков быстрее, и надо иметь в виду, что существует математическая проблема защиты при подписывании сообщений напрямую.

Таким образом, большинство алгоритмов цифровых подписей на самом деле не зашифровывают подписанные сообщения. Идея та же, но математическое исполнение отличается. Для того чтобы создать подпись, Алиса производит некоторые вычисления исходя из сообщения и своего личного ключа. Эта подпись прикрепляется к сообщению. Боб проделывает другие вычисления, основываясь на сообщении, подписи и открытом ключе Алисы, чтобы проверить подпись. Ева, которая не знает личного ключа Алисы, может проверить подпись, но не может подделать сообщение или полноценную подпись.

В настоящее время применяются несколько алгоритмов цифровой подписи. Наиболее популярен RSA. Алгоритм цифровой подписи американского правительства (Digital Signature Algorithm, DSA), который применяют в стандарте цифровой подписи (Digital Signature Standard, DSS), также используется часто. Вы можете иногда встретить алгоритм Эль-Гамаль. А еще существуют алгоритмы подписей, в основе которых лежит криптография эллиптических кривых; они похожи на все прочие, но в некоторых ситуациях работают эффективнее.

Хотя алгоритмы цифровой подписи с открытым ключом похожи на MAC, они лучше в одном важном нюансе. Используя MAC, Алиса и Боб применяют совместный секретный ключ для аутентификации сообщений. Если Алиса получит сообщение и проверит его, она будет знать, что сообщение пришло от Боба.

Но она не сможет доказать это правосудию. В чем можно его убедить — это в том, что письмо пришло или от Боба, или от Алисы: как-никак оба они знали ключ MAC. При помощи MAC можно убедить получателя, что письмо поступило от отправителя, но MAC нельзя использовать для убеждения третьей стороны. Цифровые подписи позволяют уверить третью сторону, решающую проблему отказа от подписи: Алиса не может отправить Бобу письмо, а позднее утверждать, что никогда его не посылала.

К несчастью, действительность такова, что все, что касается подписей, является черным или белым, как это предполагает математика. Законы о цифровых подписях существуют в законодательстве многих стран, но меня беспокоит, что они

не жизнеспособны. Цифровые подписи не являются аналогом автографа (собственноручной подписи). Я расскажу об этом подробнее в главе 15.

## Генераторы случайных чисел

Случайные числа — это простой элемент криптографии, о котором меньше всего говорят, но он важен не менее, чем остальные. Почти всем системам компьютерной безопасности, в которых применяется криптография, необходимы случайные числа — для ключей, уникальных чисел в протоколах и т. п. — и безопасность таких систем часто зависит от произвольности ее случайных чисел. Если генератор случайных чисел ненадежен, вся система выходит из строя.

В зависимости от того, с кем вы разговариваете, генерация случайных чисел выглядит или тривиальной, или невозможной. Теоретически это невозможно. Джон фон Нейман, отец вычислительной техники, сказал: «Любой, кто считает, что существуют арифметические методы получения случайных цифр, безусловно, грешит». Он имел в виду, что невозможно получить что-то случайное в полном смысле слова на выходе такого детерминированного зверя, как компьютер. Это правда, но, к счастью, кое-что сделать мы можем. От генератора случайных чисел нам необходимо не то, чтобы числа были действительно случайными, а чтобы их невозможно было предсказать и воспроизвести. Если у нас будут выполнены эти два условия, мы сможем достичь безопасности.

С другой стороны, если мы нарушаем эти два условия, безопасности нет. В 1994 году в казино Монреаля установили компьютерный генератор случайных чисел для лотерей. Один наблюдательный игрок, проводивший в казино очень много времени, заметил, что выигрышные номера были каждый день одни и те же. Он успешно сорвал три Джек-Пота подряд и получил 600 000 долларов. (Как следует позаламывав руки, поскрежетав зубами и расследовав все, казино заплатило выигрыш.)

Существует несколько обширных классов генераторов случайных чисел. В основе некоторых из них лежат физические процессы, которые можно считать довольно случайными. Агентство национальной безопасности любит использовать в своей аппаратуре для создания случайных чисел электрические шумы диодов. Другие возможности — счетчик Гейгера или приемники радиопомех. Одна система в Интернете использует цифровой фотоаппарат, направленный на несколько стробоскопов. В других системах применяется турбулентность воздуха в дисководах или момент поступления сетевых пакетов.

Некоторые генераторы случайных чисел отслеживают случайные движения пользователя. Программа может попросить пользователя набрать на клавиатуре большую строку произвольных символов; она может задействовать последовательность символов или даже время между нажатиями клавиш для создания случайных чисел. Другая программа запросто способна потребовать у пользователя туда-сюда подвигать мышью или похрюкать в микрофон.

Некоторые генераторы случайных чисел применяют эту введенную информацию без изменений. В других она служит затравкой *(начальным числом)* для математических генераторов случайных чисел. Этот прием работает лучше, если системе требуется случайных чисел больше, чем их обеспечивает ввод информации.

Какого бы происхождения ни была случайность, генератор создаст ряд случайных битов. Затем их можно использовать как криптографические ключи и для всего остального, что нужно системе.

## Длина ключей

Один из простейших критериев сравнения криптографических алгоритмов — длина ключа. Пресса любит обращать на нее внимание, поскольку ее легко считать и сравнивать. Как и в большинстве случаев, когда речь идет о безопасности, реальность более сложна. Короткий ключ плох, но длинный ключ не будет хорошим автоматически. Почему это так, я расскажу в следующей главе, а сейчас лучше разъяснить понятие длины ключа и его важность.

Начнем с самого начала. Криптографический ключ — это секретное число, которое делает криптографический алгоритм уникальным для тех, кто совместно пользуется этим ключом. Если Алиса и Боб договорились об общем ключе, то при помощи алгоритма они могут тайно пообщаться. Если Ева-перехватчица не знает ключа, ей придется исследовать и ломать алгоритм.

Одна очевидная вещь, которую можно сделать, — это перепробовать все возможные ключи. Это так называемая *атака «в лоб»,* или *лобовая атака*. Если длина ключа n битов, то существует $2^n$ всевозможных комбинаций ключей. При длине ключа в 40 бит, придется перебрать около триллиона вариантов ключей. Такая задача покажется ужасно скучной для Евы, но компьютеры неутомимы. Они лучше всех решают ужасно скучные задачи. В среднем компьютеру пришлось бы испробовать половину ключей, прежде чем он найдет правильный, то есть компьютер, который мог бы проверять миллиард ключей в секунду, потратил бы на поиски правильного 40-битового ключа примерно 18 минут. В 1998 году Electronic Frontier Foundation создала машину, которая могла атакой «в лоб» ломать DES-алгоритм. Эта машина, названная DES Deep Crack, проверяла 90 миллиардов ключей в секунду; она могла найти 56-битовый ключ DES в среднем за 4,5 дня. В 1999 году распределенный интернет-проект подбора ключей для взлома DES, distributed.net (включающий в себя Deep Crack), был способен проверить 250 миллиардов ключей в секунду.

Все схемы таких взломов «в лоб» линейны: вдвое большее количество ключей потребует вдвое больше компьютерного времени. Но сложность такого взлома зависит от длины ключей экспоненциально: добавим один бит к длине ключа, и взлом «в лоб» станет в два раза сложнее. Добавим два бита — в четыре раза. Десять битов — в тысячу раз сложнее.

Сила атак «в лоб» в том, что они работают против любого алгоритма. Поскольку эта атака не касается внутренней математики алгоритма, ей все равно, что же там внутри. Одни алгоритмы могут быть быстрее других, и поэтому атака «в лоб» проходит быстрее, но длина ключа легко это перевешивает. Достаточно просто сравнить длины ключей различных алгоритмов, чтобы выяснить, какие из них более уязвимы для лобовой атаки.

В 1996 году группа шифровальщиков (включающая и меня) исследовала различные технологии, которые можно использовать для создания дешифрующих

машин, действующих по принципу лобовой атаки, и пришла к выводу, что 90-битовый ключ сможет обеспечивать безопасность до 2016 года. Ключ Triple-DES состоит из 112 бит, а наиболее современные алгоритмы имеют по меньшей мере 128-битовые ключи. (Улучшенный стандарт шифрования (AES) правительства США поддерживает длины ключей 128,192 и 256 бит.) Даже машине, работающей в миллиард раз быстрее Deep Crack, потребовался бы миллион лет, чтобы перебрать $2^{112}$ ключей и восстановить открытый текст; еще более, чем в тысячу раз, возрастает время при переходе к 128-битовому ключу. Он будет обеспечивать безопасность в течение тысячелетия.

К этим числам нужно относиться с некоторым скептицизмом. Я — не ясновидящий и ничего не знаю о будущих достижениях компьютерных технологий. Реальная безопасность зависит от нескольких вещей: насколько ваши данные ценны, как долго их требуется хранить в тайне и т. п. Но это означает, что должны существовать «законсервированные» числа — длины ключей для симметричных алгоритмов и MAC. Хэш-функции должны иметь длину, равную удвоенной длине ключа.

Длина ключа для симметричного алгоритма открытого ключа определяется более сложным образом. Наиболее эффективная атака против RSA, например, состоит в том, чтобы разложить на множители большое число. Наиболее действенная атака против Эль-Гамаля, алгоритма Диффи-Хеллмана, DSA и других систем — вычислить нечто, называемое дискретным логарифмом. (По существу, это — та же проблема.) Алгоритмы эллиптических кривых еще более сложны.

Для алгоритмов с открытым ключом специалисты сейчас рекомендуют 1024-битовые и более ключи. Параноики используют ключи еще длиннее. Системы, для которых не слишком важна долговременная секретность, пользуются 768-битовыми ключами. (Для алгоритмов эллиптических кривых применяют разные длины ключей.)

Трудно оценивать будущие трудности разложения на множители и вычисления дискретных логарифмов, поскольку нет никакого математического доказательства, что эти задачи имеют фиксированную степень сложности. (С другой стороны, мы знаем, насколько трудно перебирать все возможные ключи.) Итак, еще раз: отнеситесь ко всем этим рекомендациям, как к мнению квалифицированных специалистов — и не более того.

# Глава 7. Криптография в контексте

Если криптография так надежна, то почему же происходят сбои в системах защиты? Почему существуют электронные кражи, мошенничество, нарушения конфиденциальности и все прочие проблемы безопасности, которые обсуждались в предыдущих главах? Почему криптография не соответствует всем без исключения требованиям безопасности? Зачем я докучаю вам такой толстой книгой?

Достаточно неожиданно, но причина этого не в плохом качестве криптографии. (Многое лежит на поверхности, но есть и проблемы, упрятанные гораздо более глубоко.) Ответ нужно искать в различии между теорией и практикой.

Криптография — это раздел математики. Математика — теоретическая наука; она логична. Хорошая математика исходит из правильной предпосылки, следует единственным путем — доказательство за доказательством — через неизведанные земли и заканчивается неопровержимым выводом. По природе своей она хорошо выглядит на бумаге.

Корни проблем безопасности следует искать в физическом мире. Материальный мир во многом нелогичен. Он неупорядочен. В нем не существует единственного пути. В нем есть теории и выводы, но для того чтобы согласиться с выводами, вам необходимо принять предпосылки, модели и взаимосвязи между теориями и реальным миром. А это непросто. Люди не играют по правилам. Они делают то, чего от них не ожидают; они не укладываются в жесткие рамки. То же самое касается технических средств: время от времени все начинает плохо работать, а то и вовсе ломается. Это же можно сказать о программном обеспечении. Оно должно быть логичным и упорядоченным, как-никак это — просто комбинация нулей и единиц, но иногда оно настолько сложно, что становится больше похожим на организм, а не на творение математики. Неважно, насколько хороша криптографическая теория: когда она используется в системе, она сталкивается с практикой.

Я часто говорю о программах, что они — лишь дань моде. Когда реклама утверждает, что используется RSA, тройной DES или любой другой модный алгоритм криптографии, это равносильно заявлению о том, что дом полностью безопасен только потому, что у него надежный дверной замок. Этого недостаточно.

## Длина ключа и безопасность

Несмотря на то что я сказал в предыдущей главе, длина ключа почти ничего не определяет в безопасности.

Внутри замка на входной двери вашего дома есть множество штырьков. Для каждого из них существуют различные возможные положения. Когда кто-то вставляет ключ в замочную скважину, все штырьки перемещаются в определенные позиции. Если положение, в которое ключ ставит штырьки, как раз то, которое необходимо, чтобы замок открылся, то он открывается. Если нет — не открывается.

У замков, наиболее часто используемых в жилых домах, пять штырьков, каждый из которых может располагаться в одном из десяти различных положений. Это означает, что существует 100 000 возможных ключей. Взломщик с огромной связкой ключей может перебрать все ключи один за другим и в конце концов попасть внутрь. Ему лучше набраться терпения, поскольку если даже он тратит на один ключ 5 секунд, ему потребуется примерно 69 часов, чтобы найти подходящий ключ (и это без перерывов на сон, еду и душ).

Однажды в вашу дверь звонит торговый агент и предлагает вам купить новый замок. У его замка семь штырьков по двенадцать положений у каждого. Агент скажет, что взломщику придется три года непрерывно перебирать ключи, прежде чем он сможет открыть дверь. Почувствуете ли вы себя в большей безопасности с таким замком?

Наверное, нет. Все равно ни один взломщик не стал бы стоять перед вашей дверью 69 часов. Он, скорее всего, откроет замок отмычкой, просверлит его, вышибет дверь, разобьет окно или просто спрячется в кустах до тех пор, пока вы не отправитесь на прогулку. Замок с большим количеством штырьков и положений не обеспечит вашему дому большую безопасность, поскольку атака, которая таким образом затрудняется — перебором еще большего количества ключей — не та атака, о которой стоит задумываться особо. До тех пор пока количества штырьков достаточно, чтобы сделать недопустимой такую атаку, вам не следует о ней беспокоиться.

То же самое справедливо для криптографических ключей. Если они достаточно длинные, то лобовые атаки просто лежат за пределами человеческих возможностей. Но здесь следует позаботиться о двух вещах. Во-первых, о качестве алгоритма шифрования, а во-вторых, о качестве ключа. Какой ключ является «достаточно длинным», зависит от обеих этих вещей.

Но в первую очередь нужно разъяснить понятие энтропии.

Энтропия — мера беспорядка или, более конкретно в контексте криптографии мера неопределенности. Чем больше неопределенность, тем больше энтропия. Например, случайно выбранный человек из обычной популяции является или мужчиной, или женщиной, в этом случае переменная «пол» составляет один бит энтропии. Если случайный человек сообщает, кто из четырех «Битлз» ему больше нравится, и все варианты равновероятны, этому соответствуют два бита энтропии. Пол члена женской олимпийской команды по бегу — это величина, у которой нет энтропии — они все женщины. Энтропия предпочтений одного из «Битлз» на собрании фан-клуба Джона Леннона существенно меньше двух битов, поскольку наиболее вероятно, что выбранный наугад человек предпочитает Джона. Чем больше определенность переменной, тем меньше энтропия.

То же самое верно для криптографических ключей. То, что алгоритм использует 128-битовый ключ, не означает, что у него 128 бит энтропии для ключа. Или точнее, лучший способ сломать данную реализацию 128-битового алгоритма шифрования может состоять не в том, чтобы перебрать все ключи. «128 бит» — это про¬

сто *мера максимального* количества работы, которая потребуется, чтобы восстановить ключ; но про минимум ничего не сказано.

Во-первых, следует позаботиться о происхождении ключа. Все вычисления, касающиеся длины ключа, я производил, предполагая, что каждый ключ имеет при создании максимальную энтропию. Другими словами, я рассчитывал, что все варианты ключа равновероятны, что генератор случайных чисел, создавший ключ, работает идеально. Это допущение не совсем верно.

Многие ключи создаются на основе паролей и ключевых фраз. Система, принимающая пароль из 10 ASCII-символов, предоставляет для него 80 бит, но ее энтропия будет значительно меньше 80 бит. Некоторые символы ASCII никогда не появляются, а пароли, которые представляют собой реальные слова (или что-то похожее на слова), гораздо вероятнее, чем произвольные строки символов. Я видел оценки энтропии для английского языка, меньшие 1,3 бит на символ; у пароля энтропия меньше, чем 4 бит на символ. Это значит, что пароль из 8 символов будет приблизительно соответствовать 32-битовому ключу, а если вы захотите 128-битовый ключ, вам нужен пароль из 98 символов (на базе английского алфавита).

Видите ли, разумный взломщик не будет перебирать все возможные пароли по порядку. Он сначала испробует наиболее вероятные, а затем проверит остальные — в порядке убывания вероятности. Он проверит тривиальные пароли (типа «пароль» или «1234»), после этого — весь английский словарь, а затем различные заглавные буквы, цифры и т. п. Это называют *словарной атакой*. Программа для взлома пароля, реализующая такую атаку, называется L0phtcrack; на 400-мегагерцовом Quad Pentium II она может протестировать зашифрованный пароль по 8-мегабай¬товому словарю распространенных паролей за считанные секунды.

Вот почему смешно, когда компании вроде Microsoft рекламируют 128-битовое шифрование, а затем берут за основу ключа пароль. (Это в высшей степени характерно для всей системы безопасности Windows NT[1].) Используемые алгоритмы могут допускать 128-битовый ключ, но энтропия пароля гораздо меньше. Фактически, качество криптографии и длина ключа не важны; причиной выхода системы из строя послужит слабый пароль. (Очевидное решение — не допустить, чтобы люди перебирали множество паролей, — не срабатывает. Я подробнее остановлюсь на этой проблеме в главе 9.)

Это очень важно. Я знаю сложные системы, секретный ключ которых защищен паролем. В основе безопасности практически любого зашифрованного продукта на жестком диске компьютера лежит ключ, запоминаемый пользователем. Почти вся система безопасности Windows NT приходит в негодность из-за того, что она построена на основе пароля, запоминаемого пользователем. Даже система PGP (Pretty Good Privacy) распадется, если пользователь выберет плохой пароль.

---

[1] Когда-то под шифрованием в MS Word 6.0 всего-навсего понимался запрет на открытие файла в самом текстовом процессоре. Текст оставался доступен для прочтения в чем угодно. В приложениях Office 95 для определения пароля нужно было знать 16 байт из файла Word или Excel. Перебор 24 вариантов давал пароль. В версии Office 97 уже требовался полный перебор (кроме Access, где шифровались не данные, а пароль. Всего-навсего с помощью «исключающего или». Парольная защита Word и Excel и ныне настолько слаба, что вскрытие документов программой-взломщиком компании AccessData по-прежнему занимает доли секунды, хотя ее автор, Эрик Томсон, вставил для замедления работы в код пустые циклы, чтобы создать впечатление сложности задачи. — *Примеч. ред.*

Не важно, какие алгоритмы и насколько длинные ключи используются; секреты, которые хранятся в памяти пользователя, беззащитны сами по себе.

Ключ, сгенерированный случайным образом, намного лучше, но проблемы остаются. Генератор случайных чисел должен создавать ключи с максимальной энтропией. Недостатки генератора случайных чисел — те же, что привели к сбоям системы шифрования в Netscape Navigator 1.1. Хотя генератор случайных чисел применяли для создания 128-битовых ключей, максимальная энтропия достигала примерно 20 бит. То есть алгоритм был не лучше, чем если бы использовался 20-битовый ключ[1].

Второй предмет заботы — это качество алгоритма шифрования. Все предыдущие расчеты предполагали, что алгоритм получал ключи при помощи вычислений и использовал их совершенным образом. Если в алгоритме есть слабые места, доступные для атаки, это существенно снижает энтропию ключей. Например, алгоритм A5/1, используемый европейской сетью сотовых телефонов GSM, имеет 64-битовый ключ, но может быть взломан за время, требующееся для взлома 30-битового ключа при помощи атаки «в лоб». Это значит, что хотя у алгоритма имеется ключ с 64-битовой энтропией, он задействует для ключа только 30 бит энтропии. Вы можете с тем же успехом использовать хороший алгоритм с 30-битовым ключом.

По этой причине проходит довольно много времени, прежде чем шифровальщики начинают доверять новому алгоритму. Когда кто-то предлагает новый алгоритм, у него есть определенная длина ключа. Но обеспечивает ли алгоритм реально ту энтропию, которая заявлена? Может потребоваться несколько лет анализа, прежде чем мы поверим, что он это делает. И даже тогда мы можем легко ошибиться: возможно, кто-то придумает новые математические подходы, которые понизят энтропию алгоритма и сломают его. Поэтому рекламу программ, в которых обещаются тысячебитовые ключи, трудно воспринимать серьезно — ее создатели не имеют понятия, как работают ключи и энтропия.

Похожая проблема существует и для физических ключей и замков. Принято думать, что слесарь возит в своем грузовике огромное кольцо с ключами от машин. Может потребоваться 10 000 ключей, чтобы открыть все замки, но в реальности несколько дюжин ключей откроют любой из них. Иногда слесарю достаточно просто взять другой ключ, отличающийся от предыдущего на 1-2 «бита», — отметим, что это комбинация анализа и лобовой атаки — и этого уже достаточно. Да, процесс долгий, но совсем не такой, как проверка всех 10 000 возможных ключей (старые замки — четырехштырьковые). Действительная надежность дверного замка существенно отличается от теоретической.

То же самое с комбинациями замков. Вы можете перебрать все возможные комбинации — и существуют машины для взлома сейфов, которые так и делают, — или поступить хитрее. Современные машины для взлома сейфов применяют мик-

[1] Инициализация генератора случайных чисел была основана на значении текущего времени в микросекундах и идентификаторах процесса. Исследователи Голдберг и Вагнер выяснили, что для 128-битового ключа это равнозначно 47-битовой энтропии. Используя сетевые домены, они получали вероятные значения инициализации, после чего находили 40-битовый ключ за 1 минуту, перебирая микросекунды. — *Примеч. ред.*

рофон, чтобы слушать звук, производимый дисками, когда их поворачивают, и они могут открыть сейф намного быстрее, чем старые, действующие «в лоб».

Сказанное здесь заставляет очень внимательно подходить к выбору алгоритма. Мы еще обсудим это более детально в конце этой главы.

## Одноразовое кодирование

Кодирование одноразового использования — это самый простой из всех алгоритмов, его изобрели незадолго до XX века. Основная идея его состоит в том, что у вас есть набор символов ключа. Вы прибавляете один символ ключа к каждому символу открытого текста и никогда не повторяете символы ключа. (Это «одноразовая» часть.) Например, вы прибавляете B (2) к C (3), чтобы получить E (5), или T (20) к L (12), чтобы получить F (6). ((20 + 12) mod 26 = 6.) Такая система подходит для любого алфавита, в том числе и бинарного. И это единственный имеющийся у нас алгоритм, безопасность которого может быть доказана[1].

Вспомним понятие расстояния уникальности. Оно возрастает при увеличении длины ключа. Когда длина ключа приближается к длине сообщения, расстояние уникальности стремится к бесконечности. Это означает, что невозможно восстановить открытый текст, и это доказывает безопасность одноразового кодирования.

Но, с другой стороны, это практически бесполезно. Поскольку ключ должен при этом быть такой же длины, как и сообщение, то проблема не решена. Единственный здравый подход к шифрованию должен предполагать, что очень длинная секретная информация — сообщение — превращается с его помощью в очень короткую секретную информацию — ключ. При помощи одноразового кодирования вы нисколько не сокращаете секретную информацию. Так же сложно доставить шифр получателю, как и доставить само сообщение. Современная криптография зашифровывает большие объекты, например цифровые фильмы, соединения через Интернет или телефонные разговоры, и такое шифрование практически невозможно осуществить, работая с одноразовым кодированием.

Одноразовое кодирование практически использовалось в особых случаях. Русские шпионы применяли для общения алгоритм одноразового кодирования, используя карандаш и бумагу. Агентство национальной безопасности (NSA) раскрыло эту систему, поскольку русские использовали ее повторно. Горячая линия телетайпа между Вашингтоном и Москвой была зашифрована именно таким образом.

Если утверждают, что некая программа использует этот алгоритм, то здесь почти наверняка обман. А если и нет, то программа наверняка непригодна для использования или небезопасна.

---

[1] Так называемый шифр Гронсфельда (взятый в основу одноразового кодирования) и его вариации суть модификации шифра Юлия Цезаря (где «длина ключа» была равна одному символу). Был популярен в конце XIX — начале XX века: в основу романа Жюля Верна «Жангада» была положена расшифровка именно этого шифра. Абсолютная стойкость шифра в предположении равенства длины сообщения и ключа была доказана Клодом Шенноном. — *Примеч. ред.*

# Протоколы

Шесть инструментов, о которых я говорил в предыдущей главе, — симметричное шифрование, коды аутентификации сообщений, шифрование с открытым ключом, односторонние хэш-функции, схемы цифровых подписей и генераторы случайных чисел — составляют набор инструментов шифровальщика. С его помощью мы выстраиваем криптографические решения реальных задач: «Как мне послать секретное письмо по электронной почте? Как можно предотвратить мошенничества с телефонными звонками? Как мне обеспечить безопасность системы голосования через Интернет?» Эти задачи безопасности мы решаем, компонуя простейшие элементы в так называемые протоколы. Приходится использовать и другие второстепенные элементы, но, по существу, шесть перечисленных выше элементов составляют ядро любого криптографического протокола.

Например, предположим, что Алиса хочет сохранить в тайне некоторые файлы данных. Вот протокол, который это делает. Алиса выбирает пароль или даже лучше — ключевую фразу. Криптографические программы хэшируют этот пароль, чтобы получить секретный ключ, а затем, применяя симметричный алгоритм, зашифровывают файл данных. В результате получится файл, доступ к которому есть только у Алисы или у того, кто знает пароль.

Хотите создать безопасный телефон? Используйте криптографию с открытым ключом, чтобы сформировать сеансовый ключ, а затем при помощи этого ключа и симметричной криптографии зашифруйте переговоры. Хэш-функция обеспечивает дополнительную безопасность против атак, проводимых человеком. (Подробнее об этом позже.) Чтобы засекретить сообщение электронной почты, воспользуйтесь криптографией с открытым ключом для соблюдения секретности и схемой цифровой подписи для аутентификации. Электронная торговля? Обычно для нее не требуется ничего, кроме цифровых подписей и, иногда, шифрования для секретности. Секретный контрольный журнал? Возьмите хэш-функции, шифрование, может быть, MAC и перемешайте.

То, что мы сейчас делаем, и есть создание протоколов. Протокол — это не сложнее, чем танец. Это последовательность заданных шагов, которую выполняют два (или больше) партнера и которая предназначена для решения поставленной задачи. Представьте себе протокол, которым пользуются продавец и покупатель при покупке мандаринов. Вот необходимые шаги:

1. Покупатель спрашивает у продавца мандарины.
2. Продавец дает ему мандарины.
3. Покупатель платит продавцу.
4. Продавец возвращает ему сдачу.

Все, о ком идет речь в протоколе, должны знать шаги. Например, покупатель знает, что он должен заплатить за мандарины. Все шаги должны быть однозначны: ни продавец, ни покупатель не могут достичь шага, на котором они не знают, что делать. Кроме того, протокол должен обязательно завершаться — в нем не должно быть бесконечных циклов.

Шаг 2 не будет работать как надо, если продавец не поймет семантическое содержание шага 1. Продавец не выполнит шаг 4, если на шаге 3 не признает деньги

настоящими. Попробуйте купить мандарины в США на польские злотые и посмотрите, как вам это удастся.

Нас особо волнуют протоколы безопасности. Кроме всех перечисленных выше требований мы хотим, чтобы покупатель и продавец не имели возможности обманывать (что бы ни означало «обманывать» в нашем контексте). Мы не хотим, чтобы продавец мог заглянуть в бумажник покупателя на шаге 3. Мы не хотим, чтобы продавец не отдал покупателю сдачу на шаге 4. Мы не хотим, чтобы покупатель застрелил продавца на шаге 3 и ушел с украденными мандаринами. Такие способы обмана распространены и в материальном мире, а анонимность киберпростран¬ства только увеличивает опасность.

Даже в физическом мире были разработаны более сложные протоколы, чтобы снизить опасность всякого рода жульничества. Вот как в общих чертах выглядит протокол продажи автомобиля.

1. Алиса передает Бобу ключи и документы.
2. Боб дает Алисе чек на сумму покупки.
3. Алиса кладет деньги в банк.

Здесь Боб запросто может сжульничать. Он может дать Алисе фальшивый чек. Она не будет знать, что чек фальшивый, и не узнает, пока ей не скажут об этом в банке. К тому времени Боб с ее машиной будет уже далеко.

Когда я продавал машину несколько лет назад, во избежание такой атаки придерживался несколько модифицированного протокола.

1. Боб выписывает чек и идет с ним в банк.
2. Когда на счет Боба поступит достаточно денег, чтобы покрыть чек, банк «заверяет» чек и возвращает его Бобу.
3. Алиса передает Бобу ключи и документы о праве собственности.
4. Боб дает Алисе заверенный чек на сумму покупки.
5. Алиса кладет деньги в банк.

Что при этом происходит? Банк в такой коммерческой сделке выступает доверенной третьей стороной. Алиса полагается на то, что банк выплатит по заверенному им чеку полную сумму. Боб верит, что банк сохранит деньги на счете, а не вложит их в рискованные предприятия в странах третьего мира. Алиса и Боб могут осуществить свою сделку, даже если они не доверяют друг другу, — поскольку они оба доверяют банку.

Такая система работает не потому, что банк — солидное учреждение, за которым стоит много, включая хорошую рекламную кампанию, а потому, что банк не заинтересован в выгоде кого-либо одного — Алисы или Боба — и имеет надежную репутацию. Банк будет придерживаться протокола заверения чека, что бы ни произошло. Если на счету Боба достаточно денег, банк выдаст чек. Если Алиса представит чек к оплате, банк его оплатит. Если бы банк все же скрылся с деньгами, не многое бы от него осталось. (В этом и есть сущность репутации.)

Данный протокол защищает Алису, но не Боба от покупки фальшивых документов или краденой машины. Для этого нам нужен другой протокол.

1. Алиса передает ключи и документы о праве собственности юристу.
2. Боб отдает юристу чек.

3. Юрист проверяет чек в банке.
4. По прошествии оговоренного временного промежутка, который отводится на то, чтобы проверить чек и зарегистрировать машину, юрист отдает Бобу документы. Если клиринг чека не удается осуществить в течение оговоренного времени, юрист возвращает документы Алисе. Если Боб не может получить новые документы на машину (поскольку Алиса дала ему недействительные документы), Боб предоставляет юристу соответствующие доказательства и получает назад свои деньги.

Как и в предыдущем протоколе, здесь привлекается третья доверенная сторона. В данном случае это юрист. Алиса не доверяет Бобу, Боб не доверяет Алисе, но оба они верят, что юрист на последнем шаге поведет себя беспристрастно. Юрист совершенно не заинтересован в чьей-либо выгоде — ему все равно, отдаст он документы Бобу или Алисе. Он задержит деньги на депоненте и будет действовать в соответствии с соглашением между Бобом и Алисой.

Другие протоколы могут быть более простыми и не требовать столь сложного обмена. Например, протокол, в соответствии с которым банк в состоянии удостовериться, что чек подписан Алисой.

1. Алиса подписывает чек.
2. Банк сравнивает подпись на чеке с подписью на папке Алисы.
3. Если они совпадают, банк дает Алисе деньги. Если не совпадают — не дает.

Теоретически, такой протокол обеспечивает защиту от того, что Боб обманным путем получит деньги Алисы, но, конечно, в реальности все сложнее. Боб может уметь подделывать подписи. Банк может давать рискованные займы Парагваю и разориться. Алиса может вытащить пистолет. Существуют, вероятно, тысячи способов нарушить этот протокол, но с разумными допущениями он работает.

Протоколы цифрового мира во многом похожи на предыдущие примеры. Цифровые протоколы при помощи криптографии делают то же самое: сохраняют секреты, проводят аутентификацию, охраняют справедливость, обеспечивают аудит и т. п.

В Интернете используется множество протоколов безопасности; я расскажу о них в следующем разделе. У других цифровых сетей есть свои протоколы безопасности. Индустрия сотовых телефонов применяет целый ряд протоколов — как для секретности, так и для предотвращения мошенничества (с переменным успехом). У компьютерных приставок к телевизору также есть протоколы безопасности. И у смарт-карт тоже.

Протоколы, содержащие цифровые подписи, особенно полезны для различных задач аутентификации. Например, при помощи схемы цифровых подписей можно создать подписи, которые сможет идентифицировать только определенный получатель. Это полезно для информаторов или доносчиков, поскольку получивший сообщение в силах установить, кто его отправил, но не сумеет доказать это третьему лицу. (Представьте, что вам на ухо прошептали секрет. Вы знаете, кто вам его сообщил, но никак не можете доказать, что это сделал именно он.) При помощи протоколов цифровых подписей можно подписать программу, и тогда только тот, кто приобрел программный пакет законным образом, сможет проверить подпись и узнать, что это не подделка; обладатели пиратских копий не могут быть в этом уверены. Мы можем создать групповые подписи, так что для людей, не входящих

в группу, подпись будет казаться подписью группы в целом, а члены группы различат, чья конкретно эта подпись.

Более сложные протоколы дают криптографии возможность прыгнуть через любые препятствия. Можно предоставлять так называемые доказательства с нулевыми знаниями, когда Алиса способна доказать Бобу, что она что-то знает, не открывая, что именно. Криптографические протоколы также могут обеспечивать систему одновременного подписания контрактов через Интернет. Можно создать цифровой аналог заказных писем, когда Алиса сумеет прочесть письмо, только отослав обратно «расписку в получении».

При помощи протокола, называющегося *совместным управлением секретностью* (secret sharing), мы можем предписывать требование *соглашения о доступе (collusion in access)* — секрет, который нельзя узнать, если многие люди не действуют сообща. Это — действительно необходимая вещь. Представьте себе ядерную ракетную установку. Для того чтобы запустить ракету, два человека одновременно должны повернуть ключи и разблокировать систему. А замочные скважины (или в данном случае их цифровой аналог) достаточно сложны, чтобы один-единственный солдат не мог повернуть все ключи и убить всех; чтобы запустить ракету, по крайней мере два человека должны действовать сообща. Или представьте себе контроль счетов корпорации, в соответствии с которым для особо ценных чеков требуется две подписи: по крайней мере два из пяти членов правления корпорации должны подписать чек. Мы можем создать что-то подобное при помощи криптографии.

Получится даже еще лучше. Можно создать протокол для тайного голосования в Интернете — такой, что только зарегистрированный избиратель сможет проголосовать, никто не сумеет проголосовать дважды, никто не сможет узнать, кто за кого голосовал, и все будут уверены, что выборы честные. Можно даже создать электронную валюту, однако кто-нибудь попытается использовать эти платежные средства дважды...

Честно говоря... если вы захотите, мы сможем сделать и это.

Однако существуют и проблемы, которым посвящено достаточно места в этой книге.

# Криптографические протоколы Интернета

Для Интернета криптография относительно нова и появилась она в нем только благодаря коммерциализации сети. Интернет небезопасен; чтобы обеспечить безопасность в сети, необходима криптография. Вот почему на практике криптографические протоколы являются важнейшей составной частью протоколов Интернета. Примеры, приведенные здесь, характерны для 2000 года, со временем их характер, конечно, будет меняться.

Первой областью применения криптографии в Интернете стала электронная почта. В ней работали два соответствующих протокола: S/MIME и OpenPGP. OpenPGP — это протокол в составе PGP и его разновидностей. S/MIME — стандартный интернет-протокол во всех других случаях.

Netscape изобрела SSL (Secure Sockets Layer — протокол, гарантирующий безопасную передачу данных по сети; комбинирует криптографическую систему

с открытым ключом и блочным шифрованием данных) на заре существования Веб, когда люди захотели заниматься безопасной электронной торговлей при помощи своих браузеров. SSL существовал в нескольких воплощениях (он был полем боя во время войны браузеров Netscape и Microsoft и в итоге был назван TLS (Transport Layer Security)). Эти протоколы встроены в браузеры и позволяют людям зашифровать секретную информацию, посылаемую на различные веб-сайты.

Более новые криптографические протоколы разработаны для защиты пакетов IP. Среди них Microsoft Point-to-Point Tunneling Protocol (PPTP, у которого есть грубые дефекты), Layer Two Tunneling Protocol (L2TP) и IPsec (он существенно лучше, хотя и слишком сложен). IKE (Internet Key Exchange) — это, как видно из названия, протокол обмена ключами. Сегодня эти протоколы используются преимущественно для того, чтобы обеспечить работу виртуальных частных сетей (VPN). Тем не менее протоколы безопасности Интернета «умеют» намного больше, чем протоколы VPN. У них есть возможность обеспечивать безопасность большей части трафика. Со временем, может быть, эта возможность реализуется.

Существуют также и другие интернет-протоколы. SET — разработанный компаниями Visa и MasterCard для защиты операций с кредитными картами во Всемирной паутине. (Эти протоколы никогда не будут широко применяться.) Протокол SSH (Secure Shell — защитная оболочка) используется для шифрования и идентификации команд для удаленных соединений. Другие протоколы имеют дело с сертификатами открытых ключей и инфраструктурой сертификатов: PKIX, SPKI и им подобные. Microsoft использует свои протоколы для защиты Windows NT.

Большая часть этой работы была проделана под эгидой Проблемной группы проектирования Интернета (Internet Engineering Task Force, IETF — одна из групп IAB, отвечающая за решение инженерных задач Интернета, выпускает большинство RFC, используемых производителями для внедрения стандартов в архитектуру TCP/IP). Процесс нуждается в тщательном согласовании, а это значит, что создание подобных вещей требует длительного времени и в результате они получаются более сложными, чем могли бы быть. Как мы увидим позднее, эта сложность не является хорошим фактором.

## Типы атак, направленных на протоколы

Так же как существует много различных атак, мишенью которых служат алгоритмы, множество их направлено и против протоколов. Простейшие из них — это *пассивные атаки:* вы просто присматриваетесь к протоколу и ищете, что можно понять. Часто в результате простого подслушивания можно узнать очень многое.

Существует множество веб-сайтов электронной почты. Чтобы ими воспользоваться, вы идете на этот сайт, набираете ваше имя и пароль. Как правило, такой протокол уязвим для атаки перехвата. Другая серия протоколов, уязвимых для атаки-перехвата, — протоколы для предупреждения мошенничеств с аналоговыми телефонами. Кто-то, у кого есть сканирующее устройство, может прослушать соединение телефона с базовой станцией, а затем оплачивать свои звонки со счета того телефона. (Это называется *телефонным клонированием*. Цифровые сотовые телефоны в этом смысле лучше, но ненамного.)

Узким местом атак перехвата является то, что не всегда понятно, какая информация представляет ценность. Можете представить себе зашифрованную телефонную сеть, в которой невозможно (учитывая секретность, обеспечиваемую криптографией) подслушивать телефонные разговоры. Однако информация о подключениях остается доступной. Эта информация часто тоже бывает полезной. В военной обстановке, например, вы можете многое узнать из анализа трафика: кто с кем говорит, когда и как долго.

Более сложные атаки — *активные:* вставка, удаление и изменение сообщений. Они могут оказаться существенно более действенными.

Рассмотрим систему смарт-карт. Люди кладут деньги на счет и затем используют карточку для оплаты. Эта система будет состоять из множества различных протоколов: для помещения денег на карту, для перевода денег с карты на другое устройство, для запросов информации о карте и других.

Активные атаки способны нанести множество повреждений такой системе. Предположим, вы можете вмешиваться в протокол между банком и картой. Если вам доступно воспроизведение старых сообщений, вы можете добавить на карту побольше денег. Или, например, вы умеете удалять сообщение в протоколе перевода денег с карты при покупке — тогда сумма на карте никогда не будет уменьшаться.

Одна из мощных атак — это *атака посредника, «человека посередине» (man-in-the-middle-attack).* Алиса хочет тайно поговорить с Бобом, применяя какой-то алгоритм с открытым ключом, чтобы создать свой ключ. Ева перехватывает сообщение Алисы. Она представляется Алисе как Боб, завершая протокол обмена ключами. Затем она связывается с Бобом и представляется как Алиса, выполнив тем самым второй протокол обмена ключами с Бобом. После этого она может прослушивать связь. Когда Алиса посылает Бобу сообщение, Ева его перехватывает, расшифровывает, зашифровывает заново и посылает Бобу. Когда Боб посылает сообщение для Алисы, Ева проделывает аналогичную процедуру. Это очень действенная атака.

Безусловно, грамотные разработчики протоколов принимают во внимание такие атаки и пытаются предотвращать их. Лучшие протоколы связи не допускают атак посредничества и, конечно, не дают возможности перехвата пароля. Лучшие протоколы электронной торговли не допускают, чтобы злонамеренные пользователи произвольно добавляли деньги на смарт-карты. Но людям свойственно ошибаться, и во множестве протоколов есть проблемы.

И, повторим снова, не всегда очевидно, какого рода атаки необходимо предотвращать. Существовал протокол идентификации открытого ключа, который описан в литературе, сконструированный так, что пользователи могли аутентифици¬ровать себя для хостов. Этот протокол защищал от атак пассивного прослушивания и от активных атак вставки или удаления. Но, как оказалось, он не защищал от хостов злоумышленников. Алиса может подтвердить хосту свою подлинность, и ни один перехватчик не сможет выдать себя за Алису. Но хост сможет.

Это — интересная атака. В одних обстоятельствах полагают, что хост достоин доверия и этой проблемы нет. В других случаях проблема налицо. Нам легко представить себе, что злонамеренные хосты во Всемирной паутине есть. Если бы банк, работающий через Интернет, использовал этот протокол (насколько мне известно, такого не было), то преступники смогли бы создать ложный банковский вебсайт, у которого немного отличался бы URL. Не подозревающие ни о чем пользо¬

ватели аутентифицировали бы себя на этом ложном сайте, а он затем представлялся бы реальному банку пользователем.

Множество подобных вещей формализовано. Существуют автоматические инструменты (сервисные программы) для анализа протоколов: формальные логики, компьютерные программы, которые анализируют детали протоколов, и другие. Эти инструменты полезны, они регулярно обнаруживают проблемы в существующих протоколах, но с их помощью нельзя «доказать» надежность протокола.

## Выбор алгоритма или протокола

Выбрать криптографический алгоритм или протокол трудно, поскольку нет абсолютных критериев. Мы не можем сравнивать алгоритмы шифрования по тому же принципу, по которому сравниваем алгоритмы сжатия. Со сжатием все просто: вы можете наглядно доказать, что один алгоритм сжимает лучше другого — быстрее, до меньшего размера, по любым другим параметрам. С безопасностью сложнее: хоть вы и можете показать, что определенный алгоритм ненадежен, нельзя доказать, что один алгоритм, который вы не сумели сломать, безопаснее другого. За неимением абсолютных критериев мы используем то доказательство, которое у нас есть: оценку экспертов.

Эту проблему лучше всего проиллюстрировать примером. Допустим, врач говорит вам: «Я знаю, что есть антибиотик, который хорошо помогает при лечении вашей инфекции; у него нет вредных побочных эффектов и десятилетия исследований говорят в пользу такого лечения. Но я собираюсь назначить вам вместо этого толченые сухари, потому что мне кажется, что они тоже помогут». Вы пошли бы к другому врачу.

Врачебная практика непроста. Представители этой профессии не рвутся применять новые лекарства; проходят годы испытаний, прежде чем будет доказана их действенность, установлена дозировка и составлен список побочных эффектов. Хороший врач не будет лечить бактериальную инфекцию лекарством, которое он только что придумал, если в наличии имеются испытанные антибиотики. А умный пациент захочет те же таблетки, которые помогли его приятелю, а не какие-то там другие.

Криптография также сложна. Она соединяет различные разделы математики и вычислительную технику. Она требует многих лет практики. Даже умные, знающие, опытные люди изобретают плохую криптографию. В криптографическом сообществе создатели даже не слишком расстраиваются, когда их алгоритмы и протоколы ломают. Вот насколько все трудно.

Проблема вот в чем: любой человек, сколь угодно неопытный, может разработать элемент криптографии, который сам взломать не может. Это — существенно. Это значит, что кто угодно может сесть и создать криптографический элемент, попытаться его взломать, потерпеть неудачу, а затем сказать: «Я изобрел безопасный алгоритм, протокол или что-то еще». Реально он говорит этим: «Я не могу его взломать, поэтому он безопасен». Первый вопрос, который надо задать в ответ: «Да кто ты такой?» Или более пространно: «Почему я должен верить в надежность чего-то, если ты не смог это взломать? Чем подтверждается то, что если у тебя это не вышло, то и никто другой не сможет этого сделать?»

Криптографическое сообщество обнаружило, что ни один человек не готов предоставить такие доказательства. (Может быть, и есть кто-нибудь в Агентстве на-

циональной безопасности, но эти люди не болтливы.) Нет никакого способа доказать надежность элемента — можно либо продемонстрировать ненадежность, либо признать попытку неудавшейся. Это называется проверкой гипотезы с нулевым разглашением. Лучшее, что могут сказать люди, занимающиеся безопасностью: «Мы не знаем, как взломать этот алгоритм, протокол или что-то другое, и никто другой тоже не знает». Экспертная оценка программы, длительный период испытаний — вот единственное доказательство безопасности, которое у нас есть.

Более того, нет никакого смысла нанимать группу случайных людей, оценивающих элемент; единственный способ отличить хорошую криптографию от плохой — это получить оценку специалистов. Анализировать криптографию трудно, и немногим по плечу делать это грамотно. Прежде чем элемент можно будет действительно считать безопасным, его должны проверять многие эксперты на протяжении ряда лет.

Вот поэтому шифровальщики предпочитают старое и общедоступное новому и самодельному. Открытая криптография — это то, что шифровальщики изучали, о чем писали статьи. О старых средствах написана масса статей. Если бы там были слабые места, их бы уже обнаружили. Новое же наверняка опаснее, потому что оно новое и не исследовано должным образом специалистами.

Посмотрите на следующие три варианта протоколов безопасности.

- **IPsec.** Его разработка началась в 1992 году. Разработка велась комиссией «в открытую» и была предметом тщательного публичного изучения с самого начала. Все знали, что это важный протокол, и огромные усилия прилагались для того, чтобы все было правильно. Алгоритмы защиты предлагали, взламывали, а затем модифицировали. Версии классифицировались и анализировались. Первый проект стандарта был выпущен в 1995 году. Обсуждались достоинства безопасности и эффективность, простота исполнения, возможности дальнейшего расширения и применения. В 1998 году комиссия представила окончательный вариант протокола. До сих пор каждый, кто интересуется, может открыто его изучать.
- **PPTP.** Фирма Microsoft разработала свой собственный Point-to-Point Tunneling Protocol (новая сетевая технология, которая поддерживает многопротокольные виртуальные частные сети, позволяя удаленным пользователям безопасно обращаться к корпоративным сетям с помощью коммутируемого соединения, предоставляемого интернет-провайдером или с помощью прямого соединения с Интернетом), который должен выполнять во многом схожие с IPsec функции. Был создан свой протокол аутентификации, свои хэш-функции и свои алгоритмы генерации ключа. Все эти элементы оказались крайне слабыми. В них использовался известный алгоритм шифрования, но использовался таким образом, что не обеспечивал безопасности. Программисты допустили ошибки в реализации, которые еще больше ослабляли систему. Но поскольку их коды были спрятаны внутри, никто не заметил, что PPTP недостаточно надежен. Microsoft использовала PPTP в Windows NT, 95 и 98, а также в своих продуктах для виртуальных частных сетей. Статьи, описывающие недостатки протокола, не публиковались до 1998 года.
- **Право собственности.** Некоторые компании объявляют о своих собственных решениях задачи безопасности. Они не вдаются в детали или из-за того, что это право собственности, или из-за неоформленности патента. Вам прихо¬

дится им доверять. Разработчики могут заявить о новом алгоритме или протоколе, который во многом превосходит имеющиеся сегодня. Они могут кричать о математических прорывах... о чем угодно. Очень немногое из этого оказывается правдой. И даже если к системам предоставляется открытый доступ, их запатентованность и контроль соблюдения авторских прав означают, что немногие шифровальщики будут озадачены анализом заявленных преимуществ. С другой стороны, даже если шифровальщики займутся этой проблемой, компании, конечно, не будут ждать годы, пока исследования подтвердят надежность новинок.

Вы можете выбрать одну из этих трех систем для обеспечения надежности своей виртуальной частной сети. Хотя у любой из них есть слабые места, вы сделаете опасность минимальной. Если вы работаете с IPsec — больше уверенности в надежности алгоритмов и протоколов. Конечно, это не гарантия безопасности — реализация может оказаться слабой (см. главу 13) или кто-то придумает новые способы атаки — но по крайней мере вы знаете, что алгоритмы и протоколы прошли какой-то уровень анализа.

Другой пример: рассмотрим симметричный алгоритм шифрования. Их, безусловно, существуют сотни, но давайте ограничимся пятью.

- Тройной DES, который начиная с середины 70-х был проанализирован практически всем криптографическим сообществом.
- AES, который (прежде чем его выберут) будет подвергнут трехлетнему тестированию, вовлекающему практически все криптографическое сообщество.
- Некий алгоритм X, который был представлен на академической конференции два года назад; пока что вышла одна статья с анализом, судя по которой это — надежный алгоритм.
- Алгоритм Y, который кто-то недавно поместил в Интернете и заверил нас в его надежности.
- Алгоритм Z, который компания сохраняет в секрете до получения патента; возможно, они нанимали несколько шифровальщиков для трехнедельного анализа.

Это — нетрудный выбор. Могут существовать ограничения, которые не позволят вам выбрать тот алгоритм, который вы хотите (AES существует главным образом потому, что тройной DES слишком медлителен для многих сред), но выбор достаточно ясен.

Меня постоянно изумляет, как часто люди не выбирают очевидного решения. Вместо того чтобы использовать общедоступные алгоритмы, компании цифровой сотовой связи решили создать собственный запатентованный алгоритм. За последние несколько лет все алгоритмы стали общедоступными. И став общедоступными, они были взломаны. Каждый из них. То же самое случилось с алгоритмом DVD-шифрования, алгоритмом шифрования Firewire, различными алгоритмами шифрования Microsoft и бесчисленным множеством других. Любой, кто создает собственный образец шифрования, — гений или глупец. С учетом соотношения гениев и глупцов в нашей действительности шансы выжить у образца невелики.

Иногда приводится следующий контраргумент: секретная криптография надежнее, потому что она тайная, а открытая криптография опаснее, поскольку она от¬

крытая. Это звучит правдоподобно, но если вы на минуту задумаетесь, несоответствия станут очевидными. Открытые образцы созданы так, чтобы обеспечивать безопасность, несмотря на свою открытость. Таким образом, их не опасно сделать общедоступными. Если элемент обеспечивает безопасность, только оставаясь секретным, то он будет работать до тех пор, пока кто-нибудь не разберется в его устройстве и не опубликует способ взлома. Выпускаемые запатентованные элементы включают все алгоритмы, описанные в предыдущем разделе, различные протоколы смарт-карт для электронной торговли, секретные хэш-функции в картах SecurID и протоколы, защищающие мобильный терминал данных полиции MDC-4800 компании Motorola.

Отсюда не вытекает, что все новое ущербно. Что это на самом деле значит — так то, что все новое подозрительно. Новая криптография появляется в академических статьях, а затем в демонстрационных системах. Если она действительно лучше, то в конце концов шифровальщики начнут ей доверять. И только тогда будет разумно использовать ее в реальных программах. Для алгоритма этот процесс может занять от 5 до 10 лет, для протокола или библиотек исходных кодов — поменьше.

Предпочесть патентованную систему — это то же самое, что обратиться к врачу, у которого нет медицинского образования и который лечит по собственной новой методике (какой — он отказывается объяснить), не поддержанной Американской медицинской ассоциацией. Безусловно, возможно (хотя и крайне маловероятно), что он открыл абсолютно новую область медицины, но хотите ли вы быть подопытной морской свинкой? Лучшие методы безопасности усиливаются коллективными аналитическими способностями криптографического сообщества. Ни одна отдельно взятая компания (за исключением военных) не имеет финансовых ресурсов, необходимых для оценки нового криптографического алгоритма или обнаружения недостатков сложного протокола.

В криптографии безопасность приходит путем следования за широкими массами. Доморощенные алгоритмы невозможно подвергать в течение сотен и тысяч часов криптоанализу, через который прошли DES и RSA. Компания или даже промышленное объединение не могут мобилизовать ресурсы, которые использовались, чтобы противостоять аутентификационному протоколу Kerberos или IPsec. Ни один патентованный почтовый протокол шифрования не в состоянии повторить конфиденциальность, предлагаемую PGP или S/MIME. Следуя в общей струе, вы обеспечиваете уровень криптоаналитической экспертизы всемирного сообщества, а это вам — не несколько недель работы ничем не выдающихся аналитиков.

Довольно трудно обеспечить надежную криптографическую обработку в новой системе; просто безумие использовать новую криптографию, когда существуют жизнеспособные тщательно изученные альтернативы. И все же большинство компаний, занимающихся безопасностью, и даже умные и здравомыслящие во всем остальном люди проявляют острую «неофилию», и их легко ослепляют свежеиспеченные блестящие образчики криптографии.

И остерегайтесь врача, который говорит: «Я изобрел и запатентовал абсолютно новый способ лечения, который состоит в употреблении толченых сухарей. Этот метод еще не испробован, но я уверен, что он великолепен». Помните, что новую криптографию часто называют *змеиным ядом*.

# Глава 8. Компьютерная безопасность

Защита компьютерной информации и криптография — не одно и то же. Криптография часто применяется в целях защиты, но последняя представляет собой намного более общее понятие. В общепринятом понимании компьютерная защита объединяет такие разные вещи, как контроль санкционированного (и несанкционированного) доступа, управление учетными записями и привилегиями пользователя, защиту от копирования, от вирусов и защиту баз данных. В принципе, к защите компьютерной информации также относятся защита от подсоединения других пользователей через сеть, от подбора пароля и от проникновения вирусов, но такого рода вещи мы обсудим в главе, посвященной безопасности сетей. В век Интернета понятия компьютерной безопасности и безопасности сетей практически слились. Но для ясности в этой книге я проведу некую условную границу между понятиями компьютерной и сетевой безопасности по принципу: актуальна эта проблема безопасности для любого компьютера или только для компьютера, подсоединенного к сети. Полная защита компьютерной информации, которую можно определить как предотвращение и (или) выявление недозволенных действий пользователей компьютерной системы, представляется существенно более сложной, чем простая математика криптографии. Так оно и есть.

Суть проблемы состоит в том, что одна математика не может обеспечить полную безопасность. В криптографии математика дает защите огромное преимущество перед злоумышленником. Добавьте один бит к ключу — и вы вдвое усложните работу по взлому алгоритма. Добавьте десять битов — и вы увеличите эту работу примерно в тысячу раз. Когда речь идет о компьютерной безопасности в целом, стороны находятся в равном положении: злоумышленники и защитники могут извлечь из технологии одинаковую выгоду. Это значит, что, если бы вам было достаточно криптографии для обеспечения безопасности, у вас все было бы в порядке. К сожалению, в большинстве случаев это не так.

Большинство ранних исследований по компьютерной безопасности было посвящено проблеме персонального доступа в системах совместного пользования. Как сделать так, чтобы Алиса и Боб могли пользоваться одним и тем же компьютером и одинаковыми компьютерными программами, но чтобы Алиса не могла видеть, что делает Боб, а Боб не знал, что делает Алиса? Или в общем случае: если системой пользуется большая группа людей, у каждого из которых есть определенные права использовать определенные программы и видеть определенные данные, то как мы можем реализовать такие правила контроля доступа? Вообще говоря, это не та задача, которую можно решить с помощью криптографии, хотя в чем-то она могла бы помочь. Это новая задача.

Компьютерная безопасность требует преодоления и многих других новых проблем. Как компании обеспечить правильную работу с большой базой данных, к которой разные люди имеют различный доступ? Эта проблема может быстро стать чрезвычайно сложной. Только несколько человек имеют право видеть информацию о зарплате, еще меньше людей должны видеть всю совокупность данных: среднюю зарплату, статистические данные о здоровье и т. д.

Могут ли пользователи быть уверены в том, что используемые ими компьютерные программы исправны, что они не были модифицированы? Как им удостовериться, что их данные не изменялись? Как компания, производящая программы, может обеспечить выполнение правил лицензирования: нельзя копировать программы с машины на машину, программу можно одновременно запустить только на пяти компьютерах, только десять пользователей одновременно могут работать с этой программой, программа может работать только в течение одной тысячи часов?

Все это — серьезные требования, и задачи компьютерной безопасности имеют сложные решения.

# Определения

Попытки определения понятия компьютерной безопасности стоили огромных усилий. Исторически проблема безопасности имеет три аспекта: конфиденциальность, неприкосновенность и доступность.

*Конфиденциальность* немногим отличается от секретности, о которой мы говорили в главе 5. Изначально компьютерная безопасность понималась как предотвращение несанкционированного доступа к засекреченной информации. Это предубеждение отчасти развеялось с появлением электронной торговли и практики совершения сделок в Интернете — в этой сфере существенно более важна неприкосновенность,— однако оно сохраняется при разработке большинства систем обеспечения компьютерной безопасности. Основная масса исследовательских работ на тему компьютерной безопасности сосредоточена на обеспечении конфиденциальности, главным образом потому, что большая часть ранних исследований финансировалась военными. На практике, как я заметил, понятия «конфиденциальность» и «безопасность» использовались как синонимы.

Понятию *неприкосновенности* труднее дать строгое определение. Лучшая из известных мне формулировок звучит так: все данные сохраняются в таком виде, в каком они были оставлены последним лицом, правомочным вносить изменения. В контексте компьютерной безопасности «неприкосновенность» означает защиту от записи. Неприкосновенность данных — это гарантия того, что их не удалит и не изменит кто-то, у кого нет на это права. Неприкосновенность программного обеспечения — это гарантия того, что программы не будут изменены по ошибке, по злому умыслу пользователя или вирусом.

Из определения неприкосновенности видно, что эта проблема аналогична проблеме обеспечения конфиденциальности. Если последняя перекрывает несанкционированный доступ к данным (и программам), то первая предотвращает несанкционированную запись. И фактически обе эти задачи решаются при помощи одних и тех же технологий безопасности (криптографических и других).

*Доступность* традиционно считают третьим «китом» компьютерной безопасности, хотя на самом деле понятие доступности выходит далеко за рамки этой проблемы. В различных стандартах по обеспечению защиты доступность определяется как «свойство системы, состоящее в том, что ее беспрепятственная эксплуатация возможна, когда это необходимо» или как «свойство системы быть готовой и пригодной к работе по требованию законного пользователя». Эти определения всегда поражали меня недостаточной конкретностью. Их смысл сводится к следующему: мы интуитивно знаем, что подразумеваем под доступностью — нам нужно, чтобы компьютер работал, когда мы того хотим, и так, как мы того хотим.

Конечно, бывает, что программы не работают или работают неправильно, но это — проблемы надежности вычислительных систем и качества программных продуктов и... ни одна из них не имеет отношения к безопасности. В контексте безопасности под доступностью можно понимать гарантию того, что злоумышленник не сумеет помешать работе законных пользователей. В частности, в задачу обеспечения доступности входит исключение возможности атак, вызывающих отказ в обслуживании.

## Контроль доступа

Совместное обеспечение конфиденциальности, доступности и неприкосновенности сводится к контролю доступа. Суть проблемы состоит в обеспечении законным пользователям возможности делать все то, что им дозволено делать и на что остальные не имеют права.

Проблема контроля доступа в действительности намного шире и связана не только с компьютерами. Как вообще можно ограничить доступ к чему-либо? Как можно контролировать доступ к совместно используемым ресурсам? Как обозначить уровни доступа, различные у разных людей? Эту проблему трудно решить даже для большого здания: для этого ставят замки на входе и на дверях внутренних помещений и доверяют ключи от них надежным людям, выдают всем пропуска, которые проверяются охранниками, и т. д. В случае компьютерной системы контроль доступа — тоже трудная задача.

Кроме того, актуальность этой задачи то возрастает, то убывает с течением времени. Сначала вообще не требовался контроль доступа к компьютерам, поскольку все доверяли друг другу. По мере того как все большее количество людей приобщалось к работе с большими вычислительными машинами, возникала необходимость контролировать доступ — как для соблюдения секретности, так и для получения отчетов об использовании машинного времени. Контроль доступа был прост в мире с пакетной обработкой данных.

С появлением персональных компьютеров отпала нужда в контроле: у каждого был свой собственный компьютер. Если кто-то хотел закрыть для других доступ к файлам, он просто запирал свою дверь. В настоящее время происходит возврат к системам коллективного пользования: общим сетевым ресурсам, удаленным системам и т. п. Контроль доступа представляет проблему практически для всех независимо от того, пользуются ли разные люди общим компьютером или одной учетной записью на веб-сайте.

Перед тем как поговорить о различных типах контроля доступа, нам необходимо ввести два понятия. Речь идет о так называемых *субъектах,* у которых есть доступ к неким *объектам.* Часто, хоть и не всегда, субъектом является пользователь, а объектом — компьютерный файл. Субъектом также может быть компьютерная программа или процесс, а объектом — другая компьютерная программа, сопряженная, например. Объектом может быть запись базы данных. Объектом может быть определенный ресурс, возможно, какая-то часть технического оборудования компьютера или принтер, или часть памяти компьютера. В зависимости от обстоятельств одна и та же компьютерная программа бывает субъектом доступа в одном случае и объектом в другом.

Существует два способа задать условия контроля доступа. Вы можете оговорить, что разрешено делать различным субъектам, или оговорить, что позволено сделать с разными объектами. На самом деле — это взгляд на одну и ту же задачу с двух разных сторон, и у этих подходов есть свои плюсы и минусы. Традиционно, операционные системы работали с ресурсами и файлами; таким образом, условия контроля доступа задавали в терминах этих объектов. Современные системы ориентированы на конкретное применение. Они предлагают услуги конечным пользователям, таким как большие системы управления базами данных. В этих системах часто используются те механизмы, которые контролируют доступ субъектов.

Установление доступа не означает «все или ничего»; могут быть различные его виды. Например, в системе UNIX три вида доступа предоставляют следующие права: читать, писать и выполнять. Все эти права независимы. Например, кто-то, обладающий правом только на чтение файла, не может изменять этот файл. Тот, у кого есть право только на ввод информации, может изменять файл, но не вправе его прочесть. Тот, у кого есть полномочия и на чтение, и на ввод информации, волен делать и то и другое.

Третий тип права доступа — «выполнять» — особенно любопытен. Такое право имеет смысл только для компьютерных программ — исполняемых файлов. Субъект, имеющий право только на выполнение определенных файлов, может запустить программу, но ему нельзя ни прочесть код, ни изменить содержимое. При некоторых обстоятельствах это имеет смысл: вообразите программу, хранящуюся в защищенной памяти — устройство цифровой подписи в модуле, снабженном системой защиты от вторжения, — в этом случае действительно возможно выполнение команды без прочтения кода.

Существование различных видов доступа означает, что кто-то имеет возможность решать, кому какие права предоставить. В системе UNIX это владелец файла. Владелец может установить, кому разрешается читать, записывать и выполнять файл. В UNIX принадлежность устанавливается для каждого файла в отдельности и обычно обусловливается каталогом, в котором файл находится.

В системе Windows NT более сложный набор прав доступа. В ней предусмотрены права читать, писать и выполнять, а также удалять, изменять права доступа и изменять принадлежность. Владелец файла может разрешить кому-то изменять права доступа к этому файлу или менять его принадлежность.

Представьте себе существующую в компьютере сложную систему организации доступа в виде таблицы. По вертикали расположен список всех возможных пользователей, по горизонтали — список всех файлов. В ячейках таблицы находятся ус¬

ловия доступа пользователя к соответствующему файлу. Алиса может иметь право чтения файла A, чтения и записи в файл B и вовсе не иметь доступа к файлу C. Для Боба может быть установлена подобная, не менее сложная схема доступа.

В случае компьютерной системы любого разумного размера эта таблица быстро становится очень сложной. Поэтому приходится прибегать к упрощениям. Можно установить доступ к файлу таким образом, чтобы только его владелец имел возможность читать, записывать и выполнять его. Можно сделать файл общедоступным для чтения, но лишь его владелец будет иметь возможность вносить изменения. Можно создать так называемую «группу», в которую входят несколько человек с одинаковым доступом. В этом случае, если люди, например, работают над одним проектом и должны использовать определенные файлы, только они и никто другой будут иметь необходимый доступ. В системе UNIX это легко осуществимо, причем отдельный пользователь может входить в разные группы.

Один из способов справиться со сложностью контроля доступа состоит в том, чтобы разбить таблицу. В некоторых системах список тех, кто имеет доступ к определенному объекту, хранится вместе с самим объектом. Его часто называют *списком контроля доступа* (access control list, ACL). Это обычная практика, и ACL часто используется в целях безопасности операционных систем. Хотя существуют и определенные проблемы. Такие списки работают хорошо в простых средах, когда пользователи сами устанавливают права доступа, но несколько хуже в тех случаях, когда доступ устанавливает управляющий персонал. В таких системах, например, не существует простого способа временной передачи прав доступа. Также подобные системы недостаточно хорошо обеспечивают проверку доступа по ходу работы программы. Кроме того, поскольку установка доступа привязана к объектам, а не субъектам, могут возникнуть трудности, когда понадобится лишить доступа определенного субъекта. Если кто-нибудь из сотрудников компании увольняется, система должна перебрать все объекты и исключить этого человека из каждого списка. Наконец, управление системой на основе ACL довольно трудоемко, поэтому множество предлагаемых программ предназначено для облегчения этой задачи.

## Модели безопасности

Существует множество теоретических моделей безопасности, разработка многих из них финансировалась Министерством обороны в 70-х и 80-х годах. Поскольку речь шла о системах безопасности для нужд обороны, использовалась военная схема секретности, которую мы обсуждали в главе 5. Такие системы называют *многоуровневыми системами безопасности (multilevel security system, MLS),* поскольку они предназначены для поддержки многочисленных уровней секретности в единой системе. (Альтернативные решения слишком громоздкие. Можно создать одну компьютерную систему для несекретных данных, другую, совершенно независимую, — для конфиденциальных данных, третью — для секретных данных и т. д. Или создать систему *наивысшего уровня,* в которой весь компьютер относится к самому высокому уровню секретности.)

Наиболее известна модель Белла-Лападулы — в ней определено большинство понятий, связанных с контролем доступа и описанных в предыдущем разделе.

В этой модели даются определения субъекта, объекта и операции доступа, а также математический аппарат для их описания. Эта теория долгое время оказывала влияние на проектирование систем, однако она не помогла создать практичные и экономичные системы.

Модель Белла-Лападулы предлагает два основных правила безопасности: одно относится к чтению, а другое — к записи данных. Во-первых, если пользователи имеют категорию допуска «Секретно», то они могут читать *несекретные, конфиденциальные* и *секретные* документы, но без права читать *совершенно секретные*. Во-вторых, если пользователи работают с секретными данными, они могут создавать секретные и совершенно секретные документы, но не могут создавать конфиденциальные и несекретные. (Второе условие также важно. Представьте себе, что кто-то — человек или даже компьютерный вирус — пытается украсть документы. Защита, конечно, предотвратит отправку конфиденциальных документов с используемого компьютера. Но если скопировать текст конфиденциального документа в несекретный, то последний можно будет послать по электронной почте. Чтобы этого не случилось, были введены соответствующие средства управления.) Общее правило звучит так: пользователи могут читать только документы, уровень секретности которых не превышает их допуска, и не могут создавать документы ниже уровня своего допуска. То есть теоретически пользователи могут создавать документы, прочесть которые они не имеют права.

Существует понятие *обязательного (мандатного) контроля доступа* (по терминологии Белла-Лападулы), который осуществляется системой. Он отличается от используемого в операционных системах, подобных UNIX или NT, «разграничительного» контроля доступа, который позволяет пользователям самим принимать решение о том, кто и с каким файлом может работать. (Впрочем, большинство версий UNIX могут иметь некоторые элементы обязательного контроля доступа: обладатель корневого доступа имеет обязательный доступ на чтение, запись и выполнение всех файлов компьютера.)

Модель Белла-Лападулы имеет большое значение, но у нее есть ряд ограничений. Во-первых, эта модель ориентирована на обеспечение конфиденциальности в ущерб всему остальному, а принципы конфиденциальности основаны на военной схеме секретности. Во-вторых, игнорируется проблема изменения классификации. В модели предполагается, что все сведения каким-то волшебным образом относятся к соответствующему уровню секретности, который остается неизменным. В реальной жизни все меняется: кто-то засекречивает важную по его мнению информацию, а кто-то другой впоследствии рассекречивает ее. Бывает так, что совокупность данных имеет более высокую секретность, нежели каждый элемент данных по отдельности: номера телефонов Агентства национальной безопасности относятся к несекретным данным, но полная телефонная книга АНБ классифицируется как конфиденциальная информация. Это означает, что уровень секретности сведений легко повышается сам собой, а обратный процесс возможен только после тщательной проверки. И в-третьих, бывают случаи, когда пользователи должны работать с данными, которые они не имеют права увидеть. Сведения о том, что самолет несет груз из некоторого количества бомб, возможно, имеют более высокий уровень секретности, чем уровень доступа диспетчера, но диспетчеру тем не менее необходимо знать вес груза.

В теоретической литературе обсуждались и многие другие модели безопасности. В модели под названием «Китайская стена», например, подробно рассматриваются компьютерные системы, которые работают с данными, полученными от не доверяющих друг другу пользователей, и способы, позволяющие гарантировать каждому из них конфиденциальность. (Вообразите компьютеризованную брокерскую систему, клиенты которой имеют доступ к своим счетам. Брокеры хотят исключить возможность, чтобы клиент А увидел портфель клиента Б, даже несмотря на то, что, возможно, оба портфеля классифицируются одинаково.)

Модель Кларка-Уилсона была разработана скорее для коммерческих нужд, нежели для военных структур. Требования коммерческой безопасности преимущественно касаются целостности данных, и именно на нее ориентируется эта модель. Кларк и Уилсон дали определение двум типам целостности: внутреннему соответствию, которое относится к свойствам внутреннего состояния системы, и внешнему соответствию, которое касается свойств системы по отношению к внешнему миру. Затем была построена формальная модель безопасности, в которой были систематизированы эти принципы, так же как и принципы обеспечения конфиденциальности.

В модели Кларка-Уилсона центральным является понятие данных, с которыми позволено оперировать только предписанным способом. Например, при помощи этой модели можно реализовать потребности двойной бухгалтерии: каждый кредит необходимо сопоставить равному дебету, и все должно быть записано в специальный аудиторский файл. Эта модель запрещает производить определенные действия без дополнения их другим соответствующим действием: например, запрещено кредитовать счет без записи дебета.

## Ядра безопасности и надежная вычислительная база

Многие операционные системы имеют встроенные средства безопасности. В этом есть здравый смысл — часто лучше всего поместить средства безопасности на нижних уровнях системы: на аппаратном или уровне операционной системы. Тому есть несколько причин.

Во-первых, часто существует возможность обойти средства безопасности на некотором уровне посредством атаки, проведенной уровнем ниже. Например, встроенные функции кодирования в текстовом редакторе не зависят от того, может ли злоумышленник взломать операционную систему, под управлением которой он работает. Таким образом, более надежной является защита на самом низком уровне программного обеспечения.

Во-вторых, так проще. В ядро системы обычно легче ввести дополнительные меры безопасности. Упрощается осуществление и анализ таких мер. И, как следовало ожидать, в результате получается более защищенная система.

В-третьих, часто так получается быстрее. Все средства работают лучше, если они встроены в операционную систему, и средства безопасности не исключение. Криптография, например, может съедать много времени, и имеет смысл сделать ее работу как можно более эффективной.

Поэтому безопасность операционных систем остается предметом исследований уже в течение десятилетий. Раз так, для нее разработан свой собственный набор понятий.

- **Монитор обращений.** Часть программных средств, которая осуществляет доступ субъектов к объектам. Когда некий процесс делает вызов операционной системы, монитор обращений останавливает процесс и выясняет, следует ли разрешить или запретить вызов. Например, он не позволит пользователю с конфиденциальной регистрационной учетной записью читать секретные документы или создавать несекретные документы.
- **Надежная вычислительная база.** Это все защитные устройства внутри компьютера — оборудование, программно-аппаратные средства, операционная система, программные приложения и т. д. — все, что используется для осуществления политики безопасности. Некий администратор указывает компьютеру, что, от кого и каким образом следует защищать (это и есть политика безопасности), а надежная вычислительная база обеспечивает выполнение этой задачи.
- **Ядро безопасности.** Это оборудование, программно-аппаратные средства, операционная система, программные приложения и все остальные элементы надежной вычислительной базы, которые реализуют концепцию монитора обращений.

Монитор обращений — это абстрактное устройство защиты; оно занимается такими вещами, как управление файлами и управление памятью. Ядро безопасности обеспечивает действие монитора обращений. Надежная вычислительная база содержит все средства защиты, в том числе и ядро безопасности. А все в целом реализует некую модель безопасности — Белла-Лападулы или какую-то другую — и осуществляет защиту наиболее простыми и эффективными средствами. И конечно, надежная вычислительная база по определению надежна — пользователи не должны иметь возможность изменить ее, иначе безопасность может быть утрачена.

Эту концепцию трудно осуществить в реальной операционной системе. Компьютер — сложный зверь, и все в нем должно быть надежным. Любая мелочь способна испортить все дело. Если кто угодно имеет доступ к жесткому диску с правом на чтение и запись, то как можно помешать одному пользователю читать то, что пишет другой? Что, если один пользователь хочет, чтобы второй прочитал, что он написал? Возможно ли, чтобы пользователь, используя прерывания, делал что-то, чего он делать не должен? Как защитить доступ к принтеру? Может ли один человек узнать секреты другого через клавиатуру? Что, если базовые средства надежной вычислительной базы выйдут из строя? Как вам удастся выполнить дефрагментацию диска, если у вас есть доступ только к своим файлам?

Исторический пример почти правильной реализации этой теории — это операционная система под названием Multics, которую в конце 1960-х разработали MIT, Bell Labs и Honeywell. В Multics модель Белла-Лападулы построена с нуля. (Фактически, именно проект Multics дал толчок развитию модели Белла-Лападулы.) Разработчики применяли формальную математическую систему этой модели, чтобы продемонстрировать безопасность своей системы, а затем обозначили понятия модели в своей операционной системе. Ни одного кода не было написано до того,

как спецификации были одобрены. Multics работала, хотя средства безопасности в ней были слишком громоздкими. К настоящему времени почти все уже забыли о Multics и уроках, вынесенных из этого проекта.

Один из уроков, о которых люди позабыли, — в том, что ядро должно быть простым. (Даже ядро Multics, которое содержало только 56 000 кодовых строк, как выяснилось, слишком сложно.) Ядро — это по определению высоконадежные программы. В главе 13 будет рассказано о надежности программ, мораль же в том, что неразумно ждать, что в программах не будет сбоев защиты. Поэтому чем проще программа, тем меньше в ней будет ошибок безопасности.

К сожалению, современные операционные системы страдают болезнью, известной как «распухание ядра». Это означает, что большой объем кода располагают внутри ядра, а не снаружи. Когда система UNIX была написана впервые, считалось обязательным помещение несущественных кодов за пределами ядра. С тех пор все забыли этот урок. Все имеющиеся сейчас разновидности UNIX в той или иной степени страдают распуханием ядра: у них либо слишком много команд в ядре, либо имеются непонятные утилиты, запускающиеся в случае корневого доступа, либо что-то еще.

Windows NT устроена намного хуже. Эта операционная система может служить примером того, как полностью игнорируются исторические уроки безопасности. То, что находится в ядре, по определению защищено, поэтому с точки зрения разумного проектирования необходимо уменьшить размер ядра, насколько возможно, и убедиться, что оно полностью защищено. Windows, похоже, придерживается мнения, что, поскольку то, что содержится в ядре, защищено по определению, следует просто побольше всего разместить прямо в ядре. Если разработчикам было непонятно, как обеспечить безопасность чего-либо, они просто запихивали это в ядро и считали его уже защищенным. Очевидно, это не помогает при длительной работе.

В системе Windows драйверы принтеров являются частью ядра. Пользователи регулярно загружают эти драйверы или устанавливают их, наверное, не осознавая, что норовистый (или неисправный) драйвер принтера может полностью уничтожить защиту их системы. Было бы намного разумнее разместить драйверы принтеров снаружи ядра, тогда они не должны были бы быть надежными, но это вызвало бы у разработчиков больше трудностей. А философия Windows NT в том, чтобы всегда отдавать предпочтение простому над безопасным — и в обращении, и в разработке.

Windows 2000 еще хуже.

## Тайные каналы

*Тайные каналы* — это головная боль разработчиков моделей защиты. Помните, что одно из двух основных правил безопасности в том, что пользователь или процесс не может записывать данные на более низком уровне доступа? Тайные каналы — это способ обойти этот контроль.

Тайные каналы — это способ для субъекта с доступом более высокого уровня послать сообщение на более низкий уровень защиты — обычно посредством каких-

то ресурсов совместного использования. Например, злоумышленная программа, уровень доступа которой «Совершенно секретно», могла бы послать, манипулируя сетевой пакетной передачей (два пакета подряд означают единицу, а два пакета с промежутком между ними — нуль), сообщение, в котором сообщались бы коэффициент загрузки ЦПУ, распределение памяти, доступ к жесткому диску, установка очередности печати и т. п. Тайным каналом могут быть и пробелы в документе, и «случайное» заполнение в конце записи базы данных. Не быстро, но вполне осуществимо отправить сообщения процессов высокого уровня доступа процессам более низкого уровня, разрушая модель безопасности.

Создавать тайные каналы просто и забавно. Опасность представляют не те пользователи, которые копируют с экрана данные с грифом «Совершенно секретно» и переправляют их в Китай, а те, кто пишет программы, которые могут тайно собирать данные, оставаясь в тени.

Разработчики систем тратят много времени на то, чтобы закрыть определенные тайные каналы или, по крайней мере, свести к минимуму количество информации, которую можно было бы по ним переслать. Циклы ЦПУ могут иметь фиксированные такты специально для того, чтобы сделать невозможной утечку по отдельным тайным каналам. К программам пакетной передачи может быть добавлена система случайного шума, чтобы существенно снизить использование этого тайного канала. Но фактически невозможно перекрыть все тайные каналы, и часто пытаются выйти из положения, жестко ограничивая пропускную способность. Тем не менее, если интересующая вас информация — это всего-навсего крошечный 128-битовый криптографический ключ, вы найдете тайный канал, по которому ее можно отправить.

## Критерии оценки

Если вы собираетесь приобрести компьютерную систему с определенной моделью безопасности или с определенным типом ядра, вам потребуется гарантия, что эта модель надежна. Или, другими словами, некая гарантия, что система обеспечивает достаточную защиту.

Есть два основных способа получить такую гарантию. Первый — это IVV, что означает «независимые верификация и проверка достоверности» (independent verification and validation). Основная его идея в том, что один коллектив разрабатывает и создает систему, а другой — оценивает эту разработку, вплоть до того, что иногда создает идентичную систему, чтобы сравнивать с ней оцениваемую. Это дорогой путь, его применяют в таких важных случаях, как создание системы управления ядерным оружием или компьютеров космических кораблей многоразового использования.

Более дешевый путь — оценивать систему по какому-то независимому набору критериев и присваивать ей определенный рейтинг безопасности.

Первым набором критериев оценки систем была «Оранжевая книга». В какой-то мере она уже устарела, но все же оказала большое влияние на компьютерную безопасность в 80-х годах, и до сих пор можно услышать, как перебрасываются терминами из «Оранжевой книги» — вроде «уровня безопасности C2».

На самом деле «Оранжевая книга» называется так: «Критерии оценки надежности компьютерных систем Министерства обороны США» (U.S. Department of Defense *Trusted Computer System Evaluation Criteria),* но это название трудно выговорить, а обложка у книги оранжевая. Она была опубликована в 1985 году Национальным центром компьютерной безопасности, который в некоторой степени можно считать подразделением Агентства национальной безопасности. Задачей «Оранжевой книги» было определить требования безопасности и стандартизировать правительственные требования поставок. Она дала фирмам-изготовителям вычислительных машин возможность измерить безопасность своих систем и подсказала им, что они должны ввести в свои защищенные программы. Кроме того, она предложила систему классификации различных уровней компьютерной безопасности и способы проверить, удовлетворяет ли определенная система заданному уровню.

Эта классификация выглядит так: D (минимальная секретность), C (защита по усмотрению), B (обязательная защита) и A (подтвержденный доступ). Внутри некоторых этих уровней существуют подуровни. Есть, например, C1 и C2 — защита по усмотрению и защита с контролируемым доступом, последняя более эффективная. C1 не является защищенным уровнем; это, по сути, то, что вы получаете с новеньким с иголочки UNIX'OM. (Вы ведь не замечали большого количества систем, которые гордились бы своим рейтингом безопасности C1.) C2 лучше; это, наверное, наиболее подходящий уровень безопасности для коммерческих продуктов[1]. В основе большинства процедур контроля доступа лежит модель Белла-Лападу¬лы, которая берет начало с уровня B1 — уровни B1, B2, B3 и A, как считали, больше подходят для военных систем.

Основная проблема такого подхода — с уровнями безопасности — была в том, что они не означали, что система защищена. Приобретение системы уровня B1, например, не давало гарантированной безопасности компьютера. Это всего лишь означало, что изготовители установили в систему обязательный контроль доступа и имели необходимую документацию, чтобы получить рейтинг безопасности B1. Безусловно, обязательный контроль доступа делает систему уровня B1 намного более защищенной, чем система уровня C2, но ошибки защиты одинаково вероятны в любой системе. Единственное, что было доподлинно известно, — в первом случае разработчики больше старались.

Кроме того, «Оранжевая книга» применима только к автономным системам и полностью игнорирует возможное подсоединение компьютера к сети. Несколько лет назад фирма Microsoft предпринимала большие усилия, чтобы присвоить Windows NT рейтинг безопасности C2. Усилия пошли на убыль, когда стало известно, что об этом рейтинге можно говорить, только если компьютер не подсоединен к сети, у него вовсе нет сетевой карты, дисковод заклеен эпоксидной смолой, а процессор — Compaq 386. Рейтинг C2 системы Solaris столь же необоснован. В не-

[1] Класс C1 называется Discretionary Security Protection и подразумевает, что пользователи сами решают, кому предоставлять конфиденциальную информацию, а кому — нет. Контроль обеспечивается самими пользователями. Второй класс — C2 — называется Controlled Access Protection. Для него в силе требования C1. Но за контроль предоставления или ограничения доступа к данным отвечает система. Разрешение доступа к объекту может быть дано только авторизованными пользователями. — *Примеч. ред.*

давних модификациях «Оранжевой книги» с переменным успехом делались попытки иметь дело с компьютерами, соединенными с сетью.

Кроме того, как известно, рейтинг имеет узкое значение. Рейтинг системы относится только к определенной конфигурации и определенному типу установленного программного обеспечения. Если версия 1.0 операционной системы имела определенный уровень безопасности, нет никакой гарантии, что версия 1.1 имеет такой же уровень. Если рейтинг компьютерной безопасности относится к определенной конфигурации — с определенным набором установленного программного обеспечения, — то нельзя ничего сказать о безопасности компьютеров других конфигураций.

В сегодняшнем мире, где всегда все взаимосвязано, «Оранжевая книга» вышла из употребления. Некоторые национальные и международные организации делали попытки модернизировать ее. Канадцы создали свои «Канадские критерии оценки надежности компьютерных продуктов» (Canadian Trusted Computer Products Evaluation Criteria). ЕС ответил «Критериями оценки безопасности информационных технологий» (Information Technology Security Evaluation Criteria, ITSEC), предварительно одобренными в 1995 году. Еще одно предложение США было названо «Федеральными критериями».

Недавно все собрались вместе, чтобы прекратить это сумасшествие. Были разработаны «Общие критерии» с целью удовлетворить всех и объединить хорошие идеи из всех остальных документов. Появился стандарт ISO (International Organization for Standardization, Международная организация по стандартизации) (15408, версия 2.1). Основная его мысль в том, что «Общие критерии» предоставляют свод концепций безопасности, которые пользователи могут включать в *профиль защиты,* представляющий собой, по сути, формализованные потребности пользователя в отношении безопасности. Кроме того, каждый отдельный продукт можно проверить соответственно профилю защиты. Предполагается, что правительство станет следить за тем, чтобы методология «Общих критериев» выполнялась как надо, а коммерческие лаборатории будут осуществлять фактическое тестирование и сертификацию.

«Общие критерии» основаны на соглашении о взаимном признании сертификатов качества, то есть различные страны договорились признавать проведенную друг другом сертификацию. В соглашении участвуют Австралия, Канада, Франция, Германия, Новая Зеландия, Великобритания и Соединенные Штаты.

Это огромный шаг в правильном направлении. «Общие критерии» разработаны, чтобы дать общую оценку безопасности (не работы) программ, которые можно приобрести, по ряду различных требований. Индустрия смарт-карт тратит много времени на разработку своего профиля безопасности в рамках «Общих критериев». Я связываю с этой программой большие надежды.

## Будущее безопасных компьютеров

Формальные модели хороши для блестящей теории, но менее полезны на практике. У них есть принципиальные ограничения; нет гарантии, что, реализовав модель безопасности, вы обязательно получите систему, имеющую определенные

свойства защиты. Модели могут привести к системе, непригодной для использования; принудительная подгонка системы под модель способна выработать чересчур замысловатые конструкции. Может потребоваться уйма времени, чтобы разработать и создать их. И хуже того, они даже не гарантируют безопасности. Если система соответствует требованиям формальной модели безопасности, в лучшем случае она в силах гарантировать защиту от злоумышленников, которые придерживаются этой модели. Ушлые взломщики изобретательны и всегда выдумывают что-нибудь новенькое. И в результате, поскольку злоумышленники не придерживаются моделей разработчиков, они снова и снова взламывают защиту.

Из тех систем, которые сейчас находятся в употреблении, практически ни одна не создана по формальной модели безопасности. Системы используют некоторые формальные идеи безопасности — так, все операционные системы имеют надежную вычислительную базу, — но для того, чтобы системы оставались полезными и пригодными, приходится искать компромисс. Только это и имеет смысл.

Безопасность операционной системы, а значит, и защита компьютера имеет несколько ключевых компонентов. Один из них — это жесткий мандатный механизм безопасности более общего типа, чем в формальных моделях. Этот мандатный механизм безопасности реализует политику администратора, который не обязательно является пользователем. Более того, стратегия политики безопасности заключается также в контроле доступа и шифрования. То есть политика должна устанавливать, кто (человек или процесс) и к каким данным (или другим процессам) может иметь доступ и какие средства управления шифрованием должны быть применены к этим данным. Такого рода стратегия (как и ничто другое) не может перекрыть тайные каналы, но полезна для прекращения тех злоупотреблений, с которыми мы имеем дело сегодня.

Вторым ключевым компонентом безопасной операционной системы является *выверенный канал*. Это механизм, с помощью которого пользователь (или процесс) напрямую взаимодействует с надежной вычислительной базой, который может быть задействован пользователем или проверенными программами и который не может быть подменен посторонним программным обеспечением. Например, разве это не здорово, если пользователь видит экран входа в систему и уверен, что это настоящий экран входа, а не троянский конь, пытающийся выудить его пароль? Механизмы реализации выверенного канала также будут иметь большое значение для снижения ущерба, ожидаемого от программ злоумышленников.

На рынке существуют надежные операционные системы, которые снабжены этими компонентами, но они все еще мало известны потребителям. Мне хотелось бы увидеть, как большинство этих идей реализуются в основных операционных системах, таких как Microsoft Windows. Но не похоже, что это случится в ближайшее время.

# Глава 9. Идентификация и аутентификация

Независимо от того, какую систему защиты вы используете, чаще всего первым шагом работы является *идентификация* и подтверждение подлинности *(аутентификация):* кто вы такой и можете ли это доказать? Как только компьютер узнает вас, он сможет выяснить, что вам разрешено и чего не позволено делать. Другими словами, контроль доступа не может начаться прежде, чем завершатся идентификация и аутентификация.

Давайте поговорим об этом. У Алисы есть некие возможности для работы на компьютере, и мы хотим быть уверены в том, что эти возможности есть только у нее. Иногда это — возможность получить доступ к какой-либо информации: файлам, балансу счетов и т. д. Иногда — получение доступа ко всему компьютеру: никто другой не сможет включить его и воспользоваться данными Алисы и ее программами. В других случаях возможность носит более конкретный характер: получить деньги из банкомата, воспользоваться сотовым телефоном, отключить охранную сигнализацию. Эта возможность может быть связана с веб-сайтом: например, доступ к странице Алисы или банковским документам. Иногда эта возможность — доступ к шифровальному ключу, который слишком длинен для запоминания. (Система PGP — набор алгоритмов и программ для высоконадежного шифрования, применяет контроль доступа для защиты частных ключей.) Не важно, что из перечисленного мы рассматриваем, — важно то, что некоторые меры контроля доступа требуют идентификации Алисы.

На самом деле меры контроля доступа должны обеспечить две вещи. Во-первых, Алиса должна попасть в систему, а во-вторых, система должна оставить других снаружи. Сделать только первое или только второе легко — открытая дверь позволит и Алисе, и кому-нибудь другому войти; а заложенная кирпичом дверь будет держать снаружи как остальных, так и Алису, — но выполнить оба условия сразу сложнее. Нам нужно что-нибудь, что даст возможность узнавать Алису и пускать ее внутрь, но так, чтобы другие не смогли в это повторить. Мы должны уметь идентифицировать Алису, а после этого проверить подлинность идентификации. (На самом деле меры контроля доступа обязаны осуществлять еще и третью функцию — протоколирование всего, что происходило.)

Традиционно опознавательные и проверочные меры основываются на чем-то одном из трех: «что вы знаете», «кто вы такой» и «что вам позволено». Это реализуется в виде паролей, биометрических методов распознавания и опознавательных знаков доступа. Иногда системы используют совместно любые две из этих вещей. Параноидальные системы используют все три.

# Пароли

Традиционным подходом к проверке подлинности является применение пароля. Вы наблюдаете повсеместно. Когда вы регистрируетесь в компьютерной системе, то вводите имя пользователя и пароль. Чтобы сделать звонок по телефону с использованием телефонной карты, вам необходимо набрать номер своего счета и пароль. Чтобы получить деньги в банкомате, вы вставляете свою карту и набираете идентификационный номер (пароль).

Два шага, применяемые в каждом из этих примеров, отражены в названии данной главы. Первый шаг называется идентификацией (опознаванием): вы сообщаете компьютеру, кто вы (имя пользователя). Второй шаг называется аутентификацией (подтверждением подлинности): вы доказываете компьютеру, что вы именно тот, кем себя назвали (пароль).

Компьютер, который вас опознает, имеет список имен пользователей и паролей. Когда вы вошли под своим именем пользователя и паролем (или номером вашего счета и идентификационным номером), компьютер сличил введенные данные с записями, хранящимися в его списке. Если вы ввели имя пользователя, имеющееся в списке, и пароль, соответствующий этой записи, то попадете внутрь. Если нет, вы останетесь вне системы. Иногда система будет повторно спрашивать вас об имени пользователя и пароле. Иногда она будет заблокирована после определенного числа неудачных попыток. (Вы ведь не хотите, чтобы кто-нибудь, укравший карточку банкомата, затем пробовал перебрать все десятки тысяч возможных идентификационных номеров, один за другим, в попытке найти единственный подходящий.)

К несчастью, система имен пользователей и паролей работает не столь хорошо, как предполагают люди.

Понятие паролей, вообще говоря, основывается на попытке совместить несовместимое. Идея в том, чтобы набрать легкую для запоминания случайную последовательность. К сожалению, если нечто легко запомнить, то оно не будет случайным, например «Сюзанна». А если это будет набрано наугад, например «r7U2*Qnpi», то оно запоминается нелегко.

В главе 7, когда я говорил о длине ключей и безопасности, я обсуждал проблемы изобретения и запоминания пользователями ключей. Пароль — это форма запоминания пользователем ключа, и словарные нападения на пароли поразительно эффективны.

Как работает это нападение? Подумаем о системе контроля доступа к компьютеру или веб-сайту. У компьютера есть файл имен пользователей и паролей. Если нападающий получит доступ к этому файлу, то узнает все пароли. В середине 70-х годов эксперты по компьютерной безопасности пришли к лучшему решению: вместо хранения всех паролей в файле они решили хранить хэш-функцию пароля. Теперь, когда Алиса набирает свой пароль в компьютере или на веб-сайте, программное обеспечение вычисляет хэш-значение и сравнивает его с сохраненным в файле. Если они совпадают, Алиса допускается в систему. Теперь нет файла паролей, который можно было бы украсть, — есть только файл хэшированных паролей. И так как назначение хэш-функции — воспрепятствовать незваному гостю зайти слишком далеко в своих намерениях, злоумышленник не сможет восстановить настоящие пароли из хэшированных.

И тут ему на помощь приходят словарные нападения. Предположим, что нападающий владеет копией файла хэшированных паролей. Он берет словарь и подсчитывает хэш-значение каждого слова в словаре. Если хэш-значение какого-либо слова соответствует одной из записей файла, тогда он нашел пароль. Он попробует перебрать таким образом все слова, попытается переставлять буквы, набирать некоторые буквы прописными и т. п. В конце концов он попробует все характерные комбинации короче заданной длины.

Ранее *словарные нападения* были сложны из-за медленной работы компьютеров. Они стали более легкими, потому что компьютеры стали гораздо быстрее. L0phtcrack является примером хакерского инструмента, предназначенного для восстановления паролей и оптимизированного для паролей Windows NT. Windows NT имеет две парольные защиты: более сильную, предназначенную для NT, и более слабую, совместимую со старыми сетевыми протоколами входа в систему. Эта функция работает без учета регистра с паролями не длиннее семи символов. L0phtcrack облегчает работу в парольном пространстве. На Pentium II с тактовой частотой 400 МГц L0phtcrack может перебрать каждый буквенно-цифровой пароль за 5,5 часа, каждый буквенно-цифровой пароль с прочими символами за 45 часов и каждый из возможных паролей, включающий любой знак клавиатуры компьютера, — за 480 часов. Это не сулит ничего хорошего.

Некоторые пытались решить эту проблему, используя все более и более «сильные» пароли. Это означает, что пароли сложней для угадывания и их появление в словаре менее вероятно. Старая универсальная система контроля доступа на мэйнфрэймах (RACF) требовала от пользователей ежемесячной смены паролей и не разрешала использовать слова. (В Microsoft Windows нет такого контроля, и вам услужливо предлагается сохранить любой пароль.) Некоторые системы создают пароли для пользователей случайным образом — путем связывания случайных слогов в произносимое слово (например, «талпудмокс») или соединения чисел, символов и смены регистра: например FOT78hif#elf. Система PGP использует парольные фразы, которые представляют собой сложные предложения с бессмысленным контекстом: например «Телефон333333, это должно быть вы говорите мне приятным голосом 1958???!телефон». (Однако это не так просто для запоминания и набора, как вам хотелось бы.)

Эти ухищрения становятся все менее и менее эффективными. В течение последних десятилетий действие закона Мура делает возможной «атаку в лоб» для ключей, имеющих все большую энтропию. В то же самое время есть максимум энтропии, до которого средний компьютерный пользователь (или даже пользователь уровнем выше среднего) может что-то запомнить. Вы не можете ожидать от него запоминания 32-символьного случайного шестнадцатеричного числа, так что же должно произойти, чтобы он запомнил 128-битовый ключ? Вам действительно не стоит полагаться, что он введет при входе в систему фразу с использованием набора алгоритмов и программ высоконадежного шифрования, описанных в предыдущем параграфе. Эти два фактора пересеклись; сейчас взломщики паролей могут вычислить все, что (в пределах разумного) может запомнить пользователь.

Конечно, есть исключения. Производство высоконадежных компьютеров, применяемых в ядерной отрасли, надежные дипломатические каналы, системы, применяемые для связи со шпионами, живущими на вражеской территории, — случаи, когда

пользователи найдут время, чтобы запомнить длинные и сложные парольные фразы. Эти применения не имеют ничего общего с современными компьютерными сетями и паролями для продажи товаров в электронной торговле. Проблема в том, что средний пользователь не может и даже не пытается запомнить достаточно сложные пароли для предотвращения словарных нападений. Атаковать защищенную паролем систему часто легче, чем напасть на шифровальный алгоритм с 40-битовым ключом. Пароли ненадежны, если вы не в силах предотвратить словарных нападений.

Как ни плохи пароли, люди находят способ сделать их еще хуже. Если вы попросите их выбрать пароль, он будет паршивый. Если вы принудите их выбрать хороший пароль, они напишут его на почтовой карточке и прикрепят к монитору. Если вы попросите поменять его, они сменят его на пароль, которым пользовались месяц назад. Только изучение действующих паролей обнаруживает, что в 16% из них насчитывается три и менее цифры и 86% могут быть легко взломаны. Многочисленные исследования лишь подтверждают данную статистику[1].

Одни и те же люди выбирают одни и те же пароли для множества приложений. Хотите украсть группу паролей? Создайте веб-сайт, содержащий какую-либо интересную информацию: порно, результаты хоккейных турниров, сведения об акциях или все, что касается демографии. Не делайте платным его посещение, а введите регистрацию имен пользователей и паролей для просмотра информации. В большинстве случаев вы будете получать те же самые имя пользователя и пароль, которые пользователь использует в последнее время. Может быть, они позволят вам войти в его банковский счет. Сохраняйте и неправильные пароли; иногда люди по ошибке вводят пароль, предназначенный для системы A, в систему B. Заставьте пользователя заполнить небольшую анкету при регистрации: «Какие другие системы вы используете регулярно? Банк X, брокерскую фирму Y, службу новостей Z?» Я знаю, что один исследователь сделал нечто подобное в 1985 году — он получил дюжины паролей системных администраторов.

И даже когда люди выбирают хорошие пароли и меняют их регулярно, они слишком часто хотят поделиться ими с другими, состоящими и не состоящими в организации, особенно когда им нужна помощь, чтобы справиться с работой. Ясно, что такие откровения несут величайший риск для безопасности, но в сознании людей риск представляется минимальным, а потребность выполнить работу преобладает.

Это не говорит о том, что нет лучших или худших паролей. Предшествующий пример парольной фразы из PGP все еще защищен от словарных нападений. В целом, чем проще пароль для запоминания, тем он хуже. Вообще словарные нападения пытаются сначала разгадать заурядные пароли: словарные слова, перевернутые слова, слова с некоторыми прописными буквами, их же с незначительными изменениями — как, например, с числом 1 вместо буквы I, и т. п.

К сожалению, многие системы ненадежны так же, как и самые слабые пароли. Когда нападающий хочет получить вход в систему, его не волнует, чей он получает доступ. Согласно рабочим тестам, L0phtcrack восстанавливает около 90% всех па-

[1] Червь Морриса — новаторство не только в области механизма распространения. Он попутно проводил словарную атаку на предмет вскрытия паролей пользователей, руководствуясь простыми соображениями: проверял в этом качестве входные имена пользователя на верхнем и нижнем регистрах, в зеркальном отображении и т. д., прибегая как к подручному средству к встроенному в UNIX корректору Unispell, который включает и файл словаря. — *Примеч. ред.*

ролей менее чем за день и 20% всех паролей в течение нескольких минут. Если на 1000 входов 999 пользователей выберут исключительно сложные пароли, такие что L0phtcrack не сможет установить их, программа взломает систему, подобрав единственный слабый пароль.

С другой стороны, с точки зрения пользователя это может быть примером того, что «нет необходимости бегать быстрее медведя — достаточно опережать тех, кто рядом с вами». Любое словарное нападение будет успешным против тех многих входов, чей пароль «Сюзанна» — они-то в первую очередь и станут жертвой атаки. Если же ваш пароль «молот-бабочка», то, хотя он тоже достаточно уязвим для словарных нападений, вероятно, не он станет жертвой.

Принимая в расчет вероятный тип нападающего, вы можете сделать систему с длинными и сильными паролями надежной. Но все постоянно меняется: закон Мура гласит, что сегодняшние сильные пароли — это завтрашние слабые пароли. В общем, если система основана на паролях, нападающий может организовывать словарное нападение в ожидании времени, когда система станет уязвимой. Периодически.

Подведем итоги. Все основывалось на нападающем, захватившем файл хэширо¬ ванных паролей. Стоит предотвратить словарные нападения, и пароли снова станут пригодными. Это возможно, хотя и нелегко для компьютеров с общим доступом. Парольный файл UNIX, например, может читать кто угодно. В наши дни в UNIX есть *теневой парольный файл;* в нем находятся действительные хэшированные пароли, а в общедоступном парольном файле не содержится ничего полезного. Файл хэширо¬ ванных паролей в NT хорошо защищен, и его трудно украсть; для этого вам нужен или доступ администратора, чтобы разыскать хэшированные пароли через сеть (хотя поздние версии NT и Windows 2000 предотвращают и это), или же вам нужно отлавливать пароли, когда они используются для других сетевых приложений.

Система также может «захлопываться» после нескольких попыток неудачного ввода пароля, например десяти. Таким образом, если кто-то пытается войти под именем Алисы и начать угадывать пароли, он введет только 10 вариантов. Конечно, это будет надоедать Алисе, но это лучше, чем подвергать риску ее имя пользователя. И точное определение времени «замораживания» может зависеть от обстоятельств. Может быть, ее вход будет закрыт на 5 минут или на 24 часа. Может быть, до тех пор пока она не поговорит с каким-нибудь администратором. Высоконадежные механизмы после определенного числа попыток неудачного ввода пароля или его неправильного набора могут замораживаться надолго, с уничтожением информации внутри.

Другое решение состоит в том, чтобы использовать интерфейс, несовместимый с компьютером. Ваша магнитная карта, по которой вы вправе получить наличные деньги, защищена четырехзначным идентификационным номером. Что может быть более незначительным для компьютерного взлома? Требуется несколько миллисекунд, чтобы перебрать все возможные 10 000 идентификационных номеров, но в данном случае компьютер сложно присоединить к интерфейсу пользователя. Человек может стоять у банкомата и перебирать все эти номера один за другим. И для того, чтобы проверить все 10 000 идентификационных номеров, может потребоваться вместо 10 секунд — 28 часов безостановочной работы.

Так же как люди способны быть достаточно отчаянными, чтобы постараться осуществить такое нападение, так и банкомат будет «глотать» карточки, если вы

вводите слишком много неверных паролей. До сих пор эта мера безопасности все еще применяется во многих системах: кодах дезактивации сигнализации (конечно, вы можете попытаться перебрать 10 000 возможных кодов, но на это у вас есть всего лишь 30 секунд), электронных дверных замках, телефонных карточках и т. п. Эти системы работают потому, что здесь нападение не может быть автоматизировано; но если вы сумеете использовать компьютер для перебора всех идентификационных номеров (или паролей) данных систем, вы взломаете эти системы.

Большинство системных проектировщиков не осознают разницы между системой с ручным интерфейсом, которая может быть надежна с четырехзначным личным идентификационным кодом, и системами, имеющими компьютерный интерфейс. Это та причина, по которой мы видим слабые, подобные идентификационному коду, пароли в очень многих веб-системах (включая некоторые брокерские сайты).

Что все-таки делать, если вы не можете предотвратить словарные нападения? Один из приемов — найти более объемный словарь. Другой — прибавить случайные числа к паролям (как говорят, «посолить»). В работе должно быть несколько различных типов визуальных и графических паролей; идея состоит в том, что чем больше возможных паролей, тем, следовательно, сложнее устроить словарное нападение. Однако все это ограничено памятью пользователя.

Пароли — это то, что знает пользователь. Другие техники проверки подлинности базируются на том, кем является пользователь, — на биометрических данных, и на том, что пользователь имеет, — на опознавательных знаках доступа.

## Биометрические данные

Идея проста: вы сами подтверждаете свою подлинность. Ваш «отпечаток голоса» отопрет дверь в вашем доме. Сканирование сетчатки глаза пустит вас в офис. Отпечаток большого пальца зарегистрирует вас в вашем компьютере. Это использовалось даже в фильме «Звездный путь»: капитан Пикард «подписывает» электронный бортовой журнал отпечатком своего большого пальца.

Биометрические данные — самые старая из форм опознавания. Физическое узнавание является биометрикой; наши предки использовали его задолго до того, как они эволюционировали в людей. Коты метят свою территорию. Дельфины издают индивидуальные, как подпись человека, звуки.

Биометрические данные также используются для опознавания в системах связи. Если вы разговариваете по телефону, человек на другом конце провода идентифицирует вас по голосу. Ваша подпись в контракте идентифицирует вас как лицо, подписавшее его. Ваша фотография идентифицирует вас как лицо, на имя которого выдан именно этот паспорт.

Для большинства методов биометрические данные нужно сохранять в базе данных, как и пароли. Голос Алисы будет служить биометрическим опознавательным знаком в разговоре по телефону, если вы Алису уже знаете. Если она незнакомка, то вам это не поможет. Точно так же и с почерком Алисы — вы можете узнать его, только если уже видели. Для разрешения этой проблемы карточки с подписями хранятся в банках в картотеке. Алиса пишет свое имя на карточке, когда открывает свой счет; эта карточка хранится в банке. Когда Алиса подписывает чек, банк сопо¬

ставляет ее подпись с той, что хранится в картотеке, для того чтобы убедиться, что чек имеет силу. (На практике это случается редко. Проверить письменную подпись так долго, что банк не побеспокоится сделать это за сумму, меньшую 1000 долларов. Он предполагает, что если возникнут проблемы, то кто-нибудь пожалуется. И разобраться с редкой проблемой дешевле, чем платить кому-либо за постоянную проверку.) Вы можете точно так же поступить и с Алисиным голосом — сравнив его с аналогичным образцом, хранящимся в центральной базе данных.

Исключения составляют ситуации, где биометрические данные подтверждаются как часть запутанного и необычного протокола. Когда Алиса подписывает контракт, например, у Боба еще нет копии ее подписи. Однако протокол работает — так как Боб знает, что он сможет проверить подпись впоследствии, если возникнет такая необходимость.

Существует много различных типов биометрических данных. Я уже упомянул почерк, звучание голоса, узнавание лица, отпечатки пальцев. К биометрикам также относятся линии на руке, сканирование сетчатки, сканирование радужной оболочки глаза, динамические характеристики подписи (не только то, как она выглядит, но и с каким нажимом, с какой скоростью она была начертана и т. д.) и другие. Одни технологии надежнее других — отпечатки пальцев намного надежней распознавания лица — но ситуация может измениться, поскольку технологии совершенствуются. Некоторые навязчивы — одна несостоявшаяся технология базировалась на образце отпечатка губ и требовала от пользователя поцеловать компьютер. В целом, биометрические данные будут считываться все лучше и лучше.

«Лучше и лучше» имеет два разных смысла. Во-первых, это значит, что самозванец не будет неправильно опознаваться в качестве Алисы. В целом, роль биометрических данных заключается в том, чтобы доказать, что Алиса-претендент и есть настоящая Алиса. Таким образом, если самозванец может успешно одурачить систему, то она работает не очень хорошо. Это называется *ложной уверенностью*. Во-вторых, это значит, что система не будет пытаться представить Алису как самозванца. Вернемся к началу: если роль биометрических данных — доказать, что Алиса — это Алиса, и если она не может убедить систему, что она — это она «не поддельная», тогда система также работает плохо. Это называется *ложным отрицанием*.

С течением времени биометрические опознавательные системы стали работать лучше в плане как ложной уверенности, так и ложного отрицания. Например, они устанавливают проверку отпечатков, так что ни пластиковый палец, ни чей-либо настоящий, но чужой палец не одурачат устройство, считывающее его отпечаток. Они лучше делают работу по отслеживанию ежедневных изменений в индивидуальных биометрических данных. Они более легки для использования.

Вообще, вы можете настроить биометрическую систему как в сторону допущения ошибки ложной уверенности, так и в сторону ложного отрицания. Здесь весьма нечеткие границы: если система получает отпечаток пальца, который, похоже, принадлежит Алисе, впустит ли она ее внутрь? Это зависит от того, склонна она в большей степени допустить ложную уверенность, или ложное отрицание. Если Алиса уполномочена взять карандаш со склада, то лучше допустить ошибку ложной уверенности; здесь больше неприятностей с отказом законному пользователю, чем если бы несколько карандашей просто потерялись. Если система защищает большие суммы денег, то ложное отрицание предпочтительней: оставить непра¬

вомочных пользователей снаружи более важно, чем иногда отказать в доступе законному пользователю. Если система приступает к выполнению ряда последовательных операций для запуска ядерных ракет, страшны оба варианта.

Биометрические данные значат очень много, так как на самом деле их сложно подделать: очень трудно нанести ложный отпечаток на свой палец или сделать сетчатку своего глаза похожей на чужую. Некоторые люди могут говорить голосами других (например, эстрадные имитаторы), а Голливуд способен сделать лица людей похожими на других. Но вообще подделать биометрические данные очень тяжело.

С другой стороны, биометрические данные даже слишком легки для подделки: не проблема украсть биометрики после того, как были сделаны начальные измерения. Во всех случаях, которые мы обсуждали, проверяющему необходимо было бы удостовериться не только в том, что биометрические данные верны, но и в том, что они были введены правильно. Вообразим удаленную систему, которая использует узнавание лица как биометрику. «Для получения разрешения возьмите свою фотографию, сделанную „Полароидом", и отправьте ее нам. Мы сравним картинку с той, что хранится у нас в файле». Как здесь осуществить нападение?

Легко. Чтобы выдать себя за Алису, возьмите ее фотографию, сделанную «Полароидом», так, чтобы владелица об этом не знала. Потом, несколькими днями позже, используйте ее, чтобы обмануть систему. Это нападение работает потому, что получить фотографию Алисы просто, это совсем не то, что сделать свое лицо таким, как у нее. И так как система не проверяет, что это изображение вашего лица, а только то, что оно соответствует Алисиному лицу в картотеке, мы в состоянии обмануть ее.

Подобным образом мы можем подделать биометрику подписи, используя фотокопировальную машину или факсимильный аппарат. Тяжело подделать президентскую подпись на официальном документе, дающем вам продвижение по службе, но легко вырезать его подпись с другого документа, поместить на письмо, дающее вам повышение, и отправить его по факсу в департамент трудовой занятости населения. Они не смогут установить, что подпись была вырезана с другого документа.

Мораль в том, что биометрические данные будут работать прекрасно только в случае, если проверяющий станет удостоверяться в двух вещах: во-первых, что они действительно поступили именно от лица, которое подлежит проверке, и во-вторых, что они соответствуют образцу в картотеке. Если система не в силах одновременно поддержать два эти условия, она ненадежна.

Еще один пример: отпечатки больших пальцев для получения разрешения на вход в систему с удаленным доступом. Алиса помещает отпечаток своего большого пальца в считывающее устройство, находящееся на клавиатуре. (Не смейтесь, большое количество компаний хотят, чтобы так и было, а технология уже существует[1].) Компьютер посылает цифровой отпечаток хосту. Хост проверяет его, и если он соответствует отпечатку, хранящемуся в файле, дает Алисе доступ. Это не будет работать потому, что легко украсть цифровой отпечаток Алисиного большого пальца, и когда он у вас будет, то обманывать хост снова будет легко.

Защищенные от несанкционированного вмешательства аппаратные средства помогают до тех пор (в пределах ограничений главы 14), пока они включают и устройство, считывающее биометрические данные, и механизм подтверждения.

[1] Выпускается фирмой Cheery, известной своими эргономичными изделиями. — *Примеч. ред.*

Это не сработает, если защищенное от несанкционированного доступа считывающее устройство посылает данные об отпечатке пальца через ненадежную сеть.

Мы подошли ко второй главной проблеме с биометрическими данными: эта система плохо справляется с отказами. Представим, что Алиса пользуется отпечатком своего большого пальца как биометрикой, и кому-нибудь вздумается украсть его. Что теперь? Он не является цифровым сертификатом (мы вернемся к этому в главе 15), который некая доверенная третья сторона может ей заменить. Это ее большой палец. У нее их всего два. Как только кто-нибудь украдет ваши биометрические данные, они останутся таковыми на всю жизнь; и их нельзя будет вернуть обратно.

Это та причина, по которой биометрические данные не могут выступать в роли шифровальных ключей (даже в том случае, если вам удастся разрешить противостояние между неясной логикой биометрических данных и безусловной математической логикой проблемы). Время от времени я вижу системы, которые используют шифровальные ключи, порожденные биометрическими данными. Это прекрасно работает до тех пор, пока данные не украдены. И я не думаю, что у кого-нибудь физически отрежут палец или нужный отпечаток пальца будет сымитирован на чьем-либо чужом пальце; я думаю, что кто-нибудь украдет цифровой отпечаток пальца. Однажды, когда это случится, система перестанет работать. (Ну, может быть, до тех пор, пока не будут украдены все 10 отпечатков пальцев...)

Биометрические данные могут быть хорошим механизмом, подтверждающим подлинность, но использовать их надо должным образом.

## Опознавательные знаки доступа

Третьим способом доказательства идентичности является использование чего-либо, что вы имеете: физического опознавательного знака любого рода[1]. Это старая форма контроля доступа: материальный ключ ограничивает доступ в сундук, комнату, здание. Обладание королевской печатью уполномочивает кого-либо на действия от имени короля. Более современные системы могут быть автоматизированными — электронные ключи в номере отеля — или ручными — распространенные предметы, предоставляющие доступ в здание. Основная идея та же самая; физический предмет служит подтверждением подлинности своего хозяина.

Для этого можно пойти по нескольким путям. Наиболее простой путь, когда хозяин может просто доказать, что данный знак принадлежит ему. Есть компьютеры, включаемые физическим ключом; так работают компьютеры, которым требуется смарт-карта. Основная идея любого опознавательного знака в том, что вы помещаете знак в некоторое отверстие в каком-то месте, и после этого компьютер подтверждает, что вы действительно это сделали. Если это так, вы попадаете в систему.

Наиболее серьезная проблема с такой системой в том, что знаки могут быть украдены. Например, если кто-нибудь украдет ключи от вашего дома, то он сумеет открыть его. Таким образом, система в действительности не может подтвердить

[1] Token — знак, символ, маркер, электронный ключ, аппаратный контроллер, устройство идентификации. Однозначного перевода нет, используется в зависимости от контекста. — *Примеч. ред.*

подлинность лица; она подтверждает подлинность знака. Большинство компьютерных систем для преодоления этой уязвимости соединяют в себе знак доступа с паролем, который иногда называют личным идентификационным кодом (PIN). Примером могут быть банковские карты. Банкоматы подтверждают подлинность карты и спрашивают идентификационный номер для подтверждения подлинности пользователя. Идентификационный номер бесполезен без знака доступа. Некоторые сотовые телефонные системы работают точно таким же образом: вам нужен физический телефон и код доступа, чтобы сделать звонок, оплачиваемый с частного телефонного счета.

Кроме того, что знак могут украсть, кто-нибудь может скопировать его. Некоторые знаки скопировать легко, например физические ключи. Таким образом, знаки могут быть украдены, скопированы и перемещены без ведома своего владельца.

Другая проблема в том, что должен быть некий путь, подтверждающий, что опознавательный знак в действительности там, где он должен находиться. Подумайте о знаке как о перемещаемой, изменяемой биометрике — и вы получите все проблемы проверки безопасности из предыдущего раздела. Однако здесь при необходимости знак может быть изменен.

Проиллюстрирую эту проблему на примере использования кредитных карт. Сложно подделать физическую кредитную карту потому, что фальшивку опасно подсунуть при покупке вещей в магазине. Нельзя полагаться, что служащий магазина не заметит, что карта не настоящая. Легче использовать поддельную кредитную карту по телефону. В магазине служащий проверит подлинность как номера счета на кредитной карте, так и ее саму — как знак. По телефону оператор не сумеет определить подлинность физического знака, только номер счета.

В этом — другая, относительно менее значимая проблема, которую можно наблюдать на примере некоторых знаков. Если пользователи могут оставить знак в отверстии, куда поместили его для операции, они часто это и делают. Если пользователи должны вставить смарт-карту в прорезь перед тем, как она загрузится, они, вероятно, оставят ее там на весь день и всю ночь, даже если их самих там не будет. На слишком долгое для идентификации время.

Все эти обсуждения предполагают, что какой-нибудь вид считывающего устройства общается со знаком, и пользователь поместил его в считывающее устройство. Но часто такой возможности не бывает: у большинства компьютеров нет требуемого считывающего устройства, или система работает с мобильным пользователем, который сидит где-то в другом месте, а не за своим привычным компьютером. С этой ситуацией связаны две различные технологии.

Первая — это «вызов/ответ». Знак — устройство идентификации — карманный калькулятор с цифровой клавиатурой и маленьким экраном. Когда пользователь хочет подключиться, он вызывает удаленный хост. Он отправляет этот вызов со своего знака. Знак подготавливает соответствующий запрос, который передает в компьютер, а тот переправляет его хосту. Хост производит аналогичные вычисления и, если результат соответствует ожидаемому, подтверждает подлинность.

Вторая технология основана на временной синхронизации. Знаком является аналогичный карманный калькулятор с одним экраном. На экране регулярно сменяются номера, обычно раз в минуту. Удаленный компьютер просит пользователя напечатать то, что показано на экране. Если ответ пользователя соответствует тому,

что ожидает удаленный компьютер, он производит подтверждение подлинности. Таким образом работает адаптер SecurID[1].

Конечно, полная система может также включать пароль, знак вызова/ответа, для начала работы может даже потребовать дополнительно ввода пароля; и другие вспомогательные меры безопасности. Основная идея все-таки в том, что некое секретное вычисление происходит внутри электронного ключа, который подменить нельзя. Нападающий не станет притворяться, будто у него есть знак, потому что не знает, как рассчитывать ответы, основанные на вызовах, или не знает, как рассчитывать величины, основанные на временной синхронизации. Сделать это можно только одним путем — имея настоящий знак.

Это работает в большей или меньшей степени. Шифровальные техники, кодирование или хэширование обеспечивают безопасность. Удаленный компьютер знает, как провести расчеты, так что система безопасна в такой же степени, что и ключевой код главного хоста. Любой, кто перепроектирует знак, сможет выяснить, как произвести расчеты; таким образом, система безопасна ровно настолько, что и знаки (см. главу 14). Но это достаточно хорошо и, конечно, намного лучше, чем «голые» пароли. Проблемы безопасности возникают в сети и при подтверждении подлинности компьютера.

Напоследок обсудим еще один знак: записанный пароль. В сообществе, занимающемся проблемами безопасности, существует реакция коленного рефлекса на запись паролей, но если это сделано должным образом, то может значительно улучшить защиту. Кто-нибудь, кто записывает свой пароль, превращает то, что он знает (свой пароль), в то, что он имеет (клочок бумаги). Эта уловка позволяет ему использовать более длинные пароли, которые являются более надежными. Здесь есть все проблемы простого знака: он может быть скопирован или украден. Защита не будет работать, если Алиса написала свой пароль на желтом липком листочке, наклеенном на монитор ее компьютера. Для нее будет лучше положить свой пароль в бумажник — это надежнее. Возможно, лучшим решением будет иметь две части пароля: одну будет помнить Алиса, а другая будет записана на листочке, лежащем в ее бумажнике.

Есть системы с одноразовыми паролями. У пользователя находится список паролей, записанных и используемых однократно. Конечно, это хорошая система подтверждения подлинности — список паролей является знаком — до тех пор, пока список находится в безопасном месте.

# Протоколы аутентификации

Протоколы аутентификации — это криптографические способы подтверждения подлинности личности Алисы через сеть. Основной протокол аутентификации достаточно прост.

[1] SecurID использует двухфакторную аутентификацию. Для входа в систему пользователь должен в ответ на приглашение, сгенерированное клиентской (UNIX, Windows NT, NetWare и другие) или серверной частью (Windows NT, UNIX), ввести с клавиатуры свой персональный идентификационный номер PIN-код («что я знаю») и цифровой код, отображаемый на жидкокристаллическом дисплее электронной карточки (в виде брелока) через каждые 30 или 60 секунд («что мне позволено») — то есть опознавательный знак и пароль. — *Примеч. ред.*

1. Алиса набирает свое имя пользователя и пароль на компьютере-клиенте. Клиент отправляет эту информацию серверу.
2. Сервер ищет указанное имя пользователя в базе данных и отыскивает соответствующий пароль. Если он соответствует паролю, набранному Алисой, ей предоставляется доступ.

Проблема в том, что база данных паролей должна быть защищена. Решение в том, чтобы хранить не пароли, а хэш-функции паролей.

1. Алиса набирает свое имя пользователя и пароль на клиенте. Клиент отправляет эту информацию серверу.
2. Сервер хэширует набранный Алисой пароль.
3. Сервер ищет имя пользователя с именем Алиса в базе данных и отыскивает соответствующее хэш-значение. Если это хэш-значение соответствует хэш-значению пароля Алисы, ей предоставляется доступ.

Уже лучше. Главная проблема со вторым протоколом в том, что пароли открыто посланы по сети. Кто-нибудь, рыскающий по сети, может собирать имена пользователей и пароли. Решение включает в себя хэширование пароля перед тем, как отослать его (более старые версии Windows NT делают это), но словарные нападения в состоянии справиться и с этим.

Так как словарные нападения стали более мощными, системы начали использовать прием, известный как «соление» (на самом деле они делали это и ранее, хороший пример предусмотрительности проектировщика). «Соль» — это известная случайная константа, хэшируемая вместе с паролем. Вследствие чего сделать словарные нападения сложней; вместо единственного хэш-значения для пароля «кот» могут быть 4096 различных вариантов для «кот» плюс 12 бит случайной «соли». Словари хэшированных паролей должны были бы быть в четыре раза «толще». Но способность произвести быстрые словарные нападения в реальном времени делает эту контрмеру устарелой; словари просто включают все возможные значения «соли».

Kerberos («Цербер») является более хитрым протоколом аутентификации. Здесь Алиса должна иметь долгосрочный ключ, используемый совместно с надежным сервером в сети, называемым Kerberos-сервером. Чтобы войти во взятый наугад сервер в сети — назовем его сервером Боба, — выполняется следующая процедура:

1. Алиса запрашивает разрешение у сервера Kerberos для входа на сервер Боба.
2. Сервер Kerberos проверяет, допускается ли Алиса на сервер Боба. (Примечание: серверу Kerberos не нужно знать, что Алиса — та, кем она себя назвала. Если это не она, протокол прервется на шаге 6.)
3. Сервер Kerberos высылает Алисе «билет», который она обязана отдать серверу Боба, и ключ к сеансу, который она может использовать, чтобы доказать Бобу, что она Алиса.
4. Алиса использует ключ к сеансу с сервера Kerberos для создания «удостоверения», которое она будет использовать, чтобы убедить Боба, что она Алиса.
5. Алиса посылает Бобу и билет, и удостоверение.
6. Боб проверяет. Если все подтверждается, он дает Алисе доступ. (Боб также имеет используемый совместно с сервером Kerberos долгосрочный ключ. Билет — это сообщение с сервера, зашифрованное в долгосрочном ключе Боба.)

Этот протокол защищен тем же способом, что и протоколы физических билетов. Сервер Kerberos печатает билеты. Он дает билеты Алисе, а она в свою очередь может предоставить их Бобу. Боб может утвердить билет, так как он знает, что Алиса получила его с сервера Kerberos.

И у этого протокола есть несколько приятных свойств. Долгосрочные ключи Алисы и Боба, которые похожи на пароли, никогда не посылались по сети. Отрицательная сторона в том, что системе для работы нужен сервер Kerberos. Сервер Kerberos является доверенной третьей стороной. Это может стать «узким местом» в системе в 9:00 утра, когда каждый пытается войти в свой компьютер.

Kerberos был изобретен Массачуссетским технологическим институтом в 1988 году и с того времени используется в мире UNIX. Kerberos является частью Windows 2000, но исполнение Microsoft отличается от стандартного и несовместимо с остальным миром Kerberos. Я могу только предполагать, что это было сделано намеренно по соображениям, связанным с рынком, но сделано таким образом, что ослабило защиту. Нельзя всего лишь изменить протокол безопасности и предполагать, что измененный протокол так же надежен.

Другие подтверждающие подлинность протоколы входа в систему используют открытые шифровальные ключи. IPsec и SSL, например, пользуются протоколами аутентификации с открытыми ключами. Некоторые системы прибегают к более простым, но тайным протоколам. Протокол, в котором звонящий с сотового телефона доказывает, что он может сделать телефонный звонок в этой частной сети, является одним из них.

## Однократная регистрация

Вещь, которая изрядно раздражает пользователей в системе со строгими требованиями к безопасности, — это большое число паролей. Пользователь должен набирать один пароль для входа в свой компьютер, другой для входа в сеть, третий для входа на отдельный сервер в сети и т. д. Люди задают вопрос: не было бы лучше, если бы пользователь зарегистрировался один раз, с одним паролем, и затем мог бы управлять компьютерами как ему угодно, без использования других паролей?

Однократная регистрация — это решение пользовательской проблемы. К несчастью, она не очень хорошо работает. Во-первых, нетрудно увязнуть в болоте различных приложений и мер безопасности, которые плохо согласуются между собой. Это значит не просто ограничиться выбором определенного пароля для каждого — что плохая идея, — но учесть все проблемы множества взаимодействующего программного обеспечения. Во-вторых, дополнительный риск связан с тем, что возможная уязвимая точка является единственной. Есть разница между потерей одной кредитной карточки и целого бумажника.

Продукты с одноразовой регистрацией существуют, и в некоторых ситуациях они работают. Но они ни в коем случае не являются панацеей, как о том любят объявлять их продавцы.

# Глава 10. Безопасность компьютеров в сети

В этой главе я хочу поговорить об атаках, совершаемых через Интернет. Такие атаки можно считать компьютерными, тогда их следовало бы рассматривать в главе 8. Можно отнести их и к сетевым нападениям, о которых речь пойдет в главе 11. Но я выделяю эти атаки в самостоятельный класс и посвящаю им отдельную главу.

## Разрушительные программы

Самое первое, с чем большинство из нас сталкивается при первом знакомстве с проблемами компьютерной безопасности — это именно разрушительные программы, то есть программы, умышленно причиняющие неприятности. Даже в том случае, если компьютер не подключен к сети и доступ к нему имеется только у вас, вам не следует забывать о вирусах. Ведь вы не знаете точно, какие программы в данный конкретный момент выполняются вашим компьютером, и только можете надеяться, что выполняемые программы работают как надо. Запуская программы, в надежности которых вы не уверены, вы рискуете.

К разрушительным программам кроме вирусов относятся так называемые «троянские кони» и «черви». Они обычно состоят из двух частей: «полезной нагрузки» и механизма распространения. «Полезная нагрузка» — это та составляющая, которая, собственно, и вызывает сбои. Традиционно нагрузка была не очень разнообразна — прототипы вирусов выводили на экран какое-либо надоедливое сообщение, переформатировали жесткий диск компьютера жертвы либо не делали вообще ничего. Но в некоторых случаях нагрузка способна причинить и большие неприятности: изменить установки контроля доступа компьютера, украсть секретный ключ и отправить его по электронной почте и т. п. Результат таких действий может оказаться опасным, и я считаю, что следует ожидать появления более коварных нагрузок в ближайшие годы[1]. В этой книге для нас наибольший интерес будут представлять механизмы распространения, по которым мы и классифицируем разрушительные программы.

[1] Вирусные атаки в 2001 году нанесли ущерб более чем на 13 миллиардов долларов (Computer Economics). Весь год в Сети свирепствовали «Code Red», «Goner» и «Qaz» — вирусы новой волны, так называемая «гибридная угроза». Это сочетание хакерских приемов и вирусной техники — вирусы оставляли люки для дальнейшего использования зараженных машин в целях проведения атак типа «отказ в обслуживании». В 2002 году специалисты компьютерной безопасности признали интернет-червя Klez вирусом года. По оценке компании Sophos (антивирусные программы), одно электронное письмо из каждых 169 не обходилось без Klez в период его победоносного шествия. — *Примеч. ред.*

## Компьютерные вирусы

Биологический вирус представляет собой невидимый в оптический микроскоп инфекционный агент, вызывающий болезни растений, животных и бактерий. По существу, он состоит из белковой оболочки, содержащей РНК или ДНК. Вирусы не способны воспроизводиться вне клетки-хозяина, поэтому их, как правило, не относят к живым организмам. Таким образом, налицо прямая аналогия с вирусами компьютерными. Компьютерный вирус представляет собой фрагмент компьютерного кода, который может прикрепляться к другой компьютерной программе (сам по себе этот фрагмент существовать не может). Прикрепившись, он воспроизводится, делает новые собственные копии, которые внедряются в другие программы. И так далее.

В 1983 году студент Фред Коэн (Fred Cohen) написал первый компьютерный вирус. Он сделал это, только чтобы создать прецедент (удивительно, но большинство людей не верили, что это возможно). Многие скопировали этот вирус, большинство этих людей хотели просто досадить окружающим. В настоящее время насчитывают от 10 000 до 60 000 вирусов (в зависимости от критериев подсчета), большинство которых написаны для IBM-совместимых персональных компьютеров. По некоторым оценкам, которые мне встречались, ежедневно создаются шесть новых вирусов, но я считаю, что это — ложь и паникерство. Всего несколько сотен вирусов встречаются «в диком виде» (имеется в виду «на жестком диске у людей, не принимающих непосредственного участия в исследованиях компьютерных вирусов»), но те, которые встречаются, могут быть разрушительными.

Вирусы можно подразделить на три основных класса: файловые вирусы, загрузочные вирусы (вирусы, поражающие загрузочный сектор) и макровирусы.

Долгое время наиболее распространенными были файловые вирусы. Они работали, присоединяясь к программным файлам, например к текстовому редактору или компьютерным играм. При запуске инфицированной программы этот вирус размещается в памяти так, чтобы заразить другие приложения, запускаемые пользователем. Таким способом вирус распространяется по компьютеру пользователя, а если пользователь даст кому-то дискету с инфицированным приложением (или пошлет это приложение по сети), то зараженным окажется и другой компьютер.

Большинство файловых вирусов уже вышли из употребления. Изменения в устройстве компьютеров привели к тому, что вирусы потеряли способность запускаться; программное обеспечение часто требует обновления при установке новой операционной системы или нового процессора. Многие файловые вирусы вымерли после того, как в 1992 году была выпущена Windows 3.1; вирусы просто рушили эту операционную систему и в результате не могли распространяться.

Загрузочные вирусы менее распространены. Эти вирусы размещаются на специальном участке диска (дискеты или жесткого диска), данные с которого загружаются в память при загрузке компьютера. После того как этот вирус внедряется в память, он может заразить соответствующие секторы всех имеющихся жестких Дисков и гибких дисков, вставленных в дисковод, и таким образом распространиться на другие системы. Загрузочные вирусы чрезвычайно эффективны, и, даже несмотря на гораздо меньшее количество штаммов (различных разновидностей), они какое-то время преобладали над файловыми вирусами.

Загрузочные вирусы могут мирно сосуществовать с Windows 3.1, но большинство из них не пережили появления Windows 95. Несовместимость при загрузке и появляющиеся на экране предупреждения сильно затруднили распространение вирусов. Были вирусы, созданные специально для Windows 95, но ни один из них не получил широкого распространения, поскольку никто уже не загружается с гибкого диска.

Последний класс вирусов — это макровирусы. Они написаны на языке сценариев и заражают не программы, а файлы данных. Во многих текстовых процессорах, электронных таблицах и программах, работающих с базами данных, используются специальные языки разработки сценариев. Такие сценарии (программы на макроязыке, или просто макросы) используются для автоматизации задач и хранятся вместе с данными. Первый макровирус для Microsoft Word — Concept — впервые обнаружили в «диком виде» в 1995 году; в текстовом редакторе Emacs такие вирусы существовали уже в 1992 году.

Макровирусы могут распространяться существенно быстрее других, поскольку люди гораздо чаще обмениваются данными, чем программами. А поскольку программное обеспечение электронной почты и передачи файлов делается все проще в обращении, эти вирусы станут распространяться еще быстрее. Бывают макровирусы, способные существовать в различных операционных средах: некоторые макровирусы для Microsoft Office могут заражать как компьютеры Windows, так и Macintosh.

Макровирусы — это будущее компьютерных вирусов. Все вирусы, которые быстро распространяются по Интернету, — это макровирусы. Лучшие из вирусов используют психологические приемы, побуждающие пользователя установить, запустить или размножить их.

Антивирусное программное обеспечение — это более выгодный бизнес, чем написание вирусов. (Я полагаю, это очевидно: за вирусы никто денег не платит.) Большинство антивирусных программ сканируют файлы, выискивая вирусы. В программах есть база данных, содержащая «отпечатки пальцев» вирусов — фрагменты кода, про которые известно, что они являются частью вирусов. Когда программа находит такой же отпечаток, она получает информацию, что файл заражен, и, чтобы «дезинфицировать» его, удаляет вирусный код. Метод сканирования «отпечатков» работает только после того, как компания, создавшая антивирусную программу, выделила вирус в лаборатории и включила в список новый отпечаток. Поэтому усовершенствование антивирусного программного обеспечения — бойкий бизнес.

Все вирусы, которые встречались до сих пор, были направлены против крупных вычислительных машин, а не против периферийных или встроенных систем. Однако можно написать вирус на языке PostScript. Он мог бы распространяться от документа к документу. Он мог бы воздействовать на принтеры. Можно создать вирус, заражающий сотовые телефоны и распространяющийся по сотовой сети. Нам уже встречался вирус, специализирующийся на устройствах WebTV. Можно создать вирус, заражающий практически любую компьютерную систему. Если его еще нет, то только потому, что люди, которые обладают необходимыми знаниями и не отягощены строгой моралью, до сих пор не удосужились его создать.

Чтобы обнаружить ранее не известные вирусы, полиморфные вирусы (видоизменяющиеся при каждом инфицировании) и зашифрованные вирусы (скрываю¬

щие свои «отпечатки» при помощи криптографии), некоторые антивирусные программы тестируют компьютерную систему, выискивая подозрительное поведение. (Обычные программы поиска вирусов довольно глупы — чтобы их обмануть, иногда достаточно просто изменить какую-нибудь мелочь.) Такие программы работают довольно хорошо, однако они перекладывают на пользователя бремя принятия решения: вирус это или ложная тревога?

«Лекарства» от вирусов не существует. Математически доказано, что всегда можно написать вирус, который не сможет нейтрализовать ни одна из существующих антивирусных программ. (Даже модель Белла-Лападулы не предохраняет от вирусных атак.) Я не буду вдаваться в подробности, но основная идея в том, что если создатель вируса знает, что именно ищет антивирусная программа, он всегда имеет шанс разработать свой вирус, незаметный для нее. Конечно, после этого программисты, борющиеся с вирусами, могут усовершенствовать свою программу, которая будет определять уже и новый вирус.

## Черви

Червями называют те разрушительные программы, которые специализируются на компьютерах, подключенных к сети. Это самовоспроизводящиеся программы, которые, в отличие от вирусов, не прячутся в других программах. Они существуют самостоятельно, блуждают по компьютерным сетям, причиняя повреждения.

Роберт Т. Моррис (Robert T. Morris) «выпустил» самого известного червя в 1988 году. Это был интернет-червь, который вывел из строя более 6000 компьютеров: 10% всех серверов Интернета. Червь появлялся на одной машине. Затем он предпринимал попытку проникнуть по сети в другие машины, используя несколько основных приемов. Когда это удавалось, червь засылал на новый компьютер копию своего кода. А затем эта копия повторяла весь процесс, пытаясь проникнуть в очередную машину. Обычно черви работают именно так. Тот червь мог бы причинить более крупные неприятности, если бы не счастливая ошибка. Изначально не планировалось, что 6000 зараженных компьютеров будут выведены из строя; червь должен был заразить их тайно, не привлекая всеобщего внимания. Ошибка в программе червя вызвала повреждение зараженных компьютеров. В главе 13 я более подробно расскажу о том, как работал тот червь и в чем конкретно заключалась ошибка.

Еще один червь известен под названием Pretty Park. Эта программа, функционирующая в среде Windows, приходит по электронной почте как вложение в сообщение. Если вы запускаете эту программу, она рассылает свои копии всем адресатам вашей записной книги в Outlook Express. Кроме того, она пытается подсоединиться к серверу IRC (Internet relay chat)[1] и отправить сообщения участникам чата. Автор червя может к тому же использовать это соединение, чтобы получать информацию из вашего компьютера. ILOVEYOU и все его варианты по сути своей — тоже черви.

[1] Глобальная система, посредством которой пользователи могут общаться друг с другом в реальном времени. — *Примеч. перев.*

## Троянские кони

Троянские кони — это разрушительные фрагменты, которые встроены в какие-то «нормальные» программы, чтобы одурачить пользователя, который будет думать, что это нечто полезное. Помните, откуда появилось это название? Греки десять лет осаждали Трою, но она не сдавалась. То ли от отчаяния, то ли от скуки Одиссей приказал греческим воинам построить большого деревянного коня, внутри которого могли бы спрятаться несколько человек. Этот сюрприз греки оставили троянцам как признание поражения, а затем сделали вид, что уплывают. Троянцы привезли деревянного коня в город — все художники изображают коня стоящим на платформе с колесами, — несмотря на предсказание Кассандры о том, что это приведет к гибели Трои. Ночью греки вылезли из коня, открыли ворота и впустили внутрь остальную греческую армию. После этого греки истребили троянцев, варварски разграбили и сожгли город. (По крайней мере, Гомер все описал именно так. Никто не знает, правда это или нет. Даже сам город считали мифом, пока Генрих Шлиман не обнаружил Трою в конце XIX века.)

Аналогично, цифровой троянский конь — это код, преднамеренно помещенный в вашу систему, который маскируется под безвредную (или полезную) программу, но делает что-то неожиданное или нежелательное. (С формальной точки зрения код, который вы сознательно размещаете в вашей системе, — это троянский конь, а код, который вводит в вашу систему кто-то другой, называют *логической бомбой.)* Программист вписывает такой код в крупное программное приложение, которое в результате может начать работать неправильно, если, например, программист будет исключен из платежной ведомости. Тимоти Ллойд (Timothy Lloyd), диспетчер сети в Omega Engineering, в 1996 году установил логическую бомбу, которая подорвала производственные мощности его бывших работодателей и обошлась им более чем в 12 миллионов долларов.

Троянским конем может быть программа, которая тайно устанавливается в вашем компьютере, следит за буфером клавиатуры до тех пор, пока не обнаружит нечто, напоминающее номер кредитной карты, — правильное количество цифр, совпадение контрольной суммы, — и посылает этот номер кому-нибудь при помощи TCP/IP. Это также и приложение Java, которое прерывает соединение вашего модема и соединяет вас с 900 номерами в Молдавии (такой троянский конь на самом деле существовал).

Атака троянского коня коварна и опасна, поскольку вы можете и не догадываться о его работе. Один из популярных троянских коней для Microsoft Windows называется Back Orifice. Если он имеется на вашем диске, удаленный пользователь может эффективно подсоединиться к вам через Интернет и хозяйничать в вашем компьютере. Он может загружать себе ваши файлы, а вам — свои, удалять файлы, запускать программы, изменять конфигурации, захватывать управление клавиатурой и мышью, видеть все, что отображается на экране сервера. Тот же удаленный пользователь может вести более разрушительную деятельность: перезагружать компьютер, показывать произвольные диалоговые окна, включать и отключать микрофон или камеру, перехватывать нажатия клавиш (и пароли). А кроме того, существует расширяемый язык, позволяющий писать модули. (Я все жду, что кто-нибудь распространит модуль, который будет автоматически выискивать

и записывать закрытые ключи PGP или последовательности регистрации для Всемирной паутины.)

Кроме Back Orifice и других хакерских инструментов по принципу троянских коней могут работать многие программы удаленного администрирования. DIRT (Data Interception by Remote Transmission, перехват данных с помощью удаленной передачи) — это троянский конь, разработанный правительством США и находящийся в распоряжении полиции.

Это все грубо действующие троянские кони, но бывают и более коварные. Некоторые из них собирают и отсылают автору имена и пароли пользователей. Другие незаметно изменяют шифровальную программу так, что выбор ключей становится ограничен. (Мне встречались измененные таким образом версии PGP.) Вас могут заставить поверить во что угодно, подсунув на ваш компьютер фальшивый сертификат. (Эта идея использовалась для лабораторных демонстраций нападений на систему кодовых обозначений Microsoft.) Эти троянские кони не делают ничего такого, о чем вы могли бы легко догадаться, но они способны на многое, что никогда не придет вам в голову. Распределенные атаки, приводящие к отказу в обслуживании, осуществлявшиеся с помощью Интернета, используют троянских коней для заражения промежуточных компьютеров.

Наиболее сложный этап при подобных атаках — это помещение троянского коня на компьютер ничего не подозревающей жертвы. Один из вариантов — проникнуть в офис жертвы и установить троянца на нужный компьютер; в следующей главе мы обсудим некоторые способы защиты от атак такого рода. Можно убедить жертву собственными руками установить троянского коня; о манипулировании людьми мы поговорим в главе 17. Можно атаковать компьютер через Интернет — мы вернемся к этому в главе И. И наконец, можно проникнуть в компьютер при помощи самих разрушительных программ, создавая вирусы.

## Современные разрушительные программы

Для программного обеспечения, призванного причинять неприятности, 1999 год стал переломным. Различные типы — вирусы, черви и троянские кони — слились и перемешались. И стали в результате еще более опасными. Новизна состояла не в автоматической пересылке разрушительных программ по электронной почте — до этого были Christma.exec в 1987 году (через систему электронной почты PROFS (Professional Office System, профессиональная офисная система)) и ShareFun в 1997 году, — но 1999 год стал первым годом, когда распространяющиеся по электронной почте «вредные» программы смогли заразить крупные зоны Интернета. Новые типы этих программ игнорировали средства корпоративной защиты и тонел¬лировали сквозь брандмауэры. Это действительно значительный шаг.

Вирус продолжает существовать, если он воспроизводится на новых компьютерах. До Интернета компьютеры сообщались в основном при помощи гибких дисков. Следовательно, большинство вирусов передавались при помощи дискет, и лишь изредка через электронные доски объявлений (bulletin board system, BBS).

Дискеты как переносчики инфекции имеют свои особенности. Во-первых, заражающие программы распространяются довольно медленно. Компьютер инфицирует через дискету другой компьютер, с него переходит на пять других, и по ис¬

течении недель или месяцев этот процесс приведет к эпидемии. Хотя возможно, что кто-то поместит инфицированную вирусом программу на электронную доску объявлений, тогда тысячи компьютеров заразятся за одну-две недели.

Во-вторых, перенос программ при помощи дискет легко блокировать. Большинство антивирусных программ могут автоматически сканировать все гибкие диски. Опасные программы на входе блокируются. Электронная доска объявлений способна все же доставить некоторые проблемы, но большинство пользователей приучили себя к тому, что не следует загружать программы с ненадежных BBS. А кроме того, антивирусные программы могут автоматически сканировать новые файлы в поисках опасного кода.

И в-третьих, при таком способе распространения вирусов антивирусные программы справляются со своей задачей. Написать программу, блокирующую известные опасные программы, в общем, не сложно. Она будет работать по принципу сканирующего поиска кода, характерного для вируса (так называемых сигнатур). Найдя зараженный участок, программа автоматически удалит вирус и вернет все в нормальное состояние. Процедура удаления уникальна для каждого вируса, но разработать ее сравнительно нетрудно. Антивирусные программы включают десятки тысяч сигнатур, каждая из которых характерна для определенного вируса. Компании выпускают новые программы в тот же день, как узнают о новом вирусе. И при условии, что вирус распространяется медленно, такая защита работает. Ежемесячно происходит автоматическое обновление большинства антивирусных программ. До 1999 года такая защита была вполне удовлетворительна.

Все изменилось с распространением электронной почты. 1999 год принес нам макровирус Melissa для Microsoft Word и червя Worm.ExploreZip, а 2000 год — червя ILOVEYOU и массу его разновидностей, не говоря обо всем остальном. Разрушительные программы такого типа размножаются по электронной почте и используют ее особенности. Эти программы отправляют себя по почте людям, с которыми переписывается хозяин инфицированного компьютера, обманом заставляя получателей открыть или запустить их. В таком случае процесс распространения вирусов занимает не недели и не месяцы, а считанные секунды.

Антивирусные компании по мере возможности выпускают усовершенствованные версии программ, которые умеют ловить определенные вирусы, но если вирус может заразить 10 миллионов компьютеров в течение нескольких часов (так оценивают скорость распространения ILOVEYOU), то прежде, чем его удастся зафиксировать, он вызывает множество повреждений. А что если вирус постарается «спрятаться», чтобы его код не обнаружили в течение нескольких дней? Что, если червь нацелен на конкретную машину и удаляет свой код из всех компьютеров, у которых идентификатор пользователя не соответствует заданному? Сколько времени потребовалось бы на то, чтобы его обнаружить? Что, если этот червь отправляет по электронной почте в анонимный почтовый ящик копию регистрационного имени пользователя (в большинстве случаев содержащего пароль) и лишь затем самоуничтожается? Что, если такой код способен автоматически обновляться? Что, если он автоматически зашифровывает свои распространяющиеся копии при помощи системы PGP? А если он видоизменяется и ускользает от антивирусных программ? Или неделями прячется в системе? Даже минутные размышления на эту тему заставляют нас нарисовать себе довольно жуткую картину.

Поскольку электронная почта встречается повсеместно, путешествующие с ее помощью опасные программы в силах проникнуть куда угодно. Они могут просочиться через такое интернет-соединение, через которое больше ничего не проходит. Их нельзя остановить брандмауэром; они проникают сквозь него, неожиданно оказываются внутри системы и производят повреждения. Эффективность брандмауэров будет снижаться по мере того, как мы будем расширять перечень используемых служб (e-mail, Веб и т. п.) и добавлять все более сложные приложения во внутреннюю сеть, а авторы разрушительных программ будут принимать все это во внимание. Такой метод «проникновения внутрь с последующей работой» еще более ухудшит ситуацию.

В настоящее время, разрабатывая средства для защиты от вирусов, червей и троянов, стараются подражать биологическим принципам борьбы с вирусами. Тем не менее я настроен скептически по двум причинам. Прежде всего биологические вирусы эволюционируют медленно: удачные новые мутации со временем могут закрепиться, а затем медленно распространиться по виду. Биологические иммунные системы приспособлены именно к атакам такого рода. А компьютерные вирусы, в отличие от биологических, специально создают «смертельными».

Вторая причина моего скепсиса в том, что биологические иммунные системы ориентированы на то, чтобы защищать вид за счет отдельной особи. Относительно генофонда — это правильная стратегия, но для защиты конкретного компьютера от разрушительных программ она не годится.

Мне кажется более интересным такое решение: связать компьютеры с центром автоматического выявления вирусов. Если какой-нибудь компьютер замечает подозрительный код, он отсылает его для последующего анализа. Такой подход имеет помимо перспективы и ряд новых опасностей. Кроме того, он все-таки не сможет обеспечить необходимую скорость обнаружения вирусов. Любая крупная рассредоточенная система в любом случае допускает возможность заражения вирусами. Если система безопасности не продумана снизу доверху, мы должны быть постоянно готовы к борьбе с попытками установить контроль над системой.

Легко критиковать Microsoft за усугубление проблемы. Языки сценариев Microsoft являются довольно мощным средством, однако они созданы в предположении, что все, с чем вы работаете, заслуживает доверия. Эти языки позволяют иметь доступ ко всем ресурсам операционной системы (сравните с моделью безопасности Java). Они позволяют разрушительным программам использовать свойства Microsoft Outlook для автоматической рассылки своих копий корреспондентам пользователя. Microsoft, безусловно, заслуживает порицания за то, что созданные ею мощные средства — Word и Excel — размывают границы между исполняемыми файлами, которые потенциально таят опасность, и файлами данных, бывшими до сих пор безопасными. Он заслуживает осуждения и за то, что интегрированная в Outlook 2000 поддержка языка HTML допускает возможность внедрения основанной на HTML разрушительной программы при простой загрузке электронного сообщения (оно автоматически открывается в режиме предварительного просмотра). А также за то, что разрушительная программа может использовать интеграцию ActiveX в Internet Explorer 5.0, чтобы распространяться без участия пользователя. Microsoft создала операционную среду, в которой разрушительные программы легко создаются, легко распространяются и могут причинять множество неприятно¬

стей. Но основная проблема, состоящая в том, что мобильный код не заслуживает доверия, намного более остра.

# Модульная программа

В прежние времена (в 1970-х) компьютерные программы были крупными и громоздкими, их было трудно писать, и еще труднее использовать. Затем кому-то пришла идея поделить большие программы на мелкие, более простые для понимания компоненты. Объектно-ориентированное программирование, C++, подключаемые модули — все это различные реализации этой идеи. Проблема в том, что современное программное обеспечение, в основе которого лежат небольшие компоненты, намного сложнее защитить.

Рисунок 10.1 иллюстрирует принцип, по которому построены старые программы: большие приложения опираются на небольшую операционную систему. Большинство современных программ похожи на Приложение 1 — приложения с компонентами — или на Приложение 2 — приложения с компонентами, состоящими из компонентов (рис. 10.2). Представьте себе, как устроен браузер. Одним из компонентов является виртуальная машина Java (Java Virtual Machine). Апплеты Java запускаются на самом верху этой конструкции. Некоторые апплеты Java могут заменяться. Имеются все виды макросов для вашего текстового редактора и электронных таблиц. Вы можете загружать сменные PGP для Eudora. Не исключено, что каждую неделю вы загружаете те или иные сменные модули для своего браузера.


**Рис. 10.1.** Устройство старого программного обеспечения


**Рис. 10.2.** Современная структура компонентно-ориентированного программного обеспечения

В действительности, хотя браузер и продается как единая программа, он состоит из множества работающих вместе разнообразных компонентов. Так же устроены текстовый редактор и электронные таблицы; в Microsoft Word свыше тысячи компонентов. Таким образом, вы имеете дело со схемой Приложение 3: небольшое приложение-основа, к которой крепится множество компонентов, состоящих, в свою очередь, из компонентов. Даже операционная система построена по тому

же принципу; на рис. 10.3 представлена модель Windows NT: компоненты состоят из компонентов.

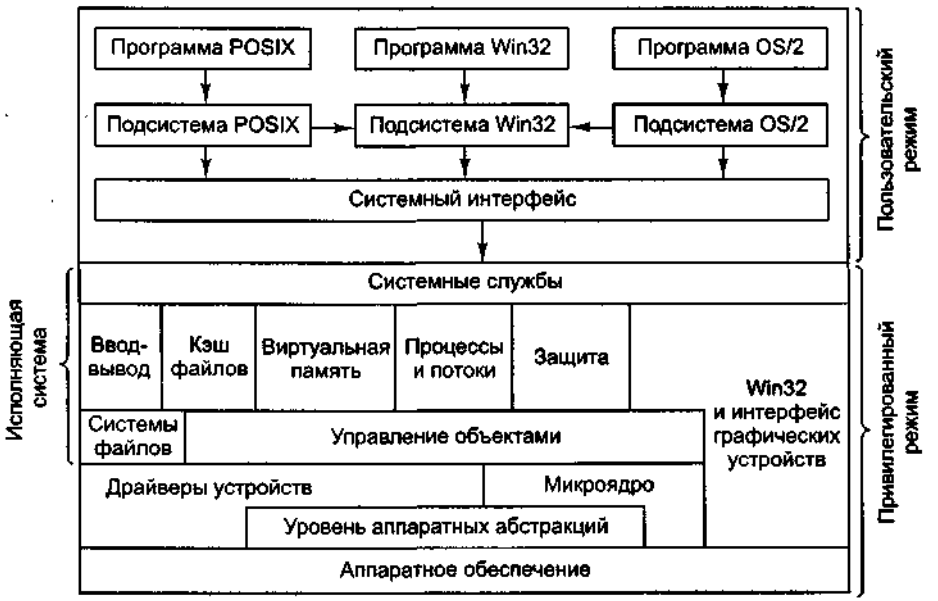



**Рис. 10.3.** Архитектура Windows NT

Безопасность страдает из-за применения динамической компоновки. В старых программах фрагменты программы соединялись вместе изготовителем (так называемая *линковка, сборка* — на программистском жаргоне) еще до того, как вы ее купили. Программисты связывали части программ и проверяли, что все работает как надо. Сейчас компоненты часто связываются динамически при запуске приложения. Пользователи Windows знают о так называемых библиотеках динамической компоновки (Dynamic Link Libraries, DLL); пользователям UNIX и Macintosh они известны как библиотеки коллективного доступа (shared libraries).

Проблемы безопасности нужно рассматривать одновременно с нескольких точек зрения. Во-первых, вы не можете быть уверены, что все модули надежны. В предыдущем разделе я рассказывал о «вредных» программах; возможно, что один или несколько модулей являются разрушительными или просто неисправными. Во-вторых, вы не должны допускать, что все модули написаны настолько хорошо, что будут работать во всех возможных конфигурациях. Достоинством крупных компьютерных программ было то, что они тестировались как единое целое. Браузер, работающий на вашем компьютере, со всеми дополнительными модулями, которые вы загружали в произвольном порядке, может быть совершенно уникален. Маловероятно, что это сочетание кто-то уже тестировал.

И в-третьих, еще не существует такой операционной системы, которая помогла бы решить две вышеуказанные проблемы. При старом принципе построения про¬

грамм различные части программного обеспечения сообщались только через операционную систему. Хорошая операционная система могла обеспечивать взаимодействие программ и не допускать, чтобы одна программа повреждала другую. Современные компоненты непосредственно обращаются друг к другу, без посредничества операционной системы, поэтому применяемые в последних меры безопасности могут не работать.

Эти проблемы безопасности пытались решать с применением некоторых общих методов, одни из которых принесли больший, другие — меньший успех. Но все эти подходы лучше выглядят в теории, чем работают на практике.

**Автономность (Isolation) и защита памяти.** Эта мера направлена на то, чтобы помешать одному из компонентов умышленно или случайно воздействовать на остальную систему: читать или вносить изменения в память другого компонента, выходить за пределы отведенной ему памяти и приводить к поломке системы или доставлять другие неприятности. Автономное использование памяти предполагает, что каждому компоненту отводится свой участок памяти, за пределами которого этот компонент не может ни читать, ни записывать. Время от времени контролирующие программы (program checkers), установленные на машине пользователя, проверяют коды компонентов, чтобы убедиться, что не происходит ничего недозволенного. Примером реализации этой идеи служит «песочница» (sandbox) Java: все компоненты вынуждены «играть» в отдельных «песочницах», из которых они не могут повредить друг друга. Этот принцип защиты работает хорошо, но некоторые ошибки он не позволяет обнаружить, кроме того, за него приходится расплачиваться скоростью работы программ.

**Контроль доступа в интерфейсе.** Сделав компонент полностью изолированным, мы не решаем проблему полностью: ведь ему необходимо взаимодействовать с другими компонентами (а также с экраном, клавиатурой, мышью и т. д.). На рис. 10.2 показаны пути взаимосвязей между компонентами. Устанавливая правила контроля доступа в точках соприкосновения, мы можем надеяться, что компоненты правильно взаимодействуют друг с другом. Проблема в том, что вы должны выбрать политику контроля доступа, которая должна быть достаточно жесткой, чтобы приносить действительную пользу. «Песочницы» Java позволяют добиться хороших результатов, однако недостаток их политики состоит в том, что она либо чересчур бескомпромиссная, либо излишне либеральная — «золотой середины» в реальности не существует. (Java 2 имеет мелкомодульный контроль, но он недостаточно используется.)

**Подписывание кода (Code signing).** Представьте себе закрытую частную вечеринку, попасть на которую реально, только предъявив какой-нибудь солидный документ (например, водительские права). При таком подходе в дом смогут пройти только друзья хозяина. Такой же смысл имеет подписывание кода. Программист подписывает отдельные компоненты. На основании этих подписей пользователь принимает решение, какие компоненты допустить на свой компьютер, а какие — нет. (В случае ActiveX подписывание кода — основной способ защиты от злонамеренного кода.) В своем сегодняшнем виде подписывание кода имеет массу проблем. Во-первых, непонятно, исходя из каких соображений пользователь должен решать, заслуживает ли доверия автор подписи. Во-вторых, сам факт существования подписи у компонента не означает, что он безопасен. В-третьих, то, что каж¬

дый из двух компонентов имеет подписи, не означает, что их можно использовать вместе: совместная работа чревата множеством непредвиденных опасных ситуаций. В-четвертых, к проблеме безопасности не следует подходить с позиции «все или ничего»: существуют разные степени безопасности. И в-пятых, фиксирование компьютером нападения (сохранение подписи кода) практически бесполезно с точки зрения атаки: при ее осуществлении злоумышленник может удалить или изменить подпись или просто переформатировать диск, на котором она была сохранена. Чем больше приходится размышлять о подписывании кода, тем меньше смысла видится в этой процедуре.

Новые технологии, зарождающиеся в университетских лабораториях, помогут когда-нибудь найти лучшее решение, но это произойдет не ранее, чем через несколько лет. Тем временем модульная программа, вероятно, станет еще большей проблемой безопасности. Все больше и больше программных пакетов обладают способностью к самоусовершенствованию, то есть регулярно загружают новые модули. Например, Internet Explorer 4.0 и последующие версии дают возможность подписаться на обновление программного обеспечения. Если вы этим пользуетесь, то программа автоматически обновляется, загружая новые модули с веб-страницы корпорации Microsoft. Это очень полезное свойство, если не пускать дело на самотек. В противном случае вы можете обнаружить, что среди ночи ваш компьютер автоматически подсоединяется к Интернету. Как может на это отреагировать пользователь, видно по следующему отрывку из программы новостей:

«Ранним утром я полез в холодильник, и вдруг услышал, что компьютер сам подключился к Интернету,— рассказал один пользователь, занимающийся бета-тести повредить его рабочим взаимоотношениям с Microsoft. — Я очень испугался и вытащил телефонную вилку из розетки».

Здесь нет ничего постыдного — пользователь просто не осознал, что происходит. Но большинство компьютерных пользователей не имеют ни малейшего представления, что происходит внутри их компьютеров. И если они привыкли, что их компьютер куда-то звонит среди ночи, они могут однажды с удивлением обнаружить, что какая-то гнусная программа резко увеличила сумму телефонного счета, позвонив по 900 номерам в Молдавии.

## Переносимый код

Если задуматься, использование программ, написанных кем-то другим, — это всегда риск. Вы принимаете на веру, что программист не злоумышленник и что про¬граммы, которые вы запускаете, работают так, как надо. (Мы снова вернемся к проблемам человеческого фактора в главе 17.) На заре существования вычислительной техники пользователи просто не имели возможности пользоваться чужими программами. Программы нужно было писать или, по крайней мере, компилировать для каждого компьютера особо.

Появление персональных компьютеров и таких программ, как VisiCalc, вынуло компьютеры из рук инженеров и поместило их на столы пользователей. Пользователи стали доверять готовым прикладным программам, они, не задумываясь,

запускали программу, не имея представления о ее содержимом, поскольку не обладали достаточной для этого квалификацией.

Я уже говорил о вирусах и троянских конях; они распространились потому, что люди обменивались копиями готовых программ (иногда нелегально), не беспокоясь о проблемах безопасности. Антивирусные средства помогли решить проблему, и на протяжении последних 20 лет пользователи безоговорочно доверяли программному обеспечению.

Однако с появлением Интернета эта прочно укоренившаяся вера стала создавать крупные проблемы.

В предыдущем разделе я объяснил, почему появление сетей сделало разрушительные программы более опасными. Там приведены и примеры переносимого кода и связанные с ними проблемы. К сожалению, существуют и более серьезные неприятности.

С появлением модульных программ все больше фрагментов программного обеспечения рассылается по Интернету. Сейчас любое ваше приобретение с большой вероятностью получено с веб-сайта: новый дополнительный модуль для браузера, драйвер для принтера, блестящая утилита или апплет на Java, который делает какие-то классные штучки. Поэтому вам следует задать себе следующие важные вопросы. Надежна ли эта программа? Надежен ли этот веб-сайт? Можно ли быть уверенным, что эта программа будет правильно взаимодействовать с остальным содержимым компьютера? И какая есть защита на случай, если эта программа окажется разрушительной? Редко встречаются пользователи, которые задаются подобными вопросами, но еще реже — те, которые могут на них ответить.

## JavaScript, Java и ActiveX

JavaScript, Java, ActiveX и загружаемые дополнительные модули — все они имеют различные модели защиты. Я по очереди расскажу о каждом из них.

JavaScript — это язык сценариев Netscape, при помощи которого можно вставлять фрагменты кода на веб-страницу. Его поддерживают все основные браузеры. Его сходство с Java исчерпывается первыми четырьмя буквами. При помощи программы на JavaScript можно проделывать обычные простые действия: открывать и закрывать окна, изменять формы на веб-страницах, настраивать параметры браузера и т. п. Все, чем докучают некоторые веб-сайты при попытке закрыть их страницы, — это тоже JavaScript[1].

Сам по себе JavaScript достаточно безвреден, но именно он лежит в основе множества атак всех видов, предпринятых за последние несколько лет. Приведу несколько примеров: 1997 год — контроль посещаемости сайтов пользователями; 1998 — чтение случайно выбранных файлов на машине пользователя; тот же 1998 — перехват адреса электронной почты пользователя. Для большинства этих атак тре-

[1] Как наследие эпохи войны браузеров, имеется две версии JavaScript. Исходная, созданная Netscape, и вариация Microsoft под названием JScript. Они практически идентичны, за исключением объектных моделей браузера (читай: веб-документа) и синтаксиса языка сценариев. Тем не менее далеко не все сайты умеют одинаково работать с обеими моделями. Формально для безопасности это угрозы не представляет, но кто знает... Чем сложнее система, тем она потенциально незащищеннее. — *Примеч. ред.*

буется слегка обмануть пользователя, заставив его сделать незначительную глупость, но это и не сложно. Недостатки в обеспечении безопасности браузеров, допускавшие такие атаки, были выявлены довольно быстро и устранены. Но регулярно обнаруживаются новые «дыры».

В ActiveX применяется защита при помощи подписывания кода. По сути, у каждого фрагмента кода ActiveX, называемого «control», проверяется цифровая подпись. (Microsoft вводит для этого понятие кода аутентичности.) После этого браузер открывает диалоговое окно и показывает пользователю имя программиста или компании, которые подписали код. Если пользователь соглашается принять код, программа загружается в браузер.

Любой подросток, к которому на вечеринку заявлялись случайные гости не лучшего сорта, знаком с этой проблемой. Безопасность системы полностью находится в руках пользователя. Как только код ActiveX проникает на машину пользователя, он может сделать все что угодно: переформатировать жесткий диск, изменить ваши однодолларовые ставки в тотализаторе на стодолларовые, собрать все ваши полные чувств любовные письма и послать их кинопродюсеру в Лос-Анджелес и пр. и пр.

Microsoft возразила, что подписи, конечно, будут указывать на автора, но это знание будет небольшим утешением человеку, у которого только что «рухнул» компьютер. Это все равно, что обязать преступников носить визитные карточки и перестать запирать двери на замок. «Мы сожалеем, что они вошли в ваш дом, съели всю вашу еду, сломали всю вашу мебель и унесли все ваши ценности. Но, по крайней мере, мы знаем, кто они». Причем в случае Интернета может быть такое уточнение: «Это были два подростка из страны, с которой у США нет договора об экстрадиции. Вам стало легче?» Кроме того, предполагается, что вы сможете определить, какая именно программа из десятков имеющихся у вас на жестком диске вызвала проблемы. Один исследователь показал, что при соединении двух безвредных средств управления ActiveX они способны стать разрушительными; и кто будет виноват в этом случае?

Существуют и более серьезные проблемы. В главе 17 подробно рассказывается о том, насколько сомнительно, что пользователи примут правильное и безопасное решение, здесь же достаточно сказать, что большинство людей не беспокоятся о том, каким средствам ActiveX можно доверять, а каким — нет. А это предполагает наличие инфраструктуры открытого ключа (PKI), поддерживающей подписи, о чем я еще буду сокрушаться в главе 15. Существует много возможностей обмануть PKI и заставить ее поверить, что средство управления подписано, когда это не так.

На самом деле ActiveX - это расширение старой системы компонентов Microsoft, так называемой DCOM. Это система, при помощи которой, например, Internet Explorer открывает и показывает таблицы Excel. Большинство используемых программами DLL на самом деле служат только транспортным средством для объектов DCOM. Explorer просто вставляет внутреннее содержимое книги Excel посредством DCOM и ActiveX. Это — невероятно мощная система, более гибкая, более доступная, более интересная архитектурно и невообразимо более опасная, чем аналогичные способы в других операционных системах.

В Java применяется совершенно иная модель. Это — просто язык программирования, специально разработанный для переносимого кода, создатели которого

не забывали и о безопасности. Программы Java, запускаемые при помощи веб-браузера, называются *апплетами,* и им для работы отводится определенный участок памяти, «песочница», которая позволяет ограничить возможные повреждения. «Песочницы» защищаются по трем схемам.

Во-первых, существует так называемый *верификатор байтового кода*. При каждой загрузке апплета Java верификатор байтового кода сначала проверяет программу. Верификатор гарантирует, что байтовый код имеет правильный формат и не создаст каких-либо общих проблем.

Во-вторых, имеется *загрузчик класса*. Этот компонент определяет, как и когда апплет может быть добавлен к среде Java. Его задача — убедиться, что апплет не заменит уже существующую важную программу.

И в-третьих, есть *администратор безопасности*. Это устройство чем-то похоже на мониторы обращений, о которых шла речь в главе 8; к нему обращаются всякий раз, когда апплеты Java пытаются проделать что-то потенциально опасное: открыть файл, установить связь с сетью и т. д. В зависимости от способа установки апплета эти операции будут или разрешены, или запрещены. (Например, апплет, загруженный по Сети, имеет больше ограничений, чем апплет, установленный на компьютер при покупке операционной системы.)

Модель «песочницы» слишком сложна, но это лучшее, что у нас есть в настоящее время. Последние версии Java имели две модификации — хорошую и плохую. В Java 1.1 реализовано подписывание кода, что роднит его с ActiveX. Апплеты, которым пользователи доверяют, могут выходить за пределы «песочницы» и без ограничений работать на машине пользователя. Нужно ли говорить, насколько при этом делаются актуальными все проблемы безопасности модели ActiveX?

В Java 2 усовершенствована модель «песочниц». Вместо подхода «все или ничего» — или в «песочнице», или за ее пределами — Java 2 обеспечивает большую гибкость модели безопасности. Апплеты получают именно такие полномочия, которые необходимы для выполнения их работы. Например, один апплет может иметь доступ к файловой системе компьютера, но не иметь сетевого доступа. Другому разрешен сетевой доступ, но запрещен доступ к файловой системе. Третий апплет может иметь только доступ к определенной части файловой системы. То есть каждому апплету предназначается своя «песочница». Такая система работает намного лучше, но, как оказалось, она сложна в использовании.

Дополнительные модули хуже всего, поскольку они автоматически считаются надежными. Это — программные модули, которые вы можете добавить к своему браузеру, чтобы расширить его функциональные возможности, например средства просмотра файлов формата PDF, медиапроигрыватели или что-нибудь еще. У них нет никакой системы безопасности. Если вы их загружаете и устанавливаете, значит, вы им доверяете. Точка.

# Безопасность Веб

HTTP (протокол, используемый в Веб), как и большая часть информации, блуждающей в Интернете, незашифрован и неаутентифицирован. Многие боятся доверять номера своих кредитных карт незашифрованной веб-связи. (Я не думаю, что

в этом есть смысл, но кое-что я бы посылать по Сети незашифрованным не стал.) Чтобы решить эту проблему, в ранние версии Netscape Navigator включали специальный протокол, так называемый SSL. Этот протокол, который был со временем переименован в TLS, обеспечивает шифрование и аутентификацию веб-связи. SSL довольно хорош, и все его проблемы касаются сертификатов и их применения (подробности — в главе 15). Некоторые веб-сайты предоставляют вам возможность выбрать защищенный SSL сеанс связи с браузером. (Веб-страница должна иметь этот параметр; браузер не будет использовать SSL, если на сервере нет соответствующих установок.) Браузер и веб-сервер применяют шифрование открытым ключом для обмена ключами и симметричное шифрование для кодирования данных. Присутствие в нижней части браузера зеленого ключа или желтого замка дает пользователю возможность почувствовать себя намного свободнее.

Однако нельзя упускать из виду, что пока пользователь не проверит вручную присланный сервером сертификат, он не имеет ни малейшего представления о собеседнике. Я повторяю, SSL устанавливает безопасную связь между браузером и кем-то на другом конце соединения. Если пользователь не проверит, кто находится на другом конце соединения, он не будет знать, с кем он секретничает. Представьте себе, что двое незнакомых друг с другом людей входят в абсолютно темную комнату со звуконепроницаемыми стенами. Они знают, что их разговор никто не подслушает. Но кто доверит свои секреты незнакомцу? Это одна из проблем, возникающих при использовании SSL-сертификатов.

Кроме того, SSL не обеспечивает защиты данных на сервере. В начале 2000 года хакеры неоднократно взламывали веб-сайты и крали информацию: номера кредитных карт, информацию о лицевых счетах и многое другое. SSL не в состоянии это предотвратить.

## Взлом URL

На унифицированный указатель информационного ресурса (URL) направлен целый ряд атак, некоторые из них рассчитаны на ошибку пользователя, а некоторые — просто на его необразованность. Первый класс — это атаки, при которых разные серверы крадут трафик друг у друга. Может быть, на ваш взгляд, это ерунда — зачем веб-сайту, продающему водопроводное оборудование, трафик веб-сайта финансовых новостей,— но некоторые сайты, например порнографические, таким образом хотят повысить посещаемость своих страниц.

Один из способов получить чужой трафик — обмануть поисковую машину. Машины поиска в большинстве своем довольно глупы: вы спрашиваете их о сайтах про водопроводное оборудование и получаете в ответ все веб-страницы, на которых где-то в тексте есть словосочетание «водопроводное оборудование». (Более современные машины поиска чуть умнее, но основной принцип сохранился.) Некоторые сайты имеют на своих страницах посторонний текст в качестве приманки для поисковой машины. Этот текст не отображается на экране — он может быть спрятан (например, белый текст на белой странице), может быть представлен в виде ключевых слов или мета-тегов в непечатаемой части страницы, — но он просматривается поисковой машиной. Так, на порнографическом сайте могут находиться слова: «котировки ценных бумаг; погода; выборы президента; Кливленд;

кулинария; садоводство», и при поиске по этим словам машина найдет, в числе прочих, этот порносайт.

Создатели некоторых веб-сайтов идут еще дальше, используя *переключатели страниц*. Подстраивая свои ключевые слова и мета-теги (встроенные в веб-страницы команды, которые сообщают поисковой машине параметры страницы), эти сайты обманывают поисковую машину, которая не отличает их от популярных веб-сайтов и при выводе результата помещает их непосредственно перед этими популярными сайтами. Ничего не подозревающие пользователи загружают поддельный сайт вместо настоящего. Главным образом, такую тактику применяли создатели порнографических сайтов, чтобы привлечь посетителей, но вы можете себе представить взлом по типу переключателя страниц, при котором поддельный сайт еще и выглядит как настоящий. Это было бы неприятной проблемой.

Атаки такого рода не ограничиваются веб-страницами и поисковыми машинами. Небольшие компании иногда включают в свои пресс-релизы название и эмблему более крупной компании, и в результате люди, которые ищут ту большую компанию, получат и этот пресс-релиз. Это называется *столкновением наименований (ticker symbol smashing)* и может выглядеть примерно так: «SmallCompany.com объявила, что ее новая программа не имеет ничего общего с Microsoft». Даже при описании лота на аукционе eBay используются слова, которые будут притягивать к нему поиск: «Этот дешевый свитер (не Прада, не Армани) красного цвета».

Возвращаясь к Веб, отмечу, что одним из способов подобных атак является регистрация сайтов, имена которых похожи на имена популярных сайтов. Этим занимаются *тайпсквоттеры («пираты» опечаток)*. Например, адрес wwwpainewebber.com (без точки, вместо www.painewebber.com) приведет кого-то на порнографический сайт. Люди, которые неправильно набрали название страховой компании (Geigo вместо Geico), окажутся на сайте, которым владеет Progressive Insurance. (Эти атаки, происходящие в результате опечаток пользователя, скорее всего, уже никому не повредят; во время написания книги проходили несколько судебных процессов как раз по такому поводу.)[1]

Подобные инциденты могут возникать и спонтанно. Компания eToys попыталась возбудить судебный процесс против группы артистов etoy, несмотря на то что доменное имя etoy.com было зарегистрировано за два года до того, как появилось eToys.com. (Хотя имена доменов действительно совпали случайно, оказалось, что etoy занимались переключением страниц на сайты Playboy.)

Все перечисленные выше случаи не относятся к *киберсквоттингу*. Этим термином обозначают регистрацию имени домена, которое может представлять интерес

[1] Тайпсквоттинг как способ имитировать заведомо посещаемый домен легален с точки зрения закона. В русском секторе Интернета он процветает. Можно перечислить тысячи интересных и забавных случаев, но не менее любопытна его разновидность, которую можно отнести к столкновению интересов. Это включение в описательные поля страницы или прямо в ее состав текста, рассчитанного на ошибки набора пользователей при запросах к поисковым системам. Здесь тоже можно говорить бесконечно, но я приведу нетривиальный пример. На сайте одного переводческого агентства есть страничка «помощи», полный текст которой гласит: «Преводов, перводов, пеерводов: Если Вы ошиблись, и вместо бюро переводов набрали в поиске бюро перводов или бюро пеерводов или бюро преводов или бюро переводо, то это не повод, чтобы не посетить сайт бюро переводов Flarus (www.flarus.ru)». И тут же ссылки на купить, заказать и т. д. На практике такие вещи срабатывают эффективнее первородного тайпсквоттинга. — *Примеч. ред.*

для кого-то. Например, кому-то другому, но не мне принадлежат имена applied¬ cryptography.com и applied-cryptography.com, а именно так называется моя первая книга[1].

*Веб-спуфинг* (Web-spoofing), или *получение доступа обманным путем,* — еще один вариант мошенничества в Интернете. Подделывая адреса URL на сайте клиента, злоумышленник может вынудить жертву всегда совершать поиск через определенный сайт. Этот сайт, которым владеет злоумышленник, может перехватывать весь сеанс поиска жертвы. Злоумышленник может сохранить записи о том, на какие сайты заходила его жертва, ее различные учетные записи, пароли и т. п. Злоумышленник может также слегка изменять различные страницы, например менять адрес отправления купленного жертвой продукта.

Такая атака возможна даже при SSL-связи. Как я отмечал ранее, SSL гарантирует только то, что канал связи недоступен посторонним. А если секретная связь установлена *со злоумышленником,* это мало поможет. Несколько других трюков облегчают нападение, поэтому выключение JavaScript в браузере дает некую защиту. Некоторые веб-сайты, например AskJeeves, усугубляют проблему, размещая у себя веб-страницы других людей и представляя их информацию как свою. В момент написания книги не было данных о существовании такого рода атак «в диком виде».

# Cookies

Cookies — это программная хитрость, встроенная в браузеры Всемирной Сети изобретательными программистами. По своей сути cookies — это небольшие порции информации, которую веб-сервер поставляет браузеру. Браузер сохраняет эти данные на компьютере пользователя и отсылает их на сервер всякий раз, когда браузер возвращается к серверу. Cookies позволяют сделать массу полезных и хороших вещей. Но, к сожалению, они также допускают множество разрушительных действий. В первую очередь я объясню, как они работают, а затем расскажу о проблемах, с ними связанных.

HTTP — по сути, протокол, не идентифицирующий пользователя. Это значит, что сервер не знает, кто с ним работает. Сервер просто обслуживает веб-страницы. Браузер запрашивает веб-страницу — сервер ему ее выдает. Сервер не имеет представления, тот ли это браузер, что и раньше, или другой — ему все равно. Такой

---

[1] В США с киберсквоттингом борются посредством Антикиберсквоттерского законодательного акта о защите потребителей (1999 год, не предусматривает наказания за тайпосквоттинг). В России право на имя регулируется с 10 апреля 2002 года документом «Регламент и тарифы на услуги по регистрации доменов второго уровня в зоне *.ru», согласно которому споры о доменных именах решает суд. На практике положительных сдвигов мало везде. Процессы вспыхивают в момент, когда какая-нибудь звезда, выходя в Сеть, обнаруживает, что домен уже занят. Таким образом отвоевали свои имена Мадонна, Джулия Роберте, Изабель Аджани, известные рок-группы и др. Ничего не вышло у Стинга из-за неудачного имени, так как суд посчитал, что это английское слово. Кибертайпсквоттер миллионер-владелец порносайтов Цуккарини проиграл ряд тяжб, в том числе вернул утраченное актеру Кевину Спейси. Но в запасе у него остались тысячи других зарегистрированных доменов. Кибер¬сквоттинг приобретает гигантские масштабы, так, например, рекорд по количеству доменов, осуществляющих переадресацию на один и тот же узел, принадлежит порносайту — 4525 имен на апрель 2002 года. — *Примеч. ред.*

подход замечательно работает для простых, статических веб-сайтов, содержащих только страницы с фиксированным содержанием.

Сложные веб-сайты имеют динамическое устройство. На веб-сайтах розничной торговли есть «корзины», с которыми вы перемещаетесь при просмотре сайта. На информационных сайтах с платным доступом имя пользователя и пароль требуется вводить при переходе со страницы на страницу. (На мой взгляд, утомительно набирать имя пользователя и пароль каждый раз, когда я хочу посмотреть еще одну статью на веб-сайте *New York Times.)* Cookies дают возможность управиться с этим.

Отправляя браузеру cookies, а затем запрашивая их обратно, сервер как бы вспоминает, кто вы. «А, конечно, вы — пользователь 12467, вот ваша корзина». Cookies позволяют браузеру добавлять возможность идентификации к протоколам Интернета. Это выглядит, как огромная рассредоточенная база данных, фрагменты которой хранятся в миллионах браузеров.

До сих пор я рассказывал о пользе cookies. Они, главным образом, полезны, если сервер, их разместивший, играет по правилам. Сервер устанавливает, сколько времени действителен cookie: хорошее значение — несколько дней. Сервер может установить ограничения на доступ к cookie. Сервер может ограничивать доступ к другим серверам в том же домене; это значит, что если ваш cookie пришел с inchoate-merchant.com, то только inchoate-merchant.com может иметь к нему доступ.

Проблемы возникают при нарушении правил игры/Некоторые серверы используют cookies, чтобы прослеживать пользователя от сайта к сайту, а некоторые с помощью cookies идентифицируют пользователя. Вот простой пример: компании занимаются перепродажей места для размещения рекламы на популярных сайтах. Одна из таких компаний — DoubleClick; именно через нее размещены многие объявления, которые вы видите на коммерческих сайтах. Если вы будете просматривать sex-site.com, вы увидите часть того окна, которое пришло с DoubleClick.com. DoubleClick.com предоставляет вам cookie. Позже (в этот же день или, может быть, в другой) вы будете просматривать CDnow.com, на котором DoubleClick разместила другую рекламу. DoubleClick может запросить cookie вашего браузера и установить, что он был создан в то время, когда вы посещали секс-сайт, после чего вам будет отправлена целевая реклама, хотя вы интересуетесь CDnow. Поскольку DoubleClick сотрудничает с целым рядом коммерческих сайтов, по ее cookies можно проследить пользователя на всех этих сайтах.

Еще более серьезные проблемы вас ожидают, если вы оставите свой электронный адрес на каком-то из этих сайтов, а они отошлют эту информацию в DoubleClick. Все, что от вас требуется — один раз набрать этот адрес, заказать одну-единственную вещь, и он останется у них навечно. (Или до окончания срока cookies, что может затянуться на годы.)

Такие действия не являются большим секретом. DoubleClick открыто признает, что собирает данные и использует их, чтобы направлять рекламу определенным пользователям. До 2000 года они отрицали создание идентификационных баз данных, но в конце концов признали это после сообщения в *USA Today.* С тех пор они отступили от идеи связывать cookies с именами и адресами. (Хотя, возможно, выплывут новые факты в результате какой-нибудь публикации.)

Идем дальше. Сайты способны послать вам по электронной почте cookie, при помощи которого они могут вас идентифицировать, если вы позже посетите этот сайт. Вот как это работает: сайт посылает вам сообщение в формате HTML. (Это предполагает, что вы пользуетесь программами для электронной почты, поддерживающими сообщения HTML; среди таких программ — Microsoft Outlook и Outlook Express, Netscape Messenger и Eudora.) Сообщение содержит уникальный URL, скрытый графикой, который сайт может использовать, чтобы послать вам cookie. Если URL имеет вид www.gotcha.com/track-cgi=schneier@counterpane.com/pixels.gif, значит, у них в cookie — ваш адрес электронной почты. Тогда, если вы просматриваете сайт в какой-то последующий день, сайт может по cookie определить ваш адрес электронной почты и отследить ваши перемещения в Интернете.

Сами cookies не умеют осуществлять активных действий. Они не могут похитить информацию из вашего компьютера. Cookies — это просто некоторые данные, которые сервер сообщает браузеру, а браузер позже возвращает. Cookies не могут украсть у вас пароли или файлы. (ActiveX, Java и JavaScript в этом отношении гораздо более опасны.) Cookies не в силах похитить номера ваших кредитных карт, но может оказаться так, что «глупые» сайты включат в cookie номер вашей кредитной карты.

Из всего вышесказанного можно сделать вывод, что cookies — по своей сути полезный инструмент, но при неумелом обращении они могут работать на злоумышленников. Это простейший способ, позволяющий веб-программистам контролировать взаимосвязи. Большинство браузеров допускают полное отключение cookies, можно купить дополнительные программы, позволяющие лучше управляться с ними. Хотя некоторые сайты — например, Hotmail и Schwab Online — не соединяются с браузерами, не принимающими cookies.

## Веб-сценарии

Мишенью всех атак, рассмотренных выше, является компьютер пользователя, а сейчас речь пойдет об атаке, направленной на сервер.

Общий шлюзовый интерфейс (Common Gateway Interface, CGI) — это стандартный способ, которым веб-сервер передает запрос клиента размещенным на нем приложениям и отсылает ответ пользователю. Если вы посылаете поисковый запрос на веб-сайт — например, на сайт розничной электронной торговли, — веб-сервер передает запрос в приложение базы данных и затем форматирует ответ перед тем, как предоставить его пользователю. Или если посетитель заполняет анкету на веб-странице, то эта информация передается в соответствующее приложение для последующей обработки. Иногда вы можете видеть команды CGI в адресной строке браузера — это непонятные значки и цифры в конце URL; в других случаях они не видны пользователю. Это часть HTTP, все ею пользуются. Сценарии CGI — это небольшие программы на веб-сервере, которые работают с данными. Например, они задают принцип обработки анкет на веб-страницах.

Сложности при работе со сценариями CGI в том, что каждый из них — потенциальная прореха в системе безопасности. Манипулируя сценариями CGI, можно совершать самые неожиданные веши. Вот ряд примеров (все они действительно имели место): загрузка файла с веб-сервера, просмотр всего содержимого базы дан¬

ных, загрузка списка покупателей и их персональных записей, кража денег у клиентов электронного банка, торговля чужими контрольными пакетами акций и просмотр системных журналов веб-сервера, в которых записаны договоры клиентов.

Другие похожие атаки проводятся путем помещения исполняемого кода (обычно сценарии Perl или код JavaScript) в текстовые поля. Таким путем можно заставить веб-сервер внести изменения в домашнюю страницу, показать секретный ключ SSL или доставить другие неприятности, о которых сказано в предыдущем абзаце. Этот метод также позволяет использовать переполнение буфера и другие ошибки программирования (см. главу 13), чтобы вызвать поломку веб-сервера или, что еще лучше, захватить этот сервер в свои руки.

Один пример: в 1998 году в результате атаки против Hotmail стало возможным увидеть учетные записи электронной почты других людей. eBay также подвергся атаке; злоумышленники поместили троянского коня, написанного на JavaScript, в поле описания товара. Это поле видел любой, кто просматривал товары, выставленные на продажу, и в результате злоумышленники получали информацию о тысячах учетных записей.

Один недостаток CGI позволил злоумышленникам загружать секретную персональную информацию с различных сайтов. Другие сценарии CGI использовались, чтобы взломать веб-сервер. В конце 1999 года были предприняты две атаки — атака *Poison Null («отравленный» нуль),* позволившая хакерам просматривать и изменять файлы на веб-сервере, и атака *Upload Bombing (бомбежка при пересылке),* наводнившая веб-серверы бесполезными файлами, которые очень быстро превращались в атаки-сценарии, так что любой мог при желании их использовать[1].

Включения на стороне сервера (Server Side Includes, SSI) — это указания для веб-серверов, встроенных в HTML-страницы. Непосредственно перед отправкой страницы браузеру веб-сервер выполняет все SSI, содержащиеся на странице, и помещает на нее результаты своей работы. Атаковать SSI так же выгодно, как и все остальное.

Можно атаковать уязвимые места в стороннем программном обеспечении: на специфических веб-серверах, в их приложениях. Сюда относятся приложения для баз данных, программы «корзины», сервер транзакций и другие. Эти атаки зависят не от того, как сайт использует приложение, а от самого приложения (СУБД Oracle, например). Злоумышленники могли бы загрузить исходную программу с веб-сервера, разрушить сервер, получить привилегии доступа на уровне администратора для входа на сервер, запустить на сервере произвольную программу и т. п. В отличие от прорех безопасности, вызванных сценариями CGI, установление уязвимых мест в дополнительных приложениях не находится под контролем сайта; это обязанность поставщиков стороннего программного обеспечения.

Существует множество похожих атак. Внося изменения в скрытые поля на некоторых веб-страницах (эти поля можно увидеть при просмотре начала страни¬

---

[1] Атаки образца конца 1999 года, актуальные и поныне (для подобных атак уязвимы до 80% интерактивных веб-сайтов). Upload Bombing работает на сайтах с запросом на загрузку файлов (агентства по трудоустройству, доски объявлений, почтовые серверы и т. д.). Смысл в том, что в ответ на предложение загрузки сервер заваливается множеством файлов, переполняющих диски, текстовых, графических, любых, и вызывает переполнение дисков. Название Poison NULL byte произошло от байта с нулевым значением, служащего ограничителем строки на многих языках программирования. — *Примеч. ред.*

цы), есть шанс взломать сценарии CGI и заставить некоторые программы «корзин» изменить цены предлагаемых товаров. (Вплоть до «назовите свою цену».) Некоторые атаки направлены на cookies: *порча cookie*. Злоумышленники входят на сервер и вручную меняют свои аутентификационные cookies на cookies других пользователей. Иногда эти cookies зашифрованы, но часто не очень надежно.

Некоторые атаки носят название *написания перекрестных сценариев*. Это довольно неудачное название: атаки заключаются не столько в написании сценариев, и речь не идет о пересечении. То, что они так называются, сложилось исторически. Суть в том, что Веб скрывает в себе множество мелких хитростей; когда вы смешиваете сценарии CGI, JavaScript, фреймы и cookies, cookies и SSL, итог может оказаться неожиданным и нежелательным. Использование различных платформ одновременно — это спорный путь, в результате нетрудно получить непредсказуемое взаимодействие различных компонентов сложных систем.

Такие атаки по нескольким причинам направлены преимущественно на сценарии CGI. Большинство сценариев CGI написаны непродуманно, и они широко распространены среди пользователей. Вы получаете набор сценариев вместе с программным обеспечением или от своего провайдера. Часто люди, занимающиеся написанием сценариев, не имеют опыта в программировании. Они не слишком хорошо разбираются в проблемах безопасности, которые могут возникнуть из-за применения сценариев или вследствие взаимодействия сценариев с другими частями программного обеспечения сервера. А веб-сервер не имеет возможности контролировать работу CGI-сценариев. Поэтому иногда сценарий создается для одной цели и, если он используется для другой, повреждает защиту.

Атаки CGI — мощные, против них мало что устоит. Конечно, можно написать безопасный CGI-сценарий, но едва ли кто-то это сделает. Одна компания занималась проверкой веб-сайтов на предмет недостатков в приложениях вроде CGI-сценариев — она не нашла ни одного сайта, который нельзя взломать. Это — стопроцентная уязвимость.

## Веб-конфиденциальность

Номинально веб-просмотр анонимен. В реальности существует много способов идентифицировать пользователя. Я уже говорил, что по cookies можно следить за перемещением пользователя с сайта на сайт и даже включить в cookie адрес электронной почты или другую личную информацию (если пользователь заполняет анкету или отвечает на электронное сообщение).

Вдобавок на большинстве веб-серверов все доступы регистрируются. В регистрацию обычно входит IP-адрес пользователя, время запроса, информация о том, какая запрошена страница, и имя пользователя (если оно известно из каких-нибудь регистрационных протоколов). Впрочем, большинство веб-сайтов просто выбрасывают эти регистрационные записи.

Конечно, IP-адрес пользователя — это все же не имя пользователя, но многие веб-браузеры установлены на машинах с единственным пользователем, напрямую подключенных к Всемирной паутине. Те пользователи, которым приходится дозваниваться, обладают большей анонимностью, чем пользователи с кабельным модемом или DSL-соединением (Digital Subscriber Line, цифровая абонентская ли¬

ния), но часто для идентификации достаточно установить провайдера. Например, в 1999 году неизвестный угрожал бомбой, отправив сообщение с регистрационной записи Hotmail. Электронная почта Hotmail содержит IP-адреса веб-браузеров, с которых отправляли почту. IP-адрес принадлежал America Online, и полиция, сравнив его и записи Hotmail, смогла проследить электронное сообщение до индивидуального пользователя America Online.

В данном случае для нарушения конфиденциальности были экстренные причины, но большинство таких действий можно автоматизировать. Коммерческие веб-сайты, как правило, не слишком стараются защитить конфиденциальность клиента. Фактически, многие из них делают деньги на рекламе. Другие сайты сознательно посягают на конфиденциальность посетителя, чтобы осуществлять направленную рекламу: это разные «цифровые бумажники», список компаний, производящих программное обеспечение и всякое другое. Многие компании рассматривают целевую рекламу как своего рода бизнес при помощи Интернета.

# Глава 11. Сетевая безопасность

Сетевая безопасность тесно связана с компьютерной безопасностью: в наши дни сложно отделить одно от другого. Все, от электронного замка на дверях отеля до сотового телефона и настольных компьютеров, присоединено к сетям. При этом надо иметь в виду, что, как ни тяжело создать надежный автономный компьютер, гораздо тяжелее создать компьютер, который надежен, будучи подключенным к сети. Последние более уязвимы: атакующий вовсе не должен находиться перед вашим компьютером во время нападения, а вполне может находиться на другой половине земного шара и атаковать его через Сеть. Сетевой мир, может быть, более удобен, но он намного менее безопасен.

В наши дни невозможно говорить о компьютерной безопасности, не затрагивая темы сетевой безопасности. Даже специализированные клиринговые системы кредитных карт работают, используя сети. Таким же образом работают сотовые телефоны и системы сигнализации. Автоматы в казино подключены к сети, как и некоторые торговые автоматы. Компьютеры ваших кухонных приборов скоро будут объединены в сеть, так же как и бортовой компьютер вашего автомобиля. В итоге все компьютеры будут подключены к сети.

Существует множество различных типов сетей, но я собираюсь посвятить большую часть времени обсуждению интернет-протокола TCP/IP. Кажется, все сетевые протоколы используются в Интернете, так что разумнее всего говорить именно о Всемирной Сети. Это не значит, что протоколы Интернета менее надежны, чем другие, — хотя, конечно, при их разработке мало думали о безопасности. Как будет ясно из дальнейшего обсуждения, наиболее фундаментальным является вопрос, что предпочесть — общеизвестный протокол, который долгое время атаковался хакерами и чья надежность, следовательно, постоянно улучшалась, или протокол малоизвестный и, возможно, менее безопасный. Помните об этом во время чтения данной главы.

## Как работает сеть

Компьютерные сети — это группы компьютеров, соединенных между собой. При этом компьютеры либо соединены физически — при помощи проводов офисной ЛВС, выделенной линии (возможно, ISDN или DSL), телефонной коммутации, оптического волокна, либо же используется электромагнитная связь — радио, высокочастотные волны и т. п.

Если один компьютер должен связаться с другим, он создает послание, называемое пакетом, с указанием имени компьютера-получателя, и отправляет его через сеть. Здесь обнаруживается фундаментальное отличие от телефонных переговоров. Когда Алисе хочется поговорить с Бобом, она сообщает компьютерной сети телефонной компании сетевое имя Боба (то есть его телефонный номер), и в сети замыкается цепь при помощи различных средств коммуникации — медных проводов, спутника, сотовой ячейки, волокна — всего, что может дать в итоге неразрывное соединение. Алиса и Боб общаются с помощью этой цепи до тех пор, пока кто-нибудь из них не повесит телефонную трубку. Тогда телефонная сеть разрывает это соединение, и другие люди получают возможность пользоваться такими же средствами для своих звонков. В следующий раз, когда Алиса позвонит Бобу, они будут соединены через совершенно другие звенья цепи. (По большей части другие, хотя телефонная линия и первичные коммутаторы будут теми же самыми.)

Компьютеры не используют цепи для переговоров друг с другом. Они не ведут разговоры, как люди, — они обмениваются небольшими пакетами данных. Эти пакеты могут содержать части самых разнообразных данных: посланий электронной почты, сжатых графических изображений обнаженных женщин, видео- и аудиопотоков, телефонных переговоров в Интернете. Для того чтобы облегчить передачу, компьютеры делят большие файлы на пакеты. (Представьте себе, что письмо объемом 10 страниц было отправлено по частям в 10 различных конвертах. Получатель вскрывает все послания и восстанавливает изначальный вид письма. При этом пакеты не обязаны прибывать в последовательности, соответствующей порядку расположения страниц, и приходить к адресату одними и теми же путями.)

Эти пакеты посылаются через сеть по маршрутам. Есть разные протоколы — Ethernet, TCP и другие, — но базовые принципы их работы одинаковы. Маршрутизаторы перенаправляют пакеты по указанным в них адресам. Они могут не знать точно местонахождения адресата, но имеют некоторые представления о том, в каком направлении следует отправить пакет. Это несколько напоминает почтовую систему. Почтальон приходит в ваш дом, забирает всю исходящую корреспонденцию и доставляет ее в местное почтовое отделение. Там могут вовсе не знать, где находится дом 173 по Питтерпат Лэйн, Фингербон, Айдахо, в котором проживает мистер X, но располагают сведениями, что конверт вместе со всей остальной корреспонденцией нужно погрузить в автомобиль, который едет в аэропорт. Служащие почты аэропорта также не осведомлены, где живет мистер X, но знают, что должны отправить письмо самолетом в Чикаго. В почтовом отделении аэропорта Чикаго знают, что должны переложить письмо в самолет, вылетающий в Бойсе. В почтовом отделении города Бойсе знают, что письмо нужно доставить к поезду, который идет в Фингербон. И наконец, на почте Фингербона имеют точную информацию, где находится указанный адрес, и почтальон доставит письмо.

## Безопасность IP

Нетрудно видеть, что любая сеть, построенная по этой модели, совершенно ненадежна. Рассмотрим Интернет. Так как пакеты проходят сначала по одному марш¬

руту, затем по второму, третьему и т. д., их данные, иногда называемые полезной нагрузкой, открыты для каждого, кто захочет их прочесть. Предполагается, что маршрутизаторы считывают только адрес получателя в заголовке пакета, но ничто не может помешать им просматривать и содержимое. Большая часть трафика (с использованием межсетевого протокола IP) проходит по немногочисленным высокоскоростным соединениям, составляющим скелет Интернета. Между отдаленными пунктами, находящимися, например, в США и Японии, пакеты проходят только по нескольким определенным маршрутам.

Хакеру сложно контролировать весь Интернет, но легко отслеживать, что происходит в небольшой части Сети. Все, к чему он стремится, — это получить доступ к отдельным компьютерам. Тогда он сможет просматривать все пакеты, проходящие через данный участок, в поисках интересующих его. Если он получает доступ к компьютеру, близко расположенному к компьютерам компании А, то, вероятно, сможет перехватывать все ее исходящие и входящие информационные потоки. (Конечно, имеется в виду «близко» в Сети, а не обязательно физически рядом.) Если ему не удастся проникнуть в компьютер рядом с компанией А, то он сможет перехватывать лишь малую часть информационного обмена этой компании (или не увидеть совсем ничего). Если он наиболее типичный хакер, которому все равно, какую компанию прослушивать, то это не имеет для него значения.

Пакеты с паролями внутри особенно интересны. *Выуживание пароля* легко осуществимо, это обычное нападение в Интернете. Нападающий устанавливает анализатор пакетов, позволяющий узнавать имена пользователей и пароли. Все, что делает программа, — это сохраняет первые две дюжины (или около того) символов, отправляемых в начале каждого сеанса. Эти символы почти наверняка содержат имя пользователя и пароль (обычно незашифрованный). А если он зашифрован, нападающий использует программу взлома паролей, а добытые пароли — для взлома других компьютеров. Это трудно обнаружить потому, что анализаторы паролей малы и незаметны. Это похоже на снежную лавину.

Возможно не только прослушивание, но также и активные нападения... на самом деле их даже легче осуществить. В большинстве систем связи значительно проще вести пассивное прослушивание сети, чем вставлять и удалять сообщения. В Интернете все с точностью до наоборот: подслушать сложно, а послать сообщение просто; любой уважающий себя хакер способен это сделать. Так как процесс передачи информации основан на пакетах, и они путешествуют многими различными путями, встречаясь лишь у своего адресата, легко незаметно подсунуть один пакет вместе с остальными. Многие нападения основаны на вставке пакетов в существующие каналы связи.

Это называется *подменой адреса (IP-spoofing)*[1] и легко осуществимо. В пакетах присутствует информация об источнике и адресате, но нападающий может изменить их как пожелает. Он может создавать пакеты, которые с виду прибывают от некоего отправителя, хотя на самом деле это не так. Компьютеры в Интернете не в состоянии проверить, соответствуют ли действительности сведения об отправителе и адресате; таким образом, если компьютер получает пакет, пришедший от

[1] Подмена адреса отправителя в заголовке IP-пакета с целью взломать аутентификацию, основанную на определении IP-адреса источника пакета. — *Примеч. ред.*

известного ему отправителя, которому можно доверять, то содержимое пакета также считается заслуживающим доверия. Нападающий может с выгодой использовать эти отношения взаимного доверия, чтобы внедриться в машину: он отправляет пакет, который будет выглядеть, как поступивший от проверенного компьютера, в надежде, что адресат, на которого нацелено нападение, поверит ему.

Это — атаки на маршрутизацию: нападающий сообщает двум узлам в Интернете, что кратчайший маршрут между ними пролегает через его компьютер. При этом легко можно прослушивать отдельные узлы. Данную тему можно продолжать и продолжать, многие книги уже написаны об атаках в Интернете.

Решение этих проблем выглядит очевидным в теории, но трудно осуществимо на практике. Если вы зашифровываете пакеты, никто не сможет прочесть их при пересылке. Если вы проверяете их подлинность, ни у кого не получится вставить дополнительные пакеты, которые имитируют адрес отправителя, а удаление пакетов не пройдет незамеченным и будут приняты меры к их восстановлению.

Фактически в Интернете уже реализуется шифрование пакетов. Программы типа SSH шифруют и аутентифицируют внешние связи пользователя с другими компьютерами через сеть. Протоколы типа SSL могут шифровать и подтверждать подлинность веб-трафика в Интернете. Протоколы типа IPsec, возможно, будут способны шифровать все и аутентифицировать всех.

## Безопасность DNS

Domain Name Service[1] (DNS) — система доменных имен (механизм, используемый в Интернете и устанавливающий соответствие между числовыми IP-адресами и текстовыми именами), — по существу, огромная распределенная база данных. Большинство компьютеров в Интернете — узлы, маршрутизаторы и серверы — имеют доменные имена, вроде brokenmouse.com или anon.penet.fi. Эти имена созданы для удобства запоминания и использования, например, в указателях информационного ресурса (URL) или адресах электронной почты. Компьютеры не понимают доменных имен; они понимают IP-адреса, наподобие 208.25.68.64. IP-адреса используются при отправке пакетов по сети.

DNS преобразует доменные имена в IP-адреса. Когда компьютер получает имя домена, он запрашивает сервер службы доменных имен для перевода этого имени в IP-адрес. Тогда он знает, куда послать пакет.

Проблема в том, что система службы доменных имен не имеет никакой защиты. Таким образом, когда компьютер посылает запрос серверу DNS и получает

[1] Сохранено написание оригинала, но в действительности во всех руководящих документах (RFC) используется слово «система» (system). Подмена одного слова другим стала настолько частой, что в лучших книгах по информационным технологиям употребляется именно service (служба) и соответственно переводится. Система доменных имен (DNS, Domain Name System) была разработана Питером Мокапетрисом и представлена в 1983 году в виде двух документов IETF. Позже этих документов стало огромное количество. Система DNS состоит из трех основных элементов: иерархического пространства имен («доменов»), серверов DNS для хранения имен поддоменов и распознавателей, генерирующих запросы для серверов DNS. Службой является последний элемент, а все в совокупности — системой. — *Примеч. ред.*

ответ, он воспринимает ответ как верный и сервер DNS как подлинный. Фактически при этом нет никакой гарантии, что сервер DNS не взломан. И ответ, который компьютер получает от сервера службы доменных имен, мог прибыть вовсе не с этого узла — он может быть сфальсифицирован. Если нападающий произведет изменения в таблицах DNS (фактических данных, которые переводят домены в IP-адреса и наоборот), компьютер будет всецело доверять измененным таблицам.

Несложно представить себе виды нападений, которые могут быть осуществлены при таком состоянии дел. Нападающий способен убедить компьютер, что ему можно доверять (изменив таблицы службы доменных имен так, чтобы компьютер нападающего выглядел как заслуживающий доверия IP-адрес). Нападающий в состоянии завладеть сетевым подключением (изменив таблицы таким образом, что кто-нибудь, желающий подключиться к legitimate.company.com, в действительности получит соединение с evil.hacker.com). Нападающий может сделать все что угодно. У серверов DNS есть процедура обновления информации: если один сервер службы доменных имен изменит запись, он сообщит об этом другим серверам DNS, и они поверят ему. Таким образом, если нападающий сделает изменения в нескольких точках, есть вероятность распространения этих поправок по всему Интернету.

В 1999 году было совершено такое нападение: кто-то взломал систему службы доменных имен для того, чтобы трафик к Network Solutions — так называлась одна из компаний регистрации доменных имен — был переадресован другим компаниям, занимающимся аналогичной деятельностью. Подобная же атака, рассчитанная на огласку, была проведена в 1997 году. Это случилось до того, как регистрация домена стала предметом конкурентной борьбы. Евгений Кашпурев, владелец альтернативного сайта AlterNIC, в качестве акции протеста перенаправил трафик Network Solutions на свой собственный сайт. Он был арестован, осужден и получил два года условно.

В 2000 году злоумышленники получили обманным путем доступ к таблицам службы доменных имен и присвоили домашнюю страницу RSA Security. Это не то же самое, что внедриться на веб-сайт и стереть страницу. Нападающий создает фальшивую домашнюю страницу и переадресует на нее весь трафик посредством манипулирования записями DNS. Хакер осуществляет вторжение, взламывая не DNS-сервер RSA, а серверы DNS в направлении, противоположном основному потоку в Сети. Умно и очень легко. Получение обманным путем доступа к записям службы доменных имен — это тривиальный путь взлома реального веб-сайта. И чтобы дела «похищенного» сайта обстояли еще хуже, взломщик вводит людей в заблуждение: они думают, что взломан веб-сайт компании A, в то время как на самом деле злоумышленниками контролируется сервер службы имен домена компании B.

Это серьезные проблемы, и они не могут быть легко решены. Аутентификация с использованием криптографии в конечном счете поможет преодолеть эти трудности, потому что больше не будет компьютеров, безоговорочно доверяющих сообщениям, которые якобы прибыли с сервера DNS. В настоящее время продолжается работа по созданию надежной версии системы DNS, которая справится с этими проблемами, но ждать придется долго.

# Нападения типа «отказ в обслуживании»

В сентябре 1996 года неизвестный хакер или группа хакеров атаковали компьютеры нью-йоркского интернет-провайдера Panix. Они посылали сообщения hello (пакеты синхронизации SYN) на компьютеры Panix. Обычно предполагается, что удаленный компьютер отправляет Panix приветственное сообщение, ожидает ответа и после этого продолжает сеанс связи. Нападавшие фальсифицировали обратный адрес удаленных компьютеров, так что Panix пытался синхронизироваться с компьютерами, которые в действительности не существовали. Компьютеры Panix 75 секунд ожидали ответа удаленного компьютера, прежде чем прервать связь. Хакеры «топили» Panix со скоростью более 50 сообщений в секунду. Это превышало возможности компьютеров Panix, что и привело их к аварийному отказу. Такое нападение называется *атакой синхронизации (SYN flooding)*.

Это был первый получивший огласку случай атаки на хосты Интернета, приводящей к отказу в обслуживании. С тех пор было предпринято много других. Атака, приводящая к отказу в обслуживании, — это особо вредоносная атака на коммуникационные системы, поскольку они разрабатывались именно для связи. В сети лавинная адресация запросов на установление связи является хорошим способом привести компьютер к аварийному выходу из строя. И часто не бывает возможности отождествления организатора этого нападения.

Существует возможность вызвать «отказ в обслуживании» обычного почтового сервиса: злоумышленник подписывает жертву на все каталоги почтовых заказов и на прочие издания, которые только могут прийти на ум. Жертва получает так много корреспонденции, допустим, 200 единиц в день, что шансы потери полезной почты среди ненужного хлама увеличиваются соответственно. Теоретически так обязательно и случится. Единственный способ воспрепятствовать этому нападению — ограничить количество рассылок ненужного хлама. А в Интернете почтовые серверы по определению рассылают почту. В 1995 году Фронт освобождения Интернета (Internet Liberation Front — это, скорее всего, вымышленное название, с тех пор о нем не было упоминаний) направил поток сообщений по электронной почте в журнал *Wired* и его автору Джошуа Квиттнеру. Поток был так велик, что компьютеры перестали работать.

Эта атака известна как *бомбежка почтой,* и она весьма эффективна. Отправьте кому-нибудь достаточное количество почтовых посланий, и его почтовый сервер будет принимать почту до тех пор, пока не «захлебнется». Наилегчайший путь сделать это — подписать жертву на тысячи почтовых рассылок. На дисках жертвы может не остаться места, сетевое подключение может перестать работать, или компьютеры могут прийти к аварийному отказу. И если вы замаскируете происхождение потока электронной почты, никто вас не поймает.

Есть другие нападения, приводящие к отказу в обслуживании. Некоторые нацелены на компьютеры, подобно только что описанному нападению на сервер электронной почты. Другие ориентированы на маршрутизаторы. Некоторые настроены на веб-серверы. Основная идея — та же самая: завалить цель таким большим количеством хлама, что он остановит ее работу. WinNuke может привести к аварийному отказу компьютеров с операционной системой Windows 95 и более ранней. Одиночное нападение в Интернете в апреле 1999 года вызвало сбой 6000 компьютеров с Windows 95.

Иногда может быть трудно отличить нападение, приводящее к отказу в обслуживании, от неправильных действий. Представьте себе городские магистрали. В нормальное время по ним можно ехать быстро. В часы пик там много заторов. Во время демонстраций они вообще закрыты. В 1999 году демонстрация против Всемирной торговой организации парализовала движение в центре Сиэтла, что, несомненно, можно рассматривать как атаку типа «отказ в обслуживании». Несколько раньше, когда пилоты American Airlines стали сказываться больными намного чаще обычного и у компании возникли проблемы с обслуживанием рейсов, это было менее очевидной разновидностью этой атаки. В 2000 году после выхода в эфир специальной телевизионной программы «Кто хочет выйти замуж за мультимиллионера» веб-сайты телеканала были сокрушены наплывом желающих войти и зарегистрироваться для участия в шоу. Было ли это нападением, приводящим к отказу в обслуживании?

Некоторые исследователи предложили средства защиты, вынуждающие клиента производить длительные вычисления, прежде чем он получит возможность соединения. Идея состоит в том, что если клиент должен потратить время на вычисления, то он не сможет «затопить» адресата многочисленными соединениями. Это хорошая идея, но она не будет работать против распределенных нападений типа «отказ в обслуживании», о чем мы поговорим в следующем разделе этой главы.

Некоторые полагают, что всему виной недостаточная аутентификация в Интернете. Это утверждение бессмысленно. Нападения типа «отказ в обслуживании» причиняют вред тем, что присылают пакеты; будут ли пакеты аутентифицированы или нет, никакого значения не имеет. Обязательная аутентификация никак не сможет предотвратить такие нападения или помочь в идентификации нападающих. Было бы иначе, если бы аутентификация могла быть проверена в каждой точке Сети. Это привело бы к изменениям в способе работы Интернета и значительно уменьшило бы пропускную способность сетей: вместо простой маршрутизации пакетов все коммутаторы и маршрутизаторы должны были бы еще аутентифицировать их.

Здесь может помочь крупномасштабная фильтрация сети интернет-провайдерами; если сеть в состоянии заблокировать нападение типа «отказ в обслуживании», оно никогда не достигнет цели. Здесь аутентификация может оказаться полезной. Но фильтрация интернет-провайдерами потребует больших усилий и значительно уменьшит сетевую пропускную способность. Подобным образом повсеместное использование коммутаторов и маршрутизаторов Интернета, в работу которых были бы внесены некоторые изменения, способствовало бы решению этой проблемы: они могли бы отказаться пересылать пакеты, которые явно подделаны. И опять-таки, это потребует значительных изменений.

И наконец, несмотря ни на что, с нападениями, которые попросту заваливают адресата потоками информации, ничего нельзя поделать. Некоторые нападения основываются не только на использовании лавинного эффекта: они также эксплуатируют определенные слабые места в защите; и они могут быть предотвращены, если будут заблокированы эти уязвимые точки. Но если у нападающего достаточно мощный «пожарный брандспойт», он может «затопить» свою жертву.

Нападения, приводящие к отказу в обслуживании, не являются вторжениями. Они не затрагивают данных веб-сайтов. Эти нападения не могут привести к завла¬

дению номерами кредитных карт или являющейся частной собственностью информацией. С их помощью нельзя перевести деньги с банковского счета или торговать акциями от чужого имени. Нападающие не в силах извлекать выгоду из своих нападений сиюминутно. (Они могут внезапно продать акции и после этого напасть на компанию.)

Это не говорит о том, что такие нападения не реальны или они не имеют значения. Для большинства крупных корпораций самый большой риск — это возможность потери дохода или репутации. И то и другое может быть изящно осуществлено при помощи блистательного нападения, приводящего к отказу в обслуживании. А если речь идет о компаниях, в режиме реального времени оперирующих данными, имеющими важное значение для решения критических задач или от выполнения которых может зависеть чья-то жизнь, нападения типа «отказ в обслуживании» буквально могут стать опасными для жизни людей.

## Распределенные нападения типа «отказ в обслуживании»

Распределенные нападения, приводящие к отказу в обслуживании, — это поистине опасная разновидность атак такого типа. Автоматические инструментальные средства для этих нападений были выпущены в 1999 году. Университет Миннесоты стал первой целью в августе того же года, а поток высококлассных нападений в начале 2000 года привел к тому, что сообщения о них попали на первые полосы всех газет.

Эти нападения точно такие же, как и традиционные нападения типа «отказ в обслуживании», только на сей раз нападение не имеет единственного источника. Нападающий сначала внедряется через Интернет в сотни или тысячи незащищенных компьютеров, называемых «зомби», и устанавливает программу атаки. После этого он координирует все эти машины для одновременного нападения. Поскольку цель подвергается нападению сразу из многих мест, ее традиционная защита не работает, и она падает «замертво».

Сказанное можно проиллюстрировать на примере доставки пиццы: Алисе не нравится Боб, и она обзванивает сотни компаний, доставляющих пиццу на дом, и просит доставить пиццу Бобу домой в 11 часов утра. В 11 часов подъезд Боба заполнен сотней разносчиков пиццы, которые требуют от него оплаты доставки. Бобу кажется, что за ним охотится мафия пиццы, но ее разносчики тоже жертвы. Настоящий же виновник происшествия никому не будет известен.

От этой атаки защититься невероятно сложно, если вообще возможно. При традиционном нападении, приводящем к отказу в обслуживании, компьютер жертвы может оказаться способным выяснить, куда ведут следы, и закрыть соответствующие соединения. При распределенном нападении такого единственного источника нет. Компьютер может закрыть все соединения, кроме заслуживающих доверия, но это не подходит для открытого сайта.

В последние годы проводилось несколько научных конференций по проблеме распределенных нападений типа «отказ в обслуживании», и по общему мнению никакой глобальной защиты от них не существует. Отчасти помогает непрерывный контроль

сетевого подключения, поскольку дает возможность переключаться на дублирующие серверы и маршрутизаторы. Иногда отдельные ошибки, которыми воспользовались для нападения, могут быть исправлены, но многие неисправимы. При разработке Интернета не были предусмотрены меры противодействия этому классу посягательств. Со временем это нападение, вероятно, станет еще более опасным.

Современные средства, обеспечивающие распределенные атаки типа «отказ в обслуживании», требуют от нападающего способности внедриться в большое количество машин, установить зомбирующие программы, принять меры к тому, чтобы эти программы не были обнаружены, и скоординировать нападение... и на каждом этапе он должен оставаться не пойманным. В качестве новейших инструментальных средств могут использоваться вирусы или троянские кони. Они могут распространить «зомбирующие» программы и затем автоматически начать нападение с помощью некоего кодового слова в открытом форуме.

Уже было одно нападение типа «отказ в обслуживании», которое произошло по подобному сценарию. В 1999 году кто-то отправил по почте фальшивое обновление Internet Explorer, якобы исходящее от Microsoft. Это был настоящий троянский конь, который заставил инфицированный компьютер посылать пакеты компьютерам болгарской телекоммуникационной компании, что на долгое время создало для нее проблемы, связанные с отказами в обслуживании.

Отслеживание нападающего также невероятно трудно. Возвращаясь к примеру с доставкой пиццы, можно сказать: единственное, что может предпринять жертва — это попросить разносчиков пиццы помочь ему поймать злоумышленника. Если бы все они сверили свои регистрационные записи заказов по телефону, возможно, удалось бы определить, кто заказал доставку всех пицц. Иногда подобное возможно в Интернете, но маловероятно, что промежуточные сайты аккуратно ведут журнал регистрации. К тому же легко скрыть свое местоположение в Интернете. И если нападающий действует из некоей восточно-европейской страны с неразвитым законодательством об уголовном преследовании компьютерных преступлений, продажной полицией, не участвующей в международных соглашениях о выдаче преступников, то в этом случае ничего нельзя поделать.

Настоящую проблему представляют сотни тысяч, возможно, миллионы ничего не подозревающих компьютерных пользователей, которые уязвимы для этого нападения. Они пользуются цифровыми абонентскими линиями или кабельными модемами, в Интернете им всегда предоставлены статические IP-адреса, и они могут быть использованы для запуска этих (и других) нападений. Средства массовой информации освещают нападения на крупные корпорации, но действительно заслуживают внимания именно индивидуальные системы.

Подлинное решение состоит в привлечении разнообразных средств «общественной гигиены». Подобно тому как малярия была побеждена в Вашингтоне (округ Колумбия) путем осушения всех болот, есть только один-единственный способ предотвращения этих нападений — это защита миллионов частных компьютеров в Интернете. К сожалению, «болото» разрастается с невероятной скоростью, и обезопасить всех невозможно. Даже если бы 99 % индивидуальных пользователей установили брандмауэры, которые все функционировали бы отлично, все равно в Интернете еще осталось бы достаточное количество незащищенных компьютеров, которые можно было бы использовать для подобного рода атак.

# Будущее сетевой безопасности

В шестидесятые годы люди поняли, что телефонные коммутаторы могут срабатывать, если свистеть, щелкать, рыгать в телефон. Это была эра телефонного жульничества: «черных» ящиков, «синих» ящиков, свистков капитана Кранча (Captain Crunch). Телефонные компании старались как могли защититься от мошенничеств: они блокировали определенные тоны, выслеживали жуликов и стали хранить в тайне технические подробности, но основная проблема состояла в том, что телефонная система была построена с передачей сигналов внутри полосы: контрольный сигнал и сигнал данных передавались по одному и тому же проводу. Это означало, что коммутаторы телефонной системы получали сигналы управления по тому же проводу, по которому велись переговоры, чем и пользовались телефонные жулики.

Телефонную систему решено было полностью перепроектировать. Современные протоколы телефонной коммутации, например SS7, или Система сигнализации 7, были разработаны с *передачей сигналов вне полосы*. Голосовые сообщения и сигналы управления были разделены, и стали передаваться по отдельным каналам. Сейчас не имеет значения, как сильно вы свистите в телефонную трубку: коммутаторы не слышат. Целые классы нападений просто не работают потому, что нападающие в конечных точках не имеют доступа к коммутатору посередине.

(Это не совсем так. «Красные» ящики все еще работают против таксофонов. Они подражают звону монет, опускаемых в телефон, это является пережитком использования в телефонной системе передачи сигнала внутри полосы: коммутатор получает сигнал по тому же звуковому каналу внутри полосы.)

В долгосрочной программе передача сигнала вне полосы является лучшим способом для того, чтобы исключить многие «слабые места» Интернета. Она не является панацеей — ненадежные узлы еще будут создавать проблемы, — но имеет перспективы.

К сожалению, есть несколько проблем. Интернет был спроектирован как равноправная сеть: любой может «передвигаться» по нему путем простого соединения с другим компьютером. Система сигнала вне полосы должна будет управляться централизованно, как телефонная система. Будут иметься конечные точки и внутренние маршруты, и они будут различными. Эта система не имеет ничего общего с сегодняшним Интернетом.

В настоящее время нет никаких планов по перепроектировке Интернета в соответствии с этой концепцией, и любое подобное мероприятие может быть слишком сложным, чтобы даже обсуждать его.

# Глава 12. Сетевые защиты

## Брандмауэры

Брандмауэры впервые появились на поездах. У паровозов, топившихся углем, в машинном отделении топливо находилось поблизости от топки. Машинист лопатой бросал уголь в топку. При этом образовывалась легко воспламеняющаяся угольная пыль. Время от времени она вспыхивала, и в машинном отделении возникал пожар, который мог перекинуться на пассажирские вагоны. Так как гибель пассажиров сказывалась на доходах железной дороги, паровозы стали оборудовать железными переборками позади машинного отделения. Они препятствовали распространению огня на пассажирские вагоны, но не защищали машиниста, находившегося между углем и топкой. (Есть над чем призадуматься системному администратору.)

В цифровом мире брандмауэр — это средство защиты внутренней компьютерной сети компании от злонамеренных хакеров, алчных преступников и прочих негодяев, которые блуждают по Интернету.

Термины не всегда точны: дело в том, что понятие «брандмауэр» изменило свое содержание с тех пор, как оно появилось в компьютерных сетях. Первые сети были очень несовершенны и могли быть легко разрушены. Брандмауэры были созданы для того, чтобы препятствовать распространению сетевого программного обеспечения, содержащего множество ошибок, на всю сеть с одного ее участка. Подобно физическим брандмауэрам, они были средством локализации пожара в месте его возникновения.

Сегодня брандмауэры выступают в роли защитников границ между локальными сетями и огромной глобальной сетью. Они оставляют снаружи, незваных гостей и впускают внутрь только полномочных пользователей. Правильнее было бы называть их «крепостными стенами», но термин «брандмауэр» уже устоялся.

Я не собираюсь вести разговор о тонкостях устройства брандмауэров и о том, как они работают; об этом написаны многие тома. Я намереваюсь рассказать об общей концепции брандмауэров, о том, насколько эффективно они противодействуют угрозам и какое у них будущее.

Первое: брандмауэр — это граница, линия обороны. Подобно стенам замка он служит для отражения нападений. Точно так же, как и стены замка, он бесполезен против вооруженного мятежа внутри. Билл Чесвик дал брандмауэру следующее определение: «твердая скорлупа, окружающая мягкое ядро». Как только нападающий преодолеет брандмауэр, последний станет бесполезным. И поскольку (согласно исследованиям Института компьютерной безопасности в 1998 году) около 70% всех нападений происходит внутри сети, об этом определенно стоит подумать.

Конечно, можно установить внутренние брандмауэры для дополнительной защиты сегментов сети. Представьте себе замок с внешним и внутренним дворами.

Второе: до изобретения артиллерийских орудий хороший замок был неуязвим; не было возможности забраться на его стены по лестнице, проломить их или сделать под ними подкоп. Однако терпеливый военачальник всегда может устроить осаду замка. Он надеется, что, лишив жителей продовольствия, воды и отрезав от внешнего мира, заставит защитников капитулировать. Иногда это действовало быстро, но некоторые осады продолжались годами. Если в стенах замка был колодец, это давало больше шансов его защитникам. Если из него были прорыты секретные тоннели наружу, они очень помогали. Если у жителей замка начиналась чума, замок не мог помочь им. (Антисанитария победила многих отважных защитников.) Точно так же можно морить голодом сеть, разорвав ее соединения с внешним миром.

Третье: замок должен быть защищен со всех сторон. Не имеет смысла воздвигать отдельно стоящую стену, нападающие просто обойдут ее. Помните Линию Мажино? Франция выстроила ее, чтобы предотвратить немецкое вторжение. Это было сделано, когда была жива память о траншейных боях Первой мировой войны, и подобные укрепления казались неприступными. Но в последующие годы развитие производства танков изменило подходы к ведению войны, и немцы изобрели Блицкриг. Они просто обошли вокруг Линии Мажино, вторгшись во Францию с территории Бельгии. Точно так же брандмауэр должен служить барьером между внутренней сетью и *всеми* внешними точками доступа. Иначе нападающий просто обойдет брандмауэр и атакует какое-нибудь незащищенное подключение.

И наконец, четвертое: замкам нужны ворота. Бесполезно и глупо строить замок, в который никто не может проникнуть ни при каких обстоятельствах: даже королям иногда нужно выходить на прогулку. Торговцы, курьеры, даже обычные горожане должны иметь возможность свободно входить в замок и выходить из него. Следовательно, в замках были привратники, чьей обязанностью было пропускать людей, желающих войти, или, наоборот, не пускать их.

Великая Китайская Стена не произвела впечатления на Чингисхана. Ему приписывают слова: «Неприступность крепости зависит от отваги ее защитников». Пропускать все, что нужно, и при этом оставлять все, что может представлять опасность, снаружи является главной задачей любого компьютерного брандмауэра. Он должен действовать как привратник. Он должен выяснить, какой код представляет опасность, и не пропустить его. Он должен делать это без необоснованной задержки движения. (Для среднестатистического пользователя Интернета необоснованной задержкой является всякое заметное замедление.) Брандмауэр должен делать это, не раздражая законного пользователя. (Среднестатистический пользователь Интернета не способен отказаться от чего-либо наподобие загрузки новой интернет-игры от компании с названием «Подозрительное программное обеспечение» или прочтения электронной почты от ненадежной машины.) Но если привратник — брандмауэр — допустит ошибку, хакер может проникнуть внутрь сети и стать ее полновластным хозяином.

Есть три основных способа преодолеть брандмауэр. О первом я уже говорил: просто обойти его с другой стороны. В обширной сети много соединений. Большие фотокопировальные устройства часто обладают возможностью подключения к Интернету; в сетевом оборудовании обычно присутствуют порты обслуживания

модемной связи. Часто компании подключают свои сети к сетям поставщиков, клиентов и т. п.; иногда намного менее защищенным. Сотрудники компании могут использовать модемы, чтобы иметь возможность работать дома. Одна супружеская пара из Силиконовой Долины периодически работала, не выходя из дома. Муж проверял свою электронную почту, в то время как жена занималась программированием, их компьютеры были объединены в маленькую домашнюю сеть. Внезапно компьютеры его компании начали обнаруживаться в сети ее компании, и наоборот.

Второе, и более сложное нападение — это украсть что-нибудь через брандмауэр. Чтобы сделать это, вы должны обмануть брандмауэр, чтобы он думал о вас, что вы хороший, честный и уполномочены сделать это. В зависимости от того, насколько хорош брандмауэр и насколько хорошо он был установлен, это просто, сложно или почти невозможно.

Основная идея в том, чтобы создать образец кода, который брандмауэр пустит внутрь сети. Код предназначен для того, чтобы использовать некий дефект в компьютерной системе, который позволит установить соединение между хакером снаружи брандмауэра и компьютером внутри него. Если все это срабатывает, хакер попадет внутрь.

Третье нападение — попытка захватить брандмауэр. Это похоже на подкуп и шантаж привратника. Как только вы сумеете заполучить его в свои ряды, он будет делать все, что вы пожелаете. Насколько это будет просто, зависит от брандмауэра. Некоторые брандмауэры работают с программным обеспечением, имеющим уязвимые места, что может помочь злоумышленникам. Некоторые используют в высшей степени ненадежные операционные системы, которые сильно облегчают задачу атакующим.

Так или иначе, разработка брандмауэра сегодня включает все, что нужно для проектирования умного привратника. На самом простом уровне брандмауэр — это маршрутизатор с последовательным набором правил, проверяющий сетевые потоки, проходящие через него, и регулирующий движение в соответствии с правилами. Примером могло бы послужить ограничение движения, основанное на адресе источника, или на адресе назначения, или на типе протокола.

Раньше это было относительно легко, но сегодняшним брандмауэрам приходится иметь дело с мультимедийным трафиком, загружаемыми программами, ап¬плетами Java (переносимыми программами на Java, распространяемыми через вебстраницы) и другими видами разных непонятных вещей. Брандмауэр должен принимать решения, имея в наличии только неполную информацию: он должен решить, можно или нельзя пропустить пакет, до того как просматривать все пакеты в передаче.

Ранние брандмауэры представляли собой пакетные фильтры. Брандмауэр просматривал каждый пакет, после чего пропускал или тормозил его в зависимости от того, соответствует ли заголовок пакета своду правил, которые известны брандмауэру. Первые пакетные фильтры были достаточно «неразумными» и позволяли попадать внутрь целым партиям пакетов, которые было бы лучше оставить снаружи. Со временем они стали умнее.

Сегодня они изменяют параметры своего состояния в процессе исполнения: вместо индивидуального просмотра каждого пакета брандмауэры хранят инфор¬

мацию о состоянии сети и о том, какие типы пакетов ожидаются. Тем не менее у брандмауэров не такая уж долгая память, и медленные нападения зачастую могут пройти.

Сейчас есть достаточно хорошие фильтрующие брандмауэры, но они по-прежнему имеют множество недостатков. Первый и самый главный заключается в том, что они трудны для правильного конфигурирования, а неподходящая конфигурация часто ведет к уязвимости защиты. Многие вещи, которые должны бы блокироваться, допускаются внутрь по умолчанию. Брандмауэры не изменяют пакетов; таким образом, если пакет проходит внутрь, он может сделать все что захочет. Есть группа скрытых нападений на пакетные фильтры: только вообразите бестолковую охрану, которая старается остановить поток опасных писем в замок, разглядывая конверты.

Другой тип брандмауэров — это прокси-брандмауэр, или система, выполняющая преобразование из одного естественного формата в другой. Подумайте о двух стражниках, один из которых находится снаружи перед стеной замка, а другой внутри, за стеной. Стражник снаружи не знает ничего о внутренней части замка. Стражник внутри ничего не ведает о мире за стеной замка. Но они переправляют друг другу пакеты. Прокси-брандмауэры могут служить иллюстрацией этой метафоры. Некоторые прокси-брандмауэры работают только в качестве посредников: если кому-то, находящемуся внутри охраняемой брандмауэром области, необходим документ «из внешнего мира», программное обеспечение клиента спрашивает у брандмауэра (внутреннего стражника) об этом, и брандмауэр (наружный стражник) соединяется с нужным веб-сайтом и получает требуемый документ. Другие прокси-брандмауэры понимают, какие правила и виды протоколов они используют. Существуют прокси-брандмауэры с промежуточным накоплением — они хранят куски данных между передачами и могут фильтровать данные, основываясь на своде правил. Лучшие прокси-брандмауэры узнают свое окружение, и поэтому способны принимать более умные решения относительно пакетов.

Слабые места прокси-брандмауэров — слишком тонкая тема, чтобы обсуждать ее здесь. У них более длительные задержки и более низкая производительность, чем у пакетных фильтров. (Фактически, так как брандмауэры должны исследовать каждый пакет, все они замедляют быстрое сетевое соединение.) Для безошибочной работы брандмауэры-«заместители» должны быть надежно конфигурированы, так же как и пакетные фильтры; в то же время они намного сложнее для конфигурирования и обслуживания; тенденция состоит в том, чтобы прекратить беспокоиться о них.

На рынке представлено около 100 разновидностей брандмауэров, и каждый месяц большое их количество покупается. Большинство из них приспособлены только под протокол IP и не обеспечивают безопасность других протоколов. Чаще всего функционирование брандмауэра основывается не на каком-то одном принципе, а на смешанных технологиях. Прогресс в технологии брандмауэров происходит постоянно и их сложно сравнивать и оценивать. Некоторые организации одобрительно отзываются об эффективности брандмауэров, но большинство хакеров считают это смешным; брандмауэры, которые просматривают пакеты, надежны только против большинства простых нападений. Вообще говоря, лучшие брандмауэры те, которые были правильно конфигурированы и вовремя обновляются.

Я слышал, что к брандмауэрам иногда относятся как к усовершенствованному маршрутизатору. Это верно. Некоторые из лучших профессионалов, хорошо знакомые с работой брандмауэров, даже не принимают их во внимание; они уверены в том, что хорошо конфигурированный маршрутизатор с высокой надежностью в конечных пунктах обеспечивает большую степень безопасности, чем брандмауэр. У них могут быть для этого основания. Конечно, брандмауэры часто создают ложное ощущение безопасности в Интернете.

Брандмауэры являются важной частью системы компьютерной безопасности любой компании, но они не могут обеспечить защиту полностью. Их модель защиты отражает ранние времена в сетевой безопасности, когда организациям нужно было обеспечить сохранность своего имущества и оставить «плохих парней» снаружи. Сегодня, когда ради успеха в торговле приходится открывать сети для покупателей и партнеров, брандмауэры кажутся анахроничными. Они могут играть важную роль, но не являются панацеей.

## Демилитаризованные зоны

ДМЗ — это демилитаризованная зона. Примером тому является ничейная земля между Северной Кореей и Южной Кореей, на которую не претендует ни одна сторона.

После всего сказанного о брандмауэрах вы понимаете, что именно в ДМЗ вы должны поместить свои общественные службы. В главе 10 я рассказывал обо всех видах нападений на веб-серверы. Вам не захочется поместить веб-сервер внутрь охраняемой брандмауэром территории, уязвимой для нападения. Вы не можете поставить веб-сервер снаружи брандмауэра, потому что тогда он будет еще более уязвим. Решение состоит в том, чтобы поместить его в ДМЗ.

Это хорошая идея, и она имеет исторические прецеденты. Замки часто имели внутренние и внешние стены. За наружной стеной укрывались жилища слуг, которыми можно было пожертвовать во время штурма. За внутренними стенами находилась резиденция знати, имевшая большую ценность. Защитники старались отстоять внешние стены, но у них оставалась возможность отступить к внутренним.

Для того чтобы выстроить ДМЗ, вам нужно два логических брандмауэра. Один из них защищает ДМЗ от внешнего мира. Другой имеет конфигурацию, которая предоставляет большие возможности для защиты внутренней сети от вторжений из ДМЗ. Результат состоит в том, что эта часть сети становится менее доступной и более защищенной. И эта идея работает.

## Частные виртуальные сети

Частная виртуальная сеть (VPN) является просто безопасным соединением через открытую сеть. В прежние времена, если Алиса и Боб хотели связаться, они должны были потратиться на частную линию и организовать свою собственную частную сеть. Сегодня Алисе и Бобу выгоднее использовать общедоступную сеть. Но Ин¬

тернет не обеспечивает безопасность связи, Алисе и Бобу следует защитить это соединение. Им нужно создать виртуальную частную сеть на основе физической открытой сети. Это и есть VPN.

У виртуальных частных сетей есть два главных применения. Первое — соединить два удаленных отрезка одной и той же сети. У корпорации может быть два офиса в различных частях планеты. У каждого офиса есть собственная сеть, и две сети соединяются посредством организации VPN через Интернет. VPN более приватна, чем «частная линия», обеспеченная телефонной компанией.

Второе применение состоит в том, чтобы соединить мобильных пользователей — работающих из дома и из номеров отелей. Старый путь для введения этих пользователей в большую открытую сеть заключался в том, чтобы подсоединить их непосредственно к сети компании, часто при этом приходилось производить очень удаленное сетевое подключение. Это дорого и вынуждает компанию обслуживать огромный банк модемов. Современный путь — в том, чтобы пользователи дозванивались до местного интернет-провайдера и затем с его помощью через Интернет подключались к компании. Для защиты этого соединения организуется VPN.

Различные VPN обеспечивают защиту, используя различные протоколы шифрования. Наиболее распространен протокол IPsec, хотя вам будут попадаться и другие протоколы, в частности PPTP и L2TP. В некоторых VPN вообще не используется шифрование.

Можно считать, что виртуальная частная сеть образует брешь в брандмауэре. Кто-нибудь, обладающий доступом к VPN, имеет возможность проникнуть через брандмауэр в сеть. И многие взломщики пользуются данной уязвимостью. По этой причине большинство систем безопасности, прежде чем разрешить соединение с VPN, проверяют, с кем имеют дело.

## Системы обнаружения вторжений

Системы обнаружения вторжений (IDS) — это сетевые контролеры. Они пристально наблюдают за вашей сетью, выискивая нечто подозрительное. Их можно сравнить с детективами, рыскающими по городу в поисках преступника: они знают, что должно вызывать подозрение — это может быть исследование возможности доступа к системе или попытки обнаружить и использовать ошибки, и они внимательно следят за этим. Они знают, как выглядит нападение. Они знают, как выглядит преступление. Маркус Ранум сравнивал брандмауэр со шлемом и бронежилетом, который вы носите в сражении, а систему обнаружения вторжений — с санитаром, который смотрит на ваше кровоточащее тело и говорит: «Видимо, это проникающее ранение грудной клетки. Это нужно проверить». IDS не являются заменой действенной защиты.

Что же делают системы обнаружения вторжений? Они информируют вас о свершившемся нападении или, возможно, даже о его развитии. Системы, которые работают хорошо, точны: они не воют волком и не заявляют о нападении там, где его нет, в то же время они не пропустят момент, когда оно случится. Хорошие системы реагируют своевременно: они предупреждают вас о нападении в то время, когда

оно еще происходит. Они ставят диагноз: на что направлена атака, откуда она исходит, и предлагают лечебное средство.

Современные системы обнаружения вторжений имеют много недостатков, но они постоянно совершенствуются. Самая сложная проблема — ложная тревога. Для объяснения этого момента понадобится напомнить статистику и показать, как вычисляется вероятность ошибки.

Предположим, у врача есть тест определения болезни с точностью 99 %. Таким образом, если кто-то болен, то есть вероятность в 99 %, что тест определит это, а если кто-то здоров, то есть вероятность в 99 %, что тест подтвердит это. Предположим, что в среднем один из десяти тысяч людей болен. Действительно ли этот тест хорош?

Нет. Если врач применяет тестирование к одному взятому наугад человеку и его тест положителен, то есть только однопроцентный шанс, что он на самом деле болен. Из-за того что здоровых людей намного больше, чем больных, тест дает ненадежные результаты, и поэтому бесполезен. (Это не столь просто понять, проще повторно протестировать человека. Но предположите, что ошибочный результат исследования повторится последовательно для этого человека.) Вывод кажется удивительным и противоречит интуиции, но он верен. Это означает, что если вы предположите, что сетевые нападения сравнительно редки, большая вероятность ошибки означает то, что ваши испытания должны быть действительно хороши, чтобы не заметить всех ложных тревог. Системы обнаружения вторжений, которые обычно сигнализируют вам в 3:00 утра о проблеме, которая в действительности не является проблемой, о ночной игре в Quake или о новом интернет-приложении, довольно быстро добьются того, что прекратят свое существование.

Есть и другие проблемы. Первая из них — это своевременное предупреждение. Я упомянул медленное нападение в предыдущем подразделе. Когда система обнаружения вторжений решит, что было произведено нападение, и уведомит вас об этом? Что, если она подумает, что это лишь отчасти выглядит как нападение? Уведомит ли она вас об этом? Когда? Опять-таки, вспомните о проблеме ложного срабатывания. Если система ошибается слишком часто, вы перестанете прислушиваться к ней.

И что вы будете делать, когда получите сигнал тревоги? Поучительные сообщения общего вида «вы подвергаетесь нападению» бесполезны, если у вас нет какого-нибудь способа защиты или даже нет времени, чтобы с этим разобраться. В 1999 году в eBay отключили электричество на 22 часа, и в течение всего этого времени система обнаружения вторжений постоянно подавала сигнал тревоги, но все были слишком заняты, чтобы ответить. В этом величайшая проблема IDS: разумно реагировать на их выводы.

Системы обнаружения вторжений еще находятся на ранней стадии своего развития, и авторы различных идей наперебой заявляют об их превосходстве. Я собираюсь только коснуться некоторых из них, в детали углубляются многие другие книги.

Для создания IDS есть два основных пути. Самый легкий — это обнаружение «неправильного» кода. IDS знают, на что похоже нападение, и ищут его. Подумайте о детекторе вирусов для сетевых пакетов. Как детекторы вирусов просматривают каждый файл, ища строки битов, указывающие на вирус, так и IDS просматри¬

вают каждый пакет, ища строки битов, которые свидетельствуют о несомненном нападении. Их легко привести в исполнение и использовать, у них низкая вероятность ложной тревоги, и они могут быть относительно быстры (принимая во внимание то, что они должны просмотреть каждый пакет).

С другой стороны, у них больше промахов. Как детекторы вирусов не в силах обнаружить вирусы, которых они никогда не видели прежде, так и подобная IDS не может обнаружить нападение, на нахождение которого она не запрограммирована. Ее легко обмануть. Иногда это можно сделать, изменив порядок следования команд в коде, предназначенном для проведения атаки. Иногда проще организовать нападение таким образом, чтобы взламывать пакеты выборочно. Так же как антивирусным компьютерным программам нужно постоянное обновление и пополнение новыми образцами кода, этому типу системы обнаружения вторжений необходимо постоянное обновление базы данных образцов нападения. Не ясно, сможет ли когда-нибудь такая база данных не отстать в соревновании с инструментом хакера.

Другим принципом работы системы обнаружения вторжений является обнаружение аномалии. IDS осуществляет некоторое статистическое моделирование вашей сети и вычисляет, что является нормой. Затем, если происходит какое-либо отклонение от нормы, она подает звуки тревоги. Здесь все может быть сделано по правилам (система знает, что нормально, и сигнализирует обо всем остальном), с использованием статистики (система статистически вычисляет, что нормально, и сигнализирует обо всем остальном) или с применением методов искусственного интеллекта.

Существует множество проблем и здесь. Что, если вас атакуют во время обучения системы? Тогда атака рассматривается как норма. Новые вещи случаются в компьютерных сетях постоянно. Знает ли система обнаружения вторжений разницу между безобидной аномалией и аномалией, указывающей на нападение? И если все, что она знает, является нормой, как же она тогда собирается катего¬ризировать нападения? Для этого вида систем вероятность ложной тревоги намного выше, и нападение на такой вид IDS включает выяснение возможности не бить в набат.

В некоторых ранних детекторах вирусов использовался этот принцип, и они забили бы тревогу, если бы вы сделали что-нибудь наподобие установки нового программного обеспечения. Они потеряли свою популярность, так же как и основанные на выявлении подозрительного кода детекторы вирусов, требующие все более совершенных словарей образцов кода; я ожидаю, что то же самое случится и с системами обнаружения атак.

Другие идеи IDS так или иначе основываются на одном из описанных выше принципов. Система обнаружения вторжения непрерывного действия (inline) может работать с сетевыми данными в реальном времени, тогда как проверяющая система использует контрольную информацию, сохранившуюся после совершения нападения. Есть IDS, базирующиеся на хостах (host-based IDS), и есть распределенные по сети IDS (network-based IDS).

Это последнее различие было темой яростных обсуждений в сообществе IDS. В своей основе IDS, распределенные по сети, построены на концепции перехвата сообщений: датчики расположены в сети, они исследуют проходящие пакеты.

У этих систем есть преимущество в скрытности — они могут быть развернуты без воздействия на остальную часть сети, и они в большей степени обеспечивают независимость от операционной системы. IDS, базирующиеся на узлах сети, исследуют систему, осуществляют контроль и регистрируют возможные нападения, помещаясь внутри отдельного компьютера. У этих систем имеется различный набор преимуществ и недостатков, составляющих их специфику.

То, что в конечном счете вы можете найти на рынке, — чаще всего гибридные системы: они являются комбинацией систем обнаружения вторжения, базирующихся на главной машине и распределенных по сети, производящих обнаружение аномалии, основанное на ожидании в совокупности с выявлением «неправильного» кода. Вы также можете найти компании, занимающиеся проверкой систем защиты, которые анализируют результаты использования этих продуктов и отвечают на подаваемые ими сигналы тревоги. Подобно брандмауэрам, системы обнаружения вторжения будут становиться все лучше и лучше, поскольку разработчики получают все больше опыта при их проектировании. И так же как у брандмауэров, их надежность в конечном итоге будет зависеть от того, насколько хорошо они конфигурированы и насколько современны их версии. И всегда будут существовать нападения, которые пройдут через них.

## Приманки и сигнализации

Сетевые сигнализации и приманки — это разновидности систем обнаружения вторжения, но они заслуживают отдельного раздела. Сигнализации — это особые системы в вашей сети, предназначенные для срабатывания в случае атаки. Приманки — это замаскированные сигнализации, которые выглядят особенно привлекательно для хакеров. Легко понять, что представляют собой сигнализации: особую сетевую команду или фиктивную сеть, про которую никто не думает, что она включает звуковой сигнал тревоги. Маркус Ранум развил эту идею дальше и предположил, что если обнаружено уязвимое место в программе, необходимо также выдавать сигнал предупреждения об опасности.

Приманки используются чаще: целые фиктивные компьютеры и фрагменты сети проектируются для привлечения нападающих. Вы можете получить от этого большое удовольствие: присвойте компьютерам такие имена, как transactions.bigcompany.com или accounting.bank.com, маскируя их под производящие впечатление счета и файлы, и используйте их для защиты вашей сети. Когда хакер проникнет в сеть, приманка будет притягивать его, поскольку она выглядит как интересное место для исследования. Затем выдается сигнал тревоги, и приманка начинает следить за активностью хакера и собирать сведения для последующего обращения в суд. Некоторые компании продают заранее сделанные приманки, просто добавляя привлекательные имена.

Интересно, что в обоих этих средствах используется одно и то же преимущество сетевого администратора над хакером: знание сети. Администратор знает, как выглядит сеть и что в ней может произойти. Он может установить сигнализацию точно так же, как домовладелец устанавливает сигнализацию на окна, которые не собирается открывать, или датчики движения в комнате, в которую не предпола¬

гает заходить. Администратор использует приманки, зная, что ни один из зарегистрированных пользователей не получит доступа к этим системам. Он может использовать любые виды сигнализации, включая и выключая их несколько раз в день, меняя их, в общем, может делать все, что он хочет. Эти средства действуют наверняка, потому что хакер не имеет информации, где и когда они могут появиться. В отличие от брандмауэров или IDS, где хакер знает, какая защита установлена, сигнализации и приманки специально разработаны для сетей, подвергающихся нападению.

## Сканеры уязвимостей

Назначение сканеров, определяющих слабые места защиты, — автоматически сканировать сеть (или компьютер) на предмет обнаружения известных недостатков. Они делают свое дело и затем выдают точный отчет о том, какие уязвимые точки имеет сеть. Обладая этой информацией, вы можете решить — усилить защиту или пользоваться сетью, несмотря на обнаруженные недостатки.

На самом деле с этими устройствами не все так ясно, и все сканеры, имеющиеся на рынке, действуют не совсем так. Если бы они работали так, как можно было бы предположить, то они испортили бы компьютеры и причинили ущерб сети. Никто не использует такие средства, поэтому приходится идти на ухищрения.

Представьте чувствительный сканер, определяющий уязвимые места в вашем доме. Он проверит, чувствительны ли ваши окна к атаке камнем. Очевидный способ сделать это — бросить камень в окно и посмотреть на результат. Но это причинит вред дому, и поэтому сканер ищет обходные пути для получения нужной информации. Он определит, одинарные или двойные стекла в окне. Возможно, постучит по ним для того, чтобы удостовериться, действительно ли это стекло или более прочный пластик. Может быть, он попытается прочитать часть номера на стекле и сделает выводы о качестве литья. Вот такие вещи приходится делать сканирующему устройству.

На самом деле все еще сложнее. Иногда трудно сказать, будет ли успешна исследуемая атака. Например, домашний сканер проверяет электробезопасность, пытаясь перерезать провод. Это ему удается, но свет не гаснет. Что это означает — что сканер на самом деле не смог перерезать линию, или что в доме имеется резервная электростанция? Или, например, сканер перерезает провод, и свет гаснет. Значит ли это, что сканер перерезал линию или сделал что-нибудь другое (несколько последовательных действий, в результате которых свет погас)? Сканер не знает об этом, и в большинстве случаев нет способа определить действительно повлиявший фактор. Сети ненадежны; чаще всего трудно понять, в чем причина неисправности.

Хотя сканирующие устройства недостаточно эффективны для обнаружения слабых мест, и они также не могут точно оценить результат своих действий, они небесполезны. Они могут исследовать, по крайней мере, окольными путями, системы на уязвимость. В результате создается список слабых мест, которые аккуратный системный администратор будет закрывать (а бесчестный хакер будет использовать). Вот тут они работают отлично.

Когда в 1995 году появился SATAN (Securuty Administrator Tool for Analyzing Networks), он произвел настоящий фурор. В средствах массовой информации он был изображен хуже, чем его тезка (сатана), и автор этой программы был уволен с работы. С тех пор отношение к сканирующим устройствам изменилось, и они стали использоваться как часть набора инструментов администратора безопасности. На рынке сейчас присутствуют несколько коммерческих продуктов подобного рода с известными именами. Их можно представлять себе как некую разновидность аудита: это похоже на то, что некоторое частное лицо исследует вашу сеть, и сообщает 0 слабых местах вашей системы безопасности. Вы можете нанять исследователя для проверки вашей системы, но хакер может нанять того же самого исследователя для проверки возможности атаки. Понятно, что это ограничения технологии.

## Безопасность электронной почты

Сейчас электронная почта широко распространена. Любой, кто присутствует в киберпространстве, имеет электронный адрес и, вероятно, получает много сообщений каждый день. Почтовые программы не имеют встроенной системы безопасности.

Любой узел сети по пути следования сообщений между отправителем и получателем способен прочитать электронную почту, так же как и любой другой сетевой пакет. (Вы можете даже увидеть имена некоторых из этих машин в заголовке полученной почты.) Интернет-сообщение можно сравнить с почтовой открыткой: любой — почтальон, сортировщик почты, любопытные перевозчики, — в общем, те, кто соприкасаются с почтовой открыткой, могут прочитать сообщение на обратной стороне. Также нет способа проверки подписи или обратного адреса (знаете ли вы, что написанное в заголовке письма имя отправителя можно легко сфальсифицировать?), поэтому мы не можем знать наверняка, откуда пришло это письмо. (Распространители спама[1] используют это для сокрытия истинных адресов массовых рассылок.) Если хакер хочет все красиво обставить, он может связаться с машиной, которая должна явиться отправителем его сообщения, и действительно послать сообщение с нее. Если ему все равно, то он просто подделывает имя в заголовке письма.

Хотелось бы, чтобы электронная почта обеспечивала две вещи. Во-первых, мы должны быть уверены, что никто не сможет прочитать сообщение, кроме того, кому оно действительно предназначается. Во-вторых, мы должны твердо знать, что сообщение на самом деле пришло от того человека, имя которого указано в заголовке, и что никто не мог его подделать.

С помощью криптографии легко защитить электронную почту, и на рынке имеются десятки продуктов, призванных обеспечить решение этой проблемы. Вот основная последовательность действий.

1. Алиса получает открытый ключ Боба.
2. Алиса подписывает сообщение своим закрытым ключом.

[1] Спам (англ. SPAM) — навязываемая широкому кругу адресатов информация, в основном рекламного характера. — *Примеч. перев.*

3. Алиса шифрует сообщение с помощью открытого ключа Боба.
4. Алиса отсылает Бобу зашифрованное и подписанное сообщение.
5. Боб расшифровывает сообщение при помощи своего закрытого ключа.
6. Боб проверяет подпись Алисы, используя открытый ключ Алисы.

У вас, скорее всего, возникают вопросы относительно открытых ключей: как их получить, где хранить, как проверять. Я расскажу об этом подробно в главе 15.

## Шифрование и сетевая защита

Защиту от сетевых **атак нельзя** свести просто к применению криптографии в системах. Часто особенности системы не позволяют использовать криптографию. Например, одна часть записи системы доменных имен постоянно изменяется, поэтому непрактично использовать цифровые подписи в этой системе. Подтверждение подлинности с помощью криптографии в данном случае просто не будет работать.

Или представьте себе виртуальный мир, в котором каждый пакет зашифрован с помощью IPsec. Как только пакеты будут зашифрованы, их нельзя будет анализировать. Сетевые инженеры не смогут больше делать анализ трафика. Системы перевода адреса не смогут работать с пакетами. Системы, которые оптимизируют размер пакета для передачи через спутник, тоже не будут работать.

Другой пример: множество сетевых защит рассчитаны на проверку пакетов. Шифрование может препятствовать такой защите.

Рассмотрим антивирусное программное обеспечение, используемое в брандмауэрах, которое автоматически сканирует все входящие электронные сообщения. В больших корпорациях эти программы, просматривая почту, могут находить более 1000 вирусов в день. Если эти корпорации будут шифровать все сообщения, то подобные программы никакой опасности не обнаружат (если они не имеют ключа).

Рассмотрим брандмауэр, который просматривает входящие пакеты на предмет выявления возможного нападения. Если в этой сети везде используется IPsec, то брандмауэры ничего не смогут проверить.

Нет хорошего решения этой проблемы. Один из возможных путей — это снабдить брандмауэр ключом, с помощью которого можно осуществить дешифрацию сообщений. Он несет в себе множество потенциальных проблем безопасности. Другой вариант — это *распределенный брандмауэр:* распределить защиту по всему сетевому пространству через каждый узел сети. И это решение имеет свой комплекс проблем, но, вероятно, за ним будущее брандмауэров.

Исследователи Интернета бьются над этой проблемой; у меня тоже нет готового ответа.

# Глава 13. Надежность программного обеспечения

Системные меры безопасности (ядра безопасности, меры контроля доступа, криптография и т. д.) в комплексе с хорошими сетевыми мерами безопасности (брандмауэрами, системами обнаружения вторжения, механизмами проверки) создают впечатление достаточной компьютерной безопасности. Почему же тогда и компьютеры, и сети так ненадежны? Почему мы так часто становимся свидетелями уязвимости компьютеров и почему не происходит изменение в лучшую сторону?

Проблема в том, что такие меры безопасности, как шифрование, ядра безопасности, брандмауэры и прочие, лучше работают в теории, нежели на практике. Другими словами, изъяны системы безопасности значительно чаще случаются при вводе ее в действие, и они намного более серьезны, чем те, что возникают при ее разработке. До сих пор во второй части данной книги говорилось о разработке. В этой главе речь пойдет о вводе в действие.

## Дефектный код

В июле 1996 года вследствие ошибки в программе вскоре после запуска взорвалась ракета «Ариан 5» Европейского космического агентства: программа пыталась поместить 64-разрядное число в 16-разрядное пространство, вызвав переполнение. Этот урок особенно важен для понимания проблем компьютерной безопасности.

По существу, проблема была связана с фрагментом кода, обрабатывавшего данные о скорости бокового ветра, написанного еще для ракеты «Ариан 4». Через 36,7 секунды после запуска управляющий компьютер попытался преобразовать значение скорости из 64-разрядного формата в 16-разрядный. Число оказалось слишком большим, что и вызвало ошибку. Обычно используется дополнительный код, который отслеживает ошибки такого рода и исправляет их. Но в данном случае программисты-разработчики решили не беспокоиться о подобном коде, так как величина скорости никогда не достигала таких больших значений, чтобы создавать проблемы. Возможно, это было верно для «Ариан 4», но «Ариан 5» — более быстрая ракета. Но хуже всего то, что эти вычисления, содержащие ошибку, не имели смысла с того момента, когда ракета оказывалась в воздухе. Программа, их производящая, была нужна лишь для того, чтобы отладить систему перед запуском, и после этого ее надо было бы сразу отключить. Но инженеры еще при разработке более ранней модели ракеты решили использовать эту функцию в течение первых 40 секунд полета, чтобы облегчить перезапуск системы в случае задержки

запуска в последний момент перед стартом. Была резервная система, предназначенная дублировать основную в случае ее отказа, но она работала с тем же самым программным обеспечением, содержавшим те же самые ошибки.

В результате всех этих событий работа системы управления, полностью запутавшей бортовой компьютер «Ариан 5», была прекращена. Это привело к ненужной корректировке курса ракеты и повлекло ее самоуничтожение.

Три года спустя во время сложных маневров исчез искусственный спутник планеты Марс, запущенный NASA. Это не было делом рук марсианской противовоздушной обороны, а произошло вследствие ошибки преобразования данных. Инженеры NASA неудачно перевели значение силы сопротивления из английской системы мер в метрическую. Значения различаются в 4,45 раза: этого оказалось достаточно, чтобы научно-исследовательская станция опустилась на 50 миль ниже и сгорела в марсианской атмосфере.

Эти две катастрофы не связаны с компьютерной безопасностью, но они могут служить для пояснения того, насколько сложно разработать и ввести в действие код без ошибок. И Европейское космическое агентство, и NASA располагают достаточно большими средствами и сильно заинтересованы в том, чтобы обеспечить качество программного обеспечения. Но они до сих пор не в состоянии сделать это.

У других дела обстоят не лучше. В 1999 году eBay потеряла 22 часа из-за связанных с программным обеспечением ошибок в коде, полученном от Sun Microsystems. Выявление ошибки задержало выпуск карманных компьютеров Visor. А в 1998 году дефект в коммутаторах, произведенных компанией Cisco Systems, привел в нерабочее состояние передающую сеть компании AT&T Interspan, что отразилось на работе 6600 клиентов.

Печальная действительность состоит в том, что подобные ошибки программного обеспечения возникают везде. Большинство из них не приводит к таким разрушительным последствиям (перезагрузка электронной таблицы после аварийного отказа вызывает всего лишь незначительное раздражение), но так как сложное программное обеспечение во многих случаях функционирует внутри жизненно важных систем (например, в системах уклонения от автокатастрофы, взлета и посадки самолетов, управления атомной электростанцией), мы, вероятно, станем свидетелями увеличения количества подобных случаев. Проводится большая работа по исправлению ошибок, устранению недостатков, это получило название отказоустойчивой стратегии: например, если в автомобиле откажет система уклонения от автокатастрофы, предполагается, что водитель будет вести себя, как в машине без компьютера, вместо того чтобы позволить ей врезаться в ближайшее дерево. Идея в том, чтобы была уверенность, что небольшие недостатки не приведут к потере контроля над ситуацией, как было в случае с «Ариан 5».

Трудно обнаружить ошибки в программном обеспечении, влияющие на правильное выполнение задачи; обнаружить ошибки в системе безопасности еще труднее.

Надежность означает, что компьютер, в первую очередь программное обеспечение, но также и любые специализированные технические средства должны работать даже при появлении случайных ошибок. Они могут возникать при проектировании (использование одинакового программного обеспечения в основной

и резервной системах), при вводе в действие (отсутствие проверки наличия ошибок при преобразовании данных), это могут быть ошибки программирования (вспомните математическую ошибку в чипе Intel Pentium[1]) или ошибки пользователя. Время от времени такие ошибки появляются. Это похоже на компьютер Мерфи: сбои происходят... редко, но постоянно. Если компьютер ошибается время от времени, пусть даже редко, это заметно любому пользователю.

Основная проблема состоит в том, что в любой сложной системе, программном обеспечении, применяемом в ракетной технике, большой базе данных, операционной системе, сетевом программном обеспечении, сложном микропроцессоре очень многие вещи могут работать со сбоями. И это определяет предел сложности. Невозможно предусмотреть или проверить абсолютно все. Неизбежно где-нибудь произойдет сбой.

Компьютерная защита более всего похожа на программирование для компьютера Сатаны. (Росс Андерсон ответствен за этот красивый оборот.) Чтобы быть безопасным, программное обеспечение должно работать, несмотря на появление неуловимых и опасных ошибок, которые могут быть преднамеренно внедрены способным нападающим с целью нанести поражение системе. Надежное программное обеспечение должно пережить и случайные ошибки, которыми может воспользоваться сообразительный хакер. (Представьте себе, что это некий хакер вызвал ошибку переполнения в программном обеспечении «Ариан 5» в самое неподходящее время.) Ошибки происходят случайно, и большинство из них редко встречается при обычном использовании. Но нападающие разыщут потенциальные ошибки и непременно воспользуются ими для достижения своих целей.

Широко применяемая стратегия для обнаружения случайных ошибок — предварительное тестирование: предоставить программное обеспечение большой группе пользователей (бета-тестирование). Люди будут пользоваться программами во всевозможных конфигурациях, на различных типах персональных компьютеров и с различными целями (о некоторых из них проектировщики даже не думали). Если они не смогут сломать систему, возможно, в ней нет ошибок. Сложно проводить предварительное тестирование программного обеспечения ракетной техники, но любое крупное коммерческое программное приложение, которое покупает пользователь, прошло тысячи часов предварительного тестирования для нахождения и исправления ошибок программирования.

---

[1] Известна как «floating point flaw». Ошибка выражалась в потере точности (от 4-го до 19-го разряда после десятичной точки) при выполнении деления (инструкция FDiv) с некоторыми сочетаниями операндов. Была обнаружена в 1994 году для процессоров Pentium 60-100 МГц. Intel долгое время не признавала наличие ошибки, но, в конце концов, была вынуждена потратить 475 миллионов долларов на замену чипов. 1997 год выявил две ошибки. Первая, для Pentium II (ошибка флагов: «Dan-0411», «Flag Erratum»), проявлялась при преобразовании форматов больших отрицательных чисел и приводила к невозможности оповещения программ о завершении некоторых операций. Метод модификации микрокода во время загрузки уже работал, и ошибка была исправлена без отзыва проданных процессоров. Осенью выяснилось, что любой процесс, выполняемый на Pentium и Pentium MMX, способен остановить «сердце» PC, выдав код F0 0F C7 C8 (сравнивающий 32-битовый — операнд с 64-битовым, чем не случай с «Ариан»?). Благодаря некоторым конструктивным условиям практически всегда такой блокировки не происходит, но теоретически «Pentium FO bug» возможен. Теперь Intel размещает на своем сайте полную техническую информацию обо всех найденных «опечатках» (erratums). Так, для Pentium III в июне 2002 года количество таких ошибок равнялось 46, но все они имеют гораздо меньший уровень трагизма. — *Примеч. ред.*

Возможно, только что приведенные рассуждения позволили вам расслабиться. Тем не менее, зная, что большинство коммерческих программ содержат большое количество ошибок, трудно доверять подобным испытаниям. Испытания происходят, но сложности остаются. Основную роль играет необходимость быстрого продвижения программных продуктов на рынке. Некоторые компании в связи с этим выпускают в широкую продажу плохо проверенные программы. (Большая часть программного обеспечения Интернета выпущена в предварительных версиях; некоторые даже доказывают, что сам Интернет все еще находится в предварительной версии.) Кроме того, такой натиск на рынок означает, что некоторые компании выпускают программное обеспечение в продажу раньше, чем будут исправлены все ошибки, которые уже установлены. (И если ошибки, найденные в бета-версии, были исправлены, часто не производится повторный цикл предварительных тестов для проверки исправленного кода.)

## Нападения на дефектный код

Большинство проблем, связанных с компьютерной безопасностью, которые нам приходится наблюдать, являются результатом дефектов в программном коде. Вот некоторые примеры:

- В 1988 году червь Морриса использовал ошибку в UNIX для получения полного доступа к компьютерам, выполняющим программы. Это привело к переполнению буфера, о чем будет рассказано в следующем подразделе.
- В 1999 году некто обнаружил ошибку в сценарии Hotmail CGI, позволяющую пользователю получить доступ к записям электронной почты другого пользователя. Дефекты такого рода обсуждались в главе 10.

Традиционно дефектный код был орудием, используемым для взлома компьютеров. Например, недостатки программ, отсылающих почту, повлекли за собой огромное количество незаконных проникновений в компьютеры с операционной системой UNIX. Цель подобных нападений состоит в использовании погрешностей таким образом, чтобы нападающий мог взять в свои руки контроль над системой. Нападения незаметны, они могут использовать настройку параметров для получения доступа или лазейки в заголовке сообщения об ошибках для прочтения защищенных файлов, количество таких нападений огромно. Временами кажется, что каждый день происходит новое нападение на почтовые программы, после которого в очередной раз исправляются не найденные до того ошибки. (Производятся ли после этого исправления у пользователей коммерческих программ — это другой вопрос.)

Более недавний пример — это модель безопасности Java. В Java используется модель комплексной безопасности для защиты компьютеров от вредоносных апп¬летов Java. Ошибка в любом месте программного кода, ответственного за работу защитных механизмов, может сделать все эти механизмы бесполезными, и, если такая ошибка случится, характерные для Java нападения хлынут широким потоком, используя любые недостатки системы.

Эти примеры вызывают больше беспокойства, чем проблема «Ариан» (несмотря на меньшую степень накала страстей), поскольку недостатки, которые могут быть использованы для взлома защиты, обычно не влияют на выполнение программ. Они незаметно присутствуют там до тех пор, пока кто-либо не воспользуется ими. Это очень важно и потому создание защиты сложнее, чем обеспечение надежности. Ошибка, повлекшая за собой катастрофу «Ариан», — это единственный случай, который затронул выполнение. Как только ошибка при выполнении найдена — и предварительное тестирование сможет обнаружить ее — она может быть исправлена. Дефекты защиты не влияют на выполнение и не проявляются в результатах предварительного тестирования. Подробнее о надежности тестирования будет рассказано в главе 22, но мораль в том, что люди постоянно спотыкаются о недостатки в системах безопасности, и только опытные эксперты на самом деле способны отыскать их.

Такое случается постоянно. Когда квалифицированный специалист производит анализ защиты программного обеспечения, он всегда обнаружит случайные недостатки, подрывающие систему безопасности. Всегда. Чем сложнее код, тем больше несовершенства в его защите.

Огрехи защиты, однажды обнаруженные, будут использоваться до тех пор, пока не будут устранены. Предположим, что нападающий нашел брешь в защите торгового протокола, что позволило ему украсть номер кредитной карты или, что еще хуже деньги. Если его действия мотивировались желанием создать саморекламу, он известит о своем достижении прессу и эта ошибка будет исправлена. (Хотелось бы надеяться, что сначала он предупредит компанию.) Если его действиями управляет желание получить деньги, нападающий станет использовать эту возможность снова и снова. Он украдет столько, сколько сможет, пока еще кто-нибудь не обнаружит этот недостаток и не исправит его. В этом основное отличие: недостатки, влияющие на выполнение, заметны, в то время как недочеты защиты могут оставаться невидимыми в течение долгого времени.

Эти недостатки не обязательно находятся в коде, относящемся к системе безопасности. Они могут присутствовать повсюду: в интерфейсе пользователя, в программах обработки ошибок, в любом другом месте. И как мы видели в главе 10, даже программы, не имеющие никакого отношения к компьютерной безопасности, могут повлиять на защищенность компьютеров, работающих в сети. Недостатки в текстовом процессоре, драйвере принтера или мультимедийном проигрывателе могут полностью подорвать систему безопасности вашего компьютера.

Еще один вывод состоит в том, что ошибки в программном обеспечении (и, следовательно, недостатки защиты) неминуемы. Предположение, что огромное пространство Интернета может быть свободно от ошибок, настолько же невероятно, как, то, что программное обеспечение «Ариан 5» было полностью защищено от сбоев и лишь несчастливое стечение обстоятельств привело к таким катастрофическим последствиям.

Мы наблюдали подобные вещи в Windows NT. Не проходит и дня без объявления об обнаружении нового просчета в системе безопасности этой программы. Те же тенденции наблюдаются и в Windows 2000.

# Переполнения буфера

Переполнения буфера (иногда называемые разрушением стека) являются обычным способом разрушения защиты. Их легко осуществить; атаки достигают своей цели чаще всего именно благодаря буферным переполнениям. Нападения такого рода могут быть разрушительными, часто они заканчиваются получением полного контроля над компьютером. Этот метод использовался во многих выдающихся нападениях. Поскольку уменьшение количества таких атак не наблюдается, стоит детально объяснить, что они собой представляют и как работают.

Давайте начнем с аналогии. Если вы попытаетесь украсть что-нибудь из близлежащего магазина, то вам придется пробираться мимо продавца. Продавец не станет творчески подходить к делу. Скорее всего, он предпримет только те действия, которые предписаны инструкцией. Инструкция служащего — это большой набор протоколов, описывающих различные ситуации. Например: «контакт с лицом, утверждающим, что он служащий».

- Шаг 1. Попросить показать удостоверение.
- Шаг 2. Убедиться, что удостоверение не поддельное.
- Шаг 3. Проверить, что на фотокарточке в удостоверении действительно изображен этот человек.
- Шаг 4. Если это так, впустить его. Если нет, не впускать.

Или: «контакт с водителем, привозящим товар»:

- Шаг 1. Взять коробку.
- Шаг 2. Расписаться за коробку.
- Шаг 3. Убедиться, что водитель уезжает.

Водитель не может пройти мимо служащего назад в магазин, потому что в инструкции ясно сказано, что после получения подписанной квитанции водитель должен уехать.

Компьютеры работают почти так же. Программы подобны шагам в инструкции; компьютеры выполняют то, что написано в программах, и ничего больше. Сетевые компьютеры работают аналогично. У них есть набор протоколов, которым они следуют, — эти протоколы описывают процедуру входа в систему, ограничения доступа, защиту паролей, и определяют, кто может быть допущен, а кто нет. Тот, кто действует в соответствии с протоколами, будет пропущен, а кто действует иначе — войти не может.

Один из способов нанести поражение такому протоколу состоит в изменении действующей компьютерной программы. Это подобно замене страницы в инструкции для служащего. Инструкции обычно пишутся для того, чтобы их исполнители не вдавались в размышления. Каждая страница — это шаг: «Если клиент дает вам кредитную карту, смотрите следующую страницу. Если клиент расплачивается наличными деньгами, смотрите страницу 264». Шаги, описывающие контакт с развозчиком товара, могут выглядеть следующим образом.

- Страница 163. Возьмите коробку. Если она одна, смотрите следующую страницу. Если коробок несколько, смотрите страницу 177.

- Страница 164. Возьмите форму для подписи, подпишите и верните ее. Смотрите следующую страницу.
- Страница 165. Спросите водителя, хочет ли он что-нибудь купить. Если он хочет, смотрите страницу 13, если нет, смотрите следующую страницу.
- Страница 166. Попросите водителя уехать. Если он.... и т. д.

Всякий раз, когда служащий магазина совершает какое-либо действие, он руководствуется открытой страницей в своей инструкции. Он не может посмотреть на вещи иначе.

Нападение состоит в следующем: притворившись развозчиком, можно поменять страницу в инструкции служащего, когда он будет занят подписью квитанций. Все, что нужно сделать, это дать ему два листа вместо одного. Верхний лист — это квитанция, а нижний — поддельная страница инструкции:

- Страница 165: Отдайте водителю все деньги из кассового аппарата. Смотрите следующую страницу.

Это сработает. Служащий возьмет коробку, как написано на странице 163. Он посмотрит страницу 164 и возьмет квитанцию (вместе с фальшивой страницей). Он положит оба листа на открытую инструкцию, подпишет и вернет квитанцию (оставив фальшивую страницу в инструкции), затем, вернувшись к инструкции, увидит поддельную страницу. Он отдаст все деньги из кассы и увидит следующую страницу (настоящую страницу 165). Шоферу нужно ответить, что он не хочет ничего купить, и уехать. Если служащий магазина на самом деле такой же тупой, как компьютерная система, развозчик сможет уехать с деньгами. Можно использовать этот способ обмана, чтобы убедить служащего магазина пустить нас на склад или чтобы исполнить любой другой замысел. Подложив страницу в инструкцию, можно произвольно поменять его действия.

По сути дела, это способ использовать ошибки переполнения буфера в компьютерных системах. Компьютеры хранят в памяти все программы и данные. Если компьютер запрашивает у пользователя пароль, который должен состоять из 8 символов, и получает пароль из 200 символов, то дополнительные символы могут записаться в какую-то другую область памяти. (Компьютер не может предположить, что происходит что-то неправильное.) Если это подходящая область памяти, и в нее записать нужные символы, то можно изменить команду «запретить подключение» на команду «разрешить доступ» или даже выполнить ваш собственный код.

Червь Морриса является, вероятно, наиболее известным примером использования ошибки переполнения. Он использует переполнение буфера в программе для UNIX, которая должна идентифицировать пользователя по вводимым им данным. К сожалению, в ней не существовало ограничения на размер вводимой информации. Ввод более чем 512 байт приводил к переполнению буфера, и специальный длинный код Морриса позволял установить его мошенническую программу на компьютер, подвергшийся нападению, и выполнить ее. (Эта ошибка, конечно, была исправлена.)

Описание этого червя особенно уместно в данном разделе, поскольку он сам содержит программную ошибку. Он должен был перепрыгивать с компьютера на компьютер в Интернете, копировать сам себя на каждый сервер и затем следовать дальше. Но опечатка в коде привела к тому, что вирус копировался неограничен¬

ное число раз на каждом компьютере. Результатом была поломка пораженных компьютеров. Крушение произошло на 6000 серверах Интернета, в то время это составляло 10% от их общего числа.

Умелое программирование способно предотвратить этот род нападений. Программа может сокращать пароль до 8 символов, так что лишние 192 символа никогда не запишутся в память. Сделать это легко, но применить везде — сложно. Проблема состоит в том, что в любой части современного большого и сложного кода есть достаточно мест, где возможно буферное переполнение (которые не столь просто обнаружить, как в этом примере). Очень трудно гарантировать, что нет никаких проблем с переполнением, даже если у вас было время для проверки. Чем больше и сложнее код, тем больше вероятность нападения.

В Windows 2000 содержится 35-60 миллионов строк кода, и никто, кроме самих разработчиков, не видел их.

## Вездесущность ошибочного кода

Этот подраздел в основном посвящен Интернету и недостаткам в его безопасности. Это не означает, что в Интернете большее количество недоработок защиты, чем в других сетях. Недостатки Интернета чаще привлекают внимание просто потому, что большинство людей имеют дело с его программным обеспечением и находят там просчеты. Программное обеспечение в других областях киберпространства — в телефонной сети, в банковской электронной сети — точно так же содержит множество ошибок.

По данным Университета Карнеги-Меллона, на 1000 кодовых строк обычно приходится от 5 до 15 ошибок. Большинство этих ошибок не влияет на выполнение программ и никогда не обнаруживается. Но любую из них можно использовать для взлома защиты.

Создается впечатление, что код Интернета быстро улучшается. Недостатки защиты находят постоянно. Некоторые компьютерные журналы еженедельно публикуют сведения об ошибках в почтовых программах, обнаруженных за прошедший период. Создатели этих программ обычно довольно быстро исправляют ошибки, как только те становятся достоянием общественности; но до тех пор большинство из них не будут беспокоиться.

Это, конечно, предполагает, что вы всегда ставите самые последние «заплаты». Обычно вслед за сообщением об уязвимости выпускается заплата. Если вы верите новостям, то история на этом завершится. Но в большинстве случаев заплаты никогда не устанавливаются. Главная проблема Интернета состоит в том, что внесенные исправления не обязательно доходят до пользователей программного обеспечения. «Эпоха Интернета» повлияла и на работу системных администраторов.

Даже при том, что заплаты доступны, прореха остается. По существующим оценкам, более чем 99 % всех нападений в Интернете могли быть предотвращены, если бы системные администраторы использовали самые свежие версии системного программного обеспечения. Поэтому сканирующие устройства, определяющие уязвимые места, являются такими подходящими инструментами и для хороших, и для плохих парней.

Даже если предположить, что каждый пользователь всегда работает с последним обновлением какой-либо программы, положение не становится лучше. При выпуске каждой новой версии в ней появляются новые ошибки. Если в версии 1.0 были найдены и исправлены десятки или сотни ошибок защиты, то это ничего не говорит о надежности версии 2.0. Вероятно, версия 2.0 больше и имеет большее количество особенностей; в ней имеются все типы нового кода. Исправления, внесенные в первую версию программы, нельзя перенести во вторую, и, кроме того, там, вероятно, ошибок стало еще больше.

# Глава 14. Аппаратные средства безопасности

Это древняя идея. Она начала воплощаться, когда первый человек нарисовал линию, обозначающую вход в его пещеру, объявив тем самым, что с одной стороны этой границы находится его территория, и затем охранял свою пещеру от всех, кто был с этим не согласен. К аппаратным средствам безопасности можно отнести массу различных вещей: компьютерные помещения за запертыми дверями и охраняемые заграждения, помехоустойчивые устройства для платного телевидения, безопасные опознавательные знаки для контроля доступа, смарт-карты для приложений электронной коммерции и мины, которые взрываются, если вы пытаетесь обезвредить их. Реализация идеи создания безопасной территории различна в каждом из этих случаев, но основная мысль такова: намного легче построить систему компьютерной безопасности, если некоторые части системы останутся недоступными для большинства людей. Для этого можно использовать изначально заложенные в устройства физические средства защиты.

И это правда. Легче сконструировать безопасную систему платы за парковку, если вы способны предположить, что мошенники не смогут использовать парковочные счетчики, чтобы набить себе карманы. Легче представить себе охрану библиотеки и предположить, что люди не смогут вынести из здания книги, спрятав их под пальто. И легче создать электронный бумажник, если у вас есть основания считать, что люди не смогут присвоить любое количество денег.

Рассмотрим совершенную безналичную денежную систему: каждый имеет при себе бумагу, на которой написано число, представляющее денежную сумму, которой он располагает. Когда кто-нибудь тратит деньги, он зачеркивает старое и пишет меньшее число. Когда он получает деньги, то поступает наоборот. Если каждый честен, система работает. Как только некто заметит, что он может написать на бумаге любое число, которое захочет, системе придет конец.

Однако это в точности та же самая система, которую до эпохи компьютеризации банки использовали для депозитных счетов. Когда кто-то кладет деньги на депозит, это отражается в банковской книге, хранящейся в специальном помещении банка, другая книга находится в распоряжении клиента[1]. В банковской книге записано число, соответствующее количеству денег, которые данное лицо хранит в банке. Если оно вносит или снимает деньги со счета, банк записывает новую сумму в обеих книгах. Эта система работает потому, что одна из книг находится в охраняемом помещении банка. Она и есть «настоящая» книга; книга, которая у вкладчи¬

[1] Это напоминает существующие у нас «сберегательные книжки». — *Примеч. перев.*

ка, — это только ее копия, выдаваемая для его спокойствия. Если вкладчик подделает запись в своей книге, это приведет к несоответствию с записью в книге, хранящейся в банке. Кассир в банке обнаружит эту неувязку, возможно, проверит другие записи, чтобы удостовериться, что действительно была предпринята попытка мошенничества, и поступит соответственно обстоятельствам. Клиент не может изменить записи в книге, хранящейся в банке, поскольку не вправе проникнуть на охраняемую территорию. (Кассир, конечно, имеет гораздо больше возможностей совершить мошенничество.)

Этот пример показывает, насколько важно создать безопасную территорию: система безопасности не будет работать без нее.

Мы можем построить систему анонимных карточек для оплаты таким же образом. Клиенты носят смарт-карты в своих бумажниках. Смарт-карта содержит в памяти информацию о количестве долларов на счете, точно так же, как банковская книга. Смарт-карты могут взаимодействовать друг с другом через некий терминал пункта продажи. Когда клиент что-нибудь покупает, его смарт-карта вычитает сумму потраченных денег из количества, которое хранилось в памяти, и записывает в память новую, меньшую сумму. Когда торговец продает что-нибудь, его смарт-карта прибавляет стоимость товара к числу, хранящемуся в памяти. Такая операция может осуществляться только одновременно с обеими картами (это легко контролировать с помощью секретных ключей смарт-карт), так что всегда сохраняется баланс. И чтобы воспрепятствовать кому бы то ни было проникнуть внутрь смарт-карты и изменить баланс, карты должны быть защищены от вторжения.

Легко ли это? Безопасная территория находится внутри карты — там хранятся ее секреты, и посторонние не могут их узнать — это сразу отметает множество проблем. Без этого единственный способ заставить подобную систему работать — это использовать нудные процедуры обращения к базам данных.

Чеки работают так же, как в первом примере, о котором я говорил: представим себе, что некто хранит в своем бумажнике документ, в котором представлен баланс на его текущем счете. Он может выписать чеки на любую произвольную сумму: ничто не вынуждает его выписывать чек на сумму меньшую, чем находится на счете. Торговцы часто принимают такие чеки на веру; они не могут знать, действительно ли данная персона имеет на счете сумму, покрывающую этот чек. Но так как здесь нет безопасной территории, которая бы вынуждала людей быть честными, существует сложная межбанковская система проверки чеков. Продавец вносит чек на депозит, но не может получить деньги сразу. Банк продавца использует идентификационную информацию на чеке — номер счета, имя банка и т. д., чтобы выяснить, с какого счета должны быть переведены деньги. Затем он обращается в банк покупателя и требует произвести платеж. Банк покупателя проверяет его личный счет. Он снимает деньги со счета клиента и переводит их в банк продавца. Наконец, деньги поступают на его счет.

Конечно, реальная система проверки чеков работает не совсем так — она оптимизирована для большей скорости и эффективности, — но основная ее идея именно такова. Нельзя рассчитывать, что владельцы счетов не станут выписывать негодные чеки, так что банки вынуждают людей быть честными.

# Сопротивление вторжению

Системы защиты от вторжения должны были бы помочь решить множество проблем компьютерной безопасности. Подумайте, насколько легче обеспечить защиту от копирования на вашем компьютере, если на нем находится процессор, снабженный такой системой, понимающей только зашифрованные команды. Или насколько легче было бы сконструировать систему условного депонирования ключей (см. главу 16), если бы аппаратные средства систем защиты от вторжения сигнализировали полиции о необходимости прослушивания. С аппаратными средствами систем защиты можно было бы реализовать в Интернете «счетчик», который фиксировал бы доступ к данным, подобно тому как электрический счетчик фиксирует количество использованной энергии.

Вообще говоря, аппаратные средства систем защиты от вторжения идеальны для сложных отношений доверия, когда одна сторона хочет передать в руки другой некий механизм безопасности и при этом иметь уверенность в том, что другая сторона не сможет изменить этот механизм. Например, когда банк хочет контролировать баланс счетов на устройстве, находящемся в руках его клиентов. Или когда полиция хочет хранить копни ключей шифрования, с тем чтобы иметь возможность прослушивать частные разговоры, даже когда люди используют механизмы шифрования. Или для дешифратора кабельного телевидения.

Основная проблема заключается в том, что таких средств защиты от вторжения не существует. Вы не можете создать механизм, в который невозможно проникнуть. Возможно даже, что вы и сделаете устройство, в которое невозможно проникнуть на данном уровне технологии. Но вы не можете создать устройство абсолютно защищенное.

Я мог бы посвятить целую книгу деталям, но они меняются так часто, что это было бы бесполезно. Достаточно сказать, что есть несколько закрытых лабораторий в Соединенных Штатах, в которых могут разрушить любую технологию систем защиты от вторжения. Гораздо больше лабораторий в различных корпорациях могут работать над разрушением сопротивления вторжению, хотя бы они и были созданы для других целей. Лаборатории разведывательного управления, разрабатывающие микросхемы, например, имеют оборудование, которое может быть использовано в основном для изучения микросхем систем защиты от вторжения, имеющихся на рынке.

В соответствии с этими реалиями, многие компании определяют название своих технологий как *сопротивление вторжению,* которое есть нечто подобное «защите от почти любого вторжения». Я думаю, это разумно: письмо, запечатанное в конверте, может рассматриваться как один из примеров сопротивления вторжению, хотя ЦРУ и другие подобные учреждения имеют потрясающий опыт вскрытия почты.

Проблемы с системами защиты от вторжения выявляются, когда происходит сопротивление реальному вторжению. Представьте себе, что вы создаете систему торговли, основанную на смарт-картах, которая использует в целях безопасности микросхему, обеспечивающую сопротивление вторжению. И это анонимная система, так что сопротивление вторжению — это вся защита, которую вы можете противопоставить широко распространенным подделкам. Насколько сильное сопро¬

тивление вторжению вам необходимо? Как вы определите, что защита вполне достаточна? Что вы будете делать, когда появятся новые технологии?

Определяя, насколько серьезное сопротивление вторжению вам необходимо, надо выдвигать выполнимые требования. Вероятно, вы сможете оценить величину возможного ущерба: сколько денег сумеет подделать тот, кто взломает систему сопротивления. Если вы сконструировали хорошую систему, ожидаемо, что вы можете ограничить количество денег, которые могут быть украдены с одной смарт-карты, скажем, сотней долларов. Следующая задача более трудная: как вы узнаете, что обеспечили такие меры по сопротивлению вторжению, что взлом системы обойдется дороже 100 долларов?

На самом деле никто не знает, насколько эффективны различные меры по сопротивлению вторжению. Конечно, в лаборатории скажут вам, сколько времени они потратили, взламывая систему, или сколько стоит оборудование, которое они при этом использовали, но некто в лаборатории на другом конце города способен воспользоваться другой техникой и получить совершенно другую картину. И помните об атаках ради огласки: некий студент последнего курса может позаимствовать оборудование и взломать вашу систему сопротивления вторжению просто шутки ради. А может быть, представители криминальных кругов купят необходимое оборудование и наймут квалифицированных взломщиков. Все это не имеет прямого отношения к простой оценке времени и денег, необходимых для осуществления атаки «в лоб» против имеющегося алгоритма шифрования.

И даже если возможно выяснить, насколько эффективна сегодня техника, отвечающая за сопротивление вторжению, вы ничего не можете сказать о том, насколько эффективна она будет завтра, или на следующий год, или через пять лет. Новые достижения в этой сфере появляются все время. Прогресс в этой области происходит от разнообразия технологий; они могут взаимодействовать неожиданным образом. То, что было трудно взломать в этом году, возможно, будет очень просто взломать в следующем. Наивно полагать, что сопротивление вторжению обеспечивает сколько-нибудь долговременную безопасность.

Другая возможность — создать систему *обнаружения вторжения*. Это легче, чем реализовать сопротивление вторжению: мы не заботимся о том, что некто может проникнуть в систему, мы заботимся только о том, чтобы он не остался при этом непойманным. Представьте себе игровое устройство с ручным управлением, снабженное такой системой. Игрок может взять его домой и выиграть или проиграть деньги. Поскольку мы собираемся позволить игроку взять устройство к себе домой, и мы знаем, что потенциально ему по силам выиграть тысячи долларов, самое лучшее, что мы можем сделать, — организовать сопротивление вторжению. Но поскольку мы знаем, что надежную систему сопротивления вторжению создать невозможно, в действительности мы полагаемся на обнаружение вторжения. Когда он возвратит игровое устройство и захочет забрать свой выигрыш, мы проверим устройство со всех сторон. Мы будем выяснять, не нарушена ли изоляция, не вскрывалась ли оболочка, не обрезаны ли провода. Конечно, самый лучший атакующий сможет все это замаскировать, но он не сможет сделать вид, что так оно и было.

Неплохо, но все же недостаточно хорошо. Я верю, что не существует абсолютно надежной системы обнаружения вторжения, хотя они имеют разную степень на¬

дежности. Использовать такую систему в качестве единственной меры безопасности было бы ошибкой.

Ничто не препятствует применению этих концепций в физическом мире. Многие системы используют устройства против вторжения. Это не обязательно плохо: устройства, осуществляющие сопротивление вторжению, защищают системы от большинства людей и от большинства атак. Меня беспокоит, когда полностью полагаются на средства сопротивления вторжению, вместо того чтобы рассматривать их как один из аспектов более разносторонней системы безопасности.

В качестве примера системы, которая эффективно использует средства сопротивления вторжению как часть общего механизма управления, приведу систему контроля за ядерными вооружениями в США. Риск реален: какой-нибудь недобросовестный командир может осуществить запуск оружия без разрешения, или случится, что тактическое ядерное оружие будет украдено или (если оно находится на заокеанской базе) попадет в руки союзников во время кризиса. Необходимо быть уверенным, что запуск ядерного оружия произойдет только при наличии соответствующей директивы из Вашингтона. Для решения этой проблемы используется система, называемая PAL (permissive active link, разрешающая активная связь), детали которой до сих пор являются секретом. Мы знаем только, что PAL может действительно быть полезной, если она запрятана глубоко внутри большой и сложной системы вооружений. Более простые виды вооружений хранятся в специальных контейнерах — это защитные системы предписанного действия (prescribed action protective systems, PAPS), которые наилучшим образом обеспечивают сопротивление вторжению.

Система сопротивления вторжению, обеспечивающая безопасность ядерных устройств, включает в себя разнообразные «ловушки для дурака»: химические вещества, которые делают ядерный материал непригодным к использованию, небольшие взрывные устройства, способные разрушить критические компоненты оружия и самого атакующего, и т. д. Только шифровальный код, поступивший из Вашингтона, сумеет отключить эти защитные механизмы и привести в действие ядерное оружие.

Эти защитные механизмы экстремальны, но и ситуация подобающая. Бывают экстремальные ситуации в коммерческом мире — ключи корневых сертифицирующих органов (см. главу 15), ключи, используемые банками для осуществления безопасных межбанковских телеграфных переводов, — но меры безопасности, существующие в подобных заботливо сконструированных системах, не являются продуктом массового производства. В обычных случаях меры безопасности в коммерческом мире значительно более примитивны.

И есть фундаментальное различие в системе управления. Ядерные вооружения находятся под экстремальным физическим контролем; это делает меры по сопротивлению вторжению более эффективными.

Представьте себе игровой автомат. Автомат находится на безопасной территории. Если вы сумеете вскрыть автомат, вы можете забрать оттуда все деньги или, что еще серьезнее, изменить микросхемы, так что он выплатит джекпот. Но этот автомат стоит на полу в казино. Там освещение, камеры наблюдения, охрана, люди... если кто-то подойдет слишком близко с дрелью или отверткой, он будет арестован. Теперь представьте, что в казино вам говорят нечто вроде: «Это игровой авто¬

мат. Возьми его домой. Играй сколько тебе угодно. Принеси его назад через несколько месяцев. Все выигрыши мы выплатим».

Это совсем другая ситуация. Злоумышленник принесет автомат в свою базовую лабораторию. Он может изучать машину как угодно. Он просветит ее рентгеном. Он может даже купить несколько таких же машин у производителей и разобрать их. В конце концов, у него намного больше возможностей атаковать систему, пользуясь своей базой, чем находясь в казино. И это верно не только для игровых автоматов, но и для банкоматов, банковских сейфов и тому подобных моделей безопасных систем.

(Это не означает, что игровой автомат, стоящий в казино, неуязвим. Деннис Никраш обеспечил себе хорошую жизнь — всего около 16 миллионов долларов, — взломав игровой автомат. Он практиковался на игровых автоматах дома и в конце концов понял, как вскрыть один из них в казино так, чтобы при этом не включилась сигнализация. Ему удалось заменить некоторые микросхемы. Системы блокировки (blockers) скрывали его от телекамер. В итоге он взял на «обработанной» им машине джекпот.)

Мораль этого раздела проста. Первое — сопротивление вторжению в значительной степени миф, но оно создает определенный барьер при входе в систему. Второе — средства сопротивления вторжению должны использоваться совместно с другими контрмерами. И третье — любая система, в которой устройства и секреты, заключенные внутри этих устройств, находятся под контролем разных людей, имеет фундаментальный изъян в системе безопасности. Можно конструировать подобные системы, но надо при этом осознавать, что они неотделимы от этого недостатка.

## Нападения через побочные каналы

В последние годы в литературе стали появляться описания новых видов крипто¬аналитических атак: атаки, целью которых являются определенные детали выполнения. *Тайминг-атака (timing attack, атака, основанная на сравнительных измерениях времени)* вызвала большой шум в прессе в 1995 году: закрытые ключи RSA могли быть восстановлены с помощью измерения относительных интервалов времени операций шифрования. Такие атаки были успешно проведены против смарт-карт и других опознавательных знаков доступа, а также против серверов электронной коммерции в Интернете.

Исследователи обобщают эти методы, относя к ним нападения на системы с помощью измерения энергопотребления, уровня излучения и использования других *побочных каналов*. Такие атаки могут осуществляться против разнообразных симметричных алгоритмов шифрования с открытым ключом в устройствах аутентификации, снабженных системами сопротивления вторжению. Обобщающие исследования представили анализ нападения: в процессор, осуществляющий шифрование, преднамеренно вводятся ошибки, что дает возможность определить секретные ключи. Последствия такой атаки могут быть разрушительными.

Допустим, нападающий хочет узнать секретные ключи, находящиеся внутри модуля, защищенного от вторжения: смарт-карты, карты PCMCIA или чего-нибудь подобного. Он не может осуществить криптографический анализ алгоритмов

или протоколов (они слишком хороши) и не способен взломать систему сопротивления вторжению. Но атакующий умен; вместо того чтобы анализировать только входные и выходные данные, он обращает внимание также на скорость, с которой модуль проводит операции. Критическое рассмотрение атаки измерения времени показывает, что выполнение многих операций шифрования происходит с различной скоростью для различных ключей. Знание скорости, с которой происходит определенная операция, дает информацию о ключе. Знание множества различных скоростей для различных операций помогает получить полную информацию о ключе.

Представьте себе атаку, направленную против склада: вы хотите узнать, что находится внутри. У вас нет возможности заглянуть на склад, чтобы посмотреть, как там расположены вещи. Однако вы можете попросить служащего «получить» материал для вас. Время, которое ему потребуется для того, чтобы сделать ту или иную вещь, скажет вам многое об этом складе. Он каждый раз тратит много времени на то, чтобы принести картридж? Должно быть, картриджи находятся в самом дальнем углу. Расходует ли он больше времени на то, чтобы принести стопку бумаги, через каждые 10 запросов? Тогда, должно быть, бумага сложена в коробки по 10 пачек. Ему требуется больше времени на то, чтобы принести карандаши, чем на то, чтобы принести резинки? Это говорит о том, какие коробки лежат сверху.

Рассмотрим тайминг-атаку против устройства проверки пароля. Берем случайный пароль и варьируем только первый символ. Так как всего букв 26, можно использовать строчные и прописные, еще 10 цифр, несколько знаков пунктуации, итого около 70 паролей. Возможно, один из них устройство будет проверять дольше, чем остальные, прежде чем отклонить. Возможно, это пароль с правильным первым символом. Повторим то же самое с остальными символами. Если вы предпринимаете попытку атаковать пароль из 8 символов, вам нужно проверить всего 560 паролей и измерить соответствующие задержки.

Атакующий не обязательно должен ограничиться тайминг-анализом. Он может посмотреть, сколько энергии затрачено на различные операции. (В случае, когда модуль потребляет разное количество энергии при осуществлении одинаковых по сути операций, в зависимости от ключа.) Он может исследовать также, сколько тепла излучается и даже где в модуле находится источник излучения. Например, *атаки измерения энергии {power attacks)* были применены, чтобы раскрыть секреты почти всех смарт-карт, имеющихся на рынке.

Все эти атаки допустимы, поскольку модуль находится в руках атакующего. Если модуль помещен в хранилище, закрытое на замок, злоумышленник не сможет проводить эти виды атак. (Хотя он, вероятно, окажется в состоянии осуществить нападение против другой копии такого же продукта, которая может обеспечить ему получение некоторой интересной информации.) Но в случае, когда проектировщики систем полагаются на аппаратные средства защиты от вторжения и дают возможность нападающему получить копию модуля, они тем самым убирают препятствия для проведения подобных систематических атак.

Иногда возможно осуществлять некоторые нападения удаленно, через сеть. Здесь не получится наблюдать за потерями тепла и энергопотреблением, но вы сможете измерять временные интервалы. Конечно, в сети всегда присутствует не¬

кий шум, что не особенно мешает получить результат математически. Или вы можете наблюдать излучение (военные называют это TEMPEST).

TEMPEST заслуживает более подробных объяснений, если по каким-то причинам различные военные ведомства тратят кучу денег на защиту против него. Система нападения извлекает информацию, содержащуюся в излучении электронной аппаратуры, используя чувствительный радиоприемник, специально настроенный на нужный канал, чтобы воспринимать нужную информацию. (Это излучение также называют излучением ван Эка.) Видеомониторы, возможно, наиболее уязвимы — с «правильным» оборудованием вы сможете читать содержимое экрана чужого компьютера, несмотря на блокировку, — но утечка информации в той или иной степени существует везде. Сотовые телефоны, факсимильные аппараты и коммутаторы — через них тоже утекает информация. Не имеет значения, что данные в этих устройствах могут быть зашифрованы; как зашифрованные, так и «открытые» данные излучают, и обладающий достаточными ресурсами атакующий способен отличить одно от другого. Кабели ведут себя как антенны; от них также распространяется излучение, несущее информацию. Линии электропередач — это трубопроводы, по которым течет информация. Это нетривиальная атака, которая может потребовать массу специального оборудования. Иногда это легко — прочитать информацию с экрана чьего-нибудь компьютера, — но в других случаях оказывается сложным и трудоемким.

Правительственное решение проблемы излучения — это экранирование. Военные покупают компьютерное оборудование, которое экранировано от TEMPEST. Когда они создают оборудование для шифрования, они тратят дополнительные деньги, чтобы быть уверенными, что открытый текст не утекает через линии передачи зашифрованных данных или из устройства для шифрования. Они покупают экранированные кабели и для передачи данных и для электропитания. Они даже строят TEMPEST-экранированные помещения или, в критических случаях, целые здания: все это называется SCIFs (Secure Compartmented Information Facilities, средства безопасности для предотвращения утечки информации).

Существуют и другие атаки с использованием побочных каналов. Иногда нагревание или охлаждение модуля может предоставить в ваше распоряжение интересную информацию; в других случаях результат даст изменение напряжения на входе. Один безопасный процессор, например, предоставлял доступ к секретным данным, если входное напряжение мгновенно понижалось. Другой имел генератор случайных чисел, который выдавал их все, если напряжение медленно понижалось. Другие модули уязвимы, когда вы задействуете тики генератора импульсов синхронизации.

Подумайте обо всем этом как о неагрессивных биологических экспериментах. Вы можете многое узнать об организме, если будете просто наблюдать за ним: что он ест, что он выделяет, когда он спит, сколько времени ему требуется для выполнения определенной задачи в различных случаях, в условиях тепла или холода, сырости или сухости. Нет необходимости его вскрывать: вы можете многое узнать о нем в процессе его нормального функционирования.

Вскрыть его всегда интересно, особенно если вы сумеете сделать это, не убивая его. Если мы взломаем систему сопротивления вторжения и не разрушим модуль, мы узнаем множество подробностей о его системе безопасности.

*Анализ ошибок* — еще одна мощная атака, поскольку шифрование чувствительно к малым изменениям. В главе 7 я говорил о том, как легко это применить к некорректной системе шифрования, нарушив ее безопасность в процессе функционирования. При анализе ошибок аналитик намеренно вводит ошибки в процесс осуществления шифрования — в особых точках, которые обеспечат максимальную утечку информации. Комбинируя это со взломом системы сопротивления вторжению — действуя на ключевые точки в том и другом направлении (не случайно, но специальным образом), — можно провести разрушительную атаку против безопасных модулей.

Систематические атаки недешевы. Маловероятно, что они будут выполнены преступниками-одиночками или сообществом террористов. Это атаки, которые могут осуществляться хорошо финансируемыми противниками: организованной преступностью, промышленными конкурентами, военными разведывательными организациями и академическими лабораториями. Но они работают, и работают хорошо. Системы, подобные смарт-картам, выполняли бы свои функции надежно, если бы заранее предполагалось, что систематические атаки возможны, и была уверенность, что даже в случае их успешного осуществления безопасность системы не будет нарушена.

Атаки с использованием побочных каналов не всегда можно распространить на любые системы. Нападение, основанное на анализе ошибок, невозможно провести, если процесс шифрования осуществляется таким образом, что атакующий не имеет возможности создать и использовать нужные ошибки. Но эта атака намного сильнее стандартных криптаналитических атак против алгоритмов. Например, известная атака против DES с использованием анализа ошибок требует от 50 до 200 блоков зашифрованного текста (без открытого текста) для того, чтобы восстановить ключ. Она работает только на определенных аутентификаторах DES, будучи выполненной определенным образом. Контраст с этим составляет лучшая атака без использования сторонних каналов, которая требует около 64 Тбайт открытого текста и зашифрованного текста для определения единственного ключа.

Некоторые исследователи заявляют, что это обман. Правильно, но в системах, существующих в реальном мире, атакующие всегда обманывают. Их работа — восстановить ключ, а не следовать неким произвольным правилам выполнения атаки. Дальновидные разработчики систем безопасности понимают это и приспосабливаются к обстановке. Мы верим, что наиболее действенные атаки используют информацию, получаемую через побочные каналы. Звук — тоже побочный канал; о прослушивании вращательного движения роторов электромеханических двигателей упоминал в своей книге «The Codebreakers» Дэвид Кан. Военные Соединенных Штатов долгое время разбирались с TEMPEST. И в своей книге «Охотник за шпионами» («Spycatcher») Питер Райт обсуждает утечку секретных данных через линию передач по побочному каналу (метод, известный среди военных как HIJACK), которая позволила взломать шифровальный механизм, использовавшийся французами.

Осуществить защиту трудно. Вы можете либо сократить количество информации, утекающей через побочные каналы, либо сделать эту информацию не соответствующей действительности. Оба метода имеют свои проблемы, хотя исследователи работают над ними. Более дорогие устройства оборудованы датчиками, позволяющими обнаружить подключение на входе, — регуляторами, определяю¬

щими падение напряжения, термометрами, фиксирующими попытки охладить устройство, часами, устойчивыми к внешним сбоям, — и не разрушить свои секреты. Другие устройства чувствуют, что их вскрыли, и реагируют подобающим образом. Но эти виды защитных мер преимущественно используются только в системах, которые покупают военные, и даже не применяются в таких безопасных устройствах, как смарт-карты.

Атаки с использованием побочных каналов очень эффективны, и это будет до тех пор, пока не существует хорошей теории, на которой могла бы строиться защита. В любом случае система, в которой устройство находится в руках одной персоны, а секреты, содержащиеся внутри, принадлежат другой, рискованна.

## Атаки против смарт-карт

Смарт-карты могут рассматриваться как своего рода «заговоренные пули» компьютерной безопасности — многоцелевой инструмент, который находит применение для контроля доступа, в электронной коммерции, аутентификации, защите секретности и других приложениях. В основном разработчики используют их свойства безопасной территории: процессор и память, находящиеся внутри, неуязвимы (предположительно) для атак. Также они маленькие, переносимые, дешевые и гибкие. Это делает их привлекательными, но недостатки прямого ввода-вывода, применяемого в смарт-картах, придают им большую уязвимость.

Наиболее интересно в смарт-картах то, что существует большое число сторон, вовлеченных в любую систему на основе смарт-карт. Это означает, что смарт-карты чувствительны ко многим видам атак. Большинство этих атак невозможно в обычных компьютерных системах, так как они могли бы иметь место только в случае проникновения внутрь безопасной территории. Но по отношению к смарт-картам все перечисленные ниже атаки вправе вполне обоснованно считаться опасными.

**Атаки со стороны терминала против владельца карты или против того,** кому **принадлежат данные.** Когда владелец карты опускает свою карту в терминал, он полагается на то, что терминал правильно осуществит ввод-вывод данных. Безопасность большинства систем с использованием смарт-карт основывается на том факте, что терминал имеет доступ к карте в течение короткого промежутка времени. На самом деле безопасность имеет отношение не только к обмену данными между смарт-картой и терминалом: должна быть законченная система обработки, которая способна отслеживать карты, терминалы и сигнализировать об их подозрительном поведении.

**Атаки со стороны владельца карты против терминала.** Более тонкими являются атаки владельцев карт против терминала. К ним относятся подмена или изменение карт с использованием программного обеспечения в жульнических целях, для того чтобы нарушить работу протокола обмена между картами и терминалом. Хорошо сконструированный протокол снижает риск подобной атаки. Риск угрозы падает еще больше, если карты имеют физические характеристики, которые сложно подделать (например, голограмма Visa-карты) и которые могут быть непосредственно проверены владельцем терминала.

**Атаки со стороны владельца карты против собственника данных.** В большинстве коммерческих систем, использующих смарт-карты, данные, хранящиеся на карте, должны быть защищены от ее владельца. В некоторых случаях владелец карты даже не должен знать эти данные. Если карта содержит значение денежной суммы, и ее владелец сумеет изменить это значение, он сможет эффективно создавать деньги из ничего. Было много успешных атак против данных, содержащихся в смарт-картах.

**Атаки со стороны владельца карты против того, кто их выпускает.** Многие финансовые атаки были направлены против того, кто выпускает смарт-карты, но фактически против целостности и подлинности данных или программ, хранящихся на карте. Если изготовитель карт поместил биты, которые ответственны за разрешение на вход в систему, на карту, он не должен удивляться, если эти биты будут атакованы. Подобные системы основываются на сомнительном предположении, что безопасной территории внутри карты достаточно для решения задачи, перед ней поставленной.

**Атаки со стороны владельцев карт против производителей программного обеспечения.** Вообще говоря, даже в системах, рассчитанных на враждебного пользователя, предполагается, что он не будет загружать новое программное обеспечение на карту. В некоторых случаях это предположение может оказаться неверным.

**Атаки со стороны владельца терминала против изготовителя карт.** В некоторых системах владелец терминала и тот, кто выпускает карты, являются различными субъектами. Это разделение создает несколько новых возможностей для атаки. Терминал управляет всеми соединениями между картой и ее изготовителем, и всегда можно сфальсифицировать записи или помешать успешному завершению одного или нескольких шагов транзакции, для того чтобы совершить мошенничество или создать проблемы в обслуживании пользователей.

**Атаки со стороны изготовителя карт против владельцев карт.** В общем, большинство систем основано на изначальном предположении, что тот, кто выпускает карты, искренно и наилучшим образом заботится об интересах владельцев карт. Но это не всегда так. Эти атаки представляют собой типичные случаи нарушения секретности тем или иным образом. При создании системы смарт-карт, которая должна служить заменой наличных денег, надо позаботиться о том, чтобы сохранились существенные свойства наличных денег: анонимность и обезличенность.

**Атаки против собственников данных, возникающие вследствие конструктивных недостатков программного обеспечения.** Некоторые конструктивные особенности программного обеспечения могут создавать существенные и неприятные эффекты для собственников данных в системе. Если производители реализовали операционную систему, которая позволяет многим пользователям запускать программы на одной и той же карте, это приводит к появлению ряда новых проблем безопасности, таких как разрушительная деятельность операционной системы, преднамеренно ограниченный генератор случайных чисел или конфликт различных приложений, запущенных на одной смарт-карте.

Все это не означает, что смарт-карты нельзя рассматривать как защищенные устройства. Смарт-карты, которые обращаются в кредитной финансовой системе,

например, очень отличаются от карт, которые используются в системе денежных вкладов (stored value system). Системы смарт-карт, которые обеспечивают идентификацию и возможность проверки, также безопасны. Смарт-карты полезны, но их использование сопряжено с известным риском. Безопасная система смарт-карт должна уметь распознавать вышеописанные атаки; она должна быть сконструирована таким образом, чтобы они не могли на нее повлиять. В лучших системах не имеет значения, например, что пользователь может взломать карту. Это очень «по-дзенски»: работа не против возможного нападающего, а на основе модели сохранения безопасности.

# Глава 15. Сертификаты и удостоверения

Понятия *сертификата открытого ключа* и *инфраструктуры открытого ключа* являются центральными для шифрования в современном Интернете. Прежде чем их рассматривать, однако, будет нелишне напомнить, что представляет собой цифровая подпись. Цифровая подпись является математической операцией над совокупностью битов, которую можно выполнить только с помощью определенного ключа. Ее подлинность может быть подтверждена с помощью другого, соответствующего первому, ключа. Допустим, ключ шифрования известен только Алисе. Следовательно, понятно, что только Алиса может выполнить операцию и таким образом «подписать» сочетание битов.

Проблема с описанной моделью заключается в следующем: предполагается, что ключ шифрования — это секрет, известный только Алисе. Путем проверки подписи мы можем действительно подтвердить только то, что сообщение подписано с помощью Алисиного ключа; мы не вправе ничего утверждать относительно того, подписала ли его сама Алиса. Мы не знаем, был ли ключ украден кем-либо еще. Мы не знаем ничего относительно намерений Алисы. Когда мы видим Алисину собственноручную подпись на бумаге, мы можем сделать некоторые утверждения относительно ее воли: она прочла и подписала документ, она поняла его содержание. Когда же документ подписан с помощью секретного ключа Алисы, мы даже не можем быть уверены, что она вообще видела его. «Цифровая подпись» — это название, которое настораживает, потому что это не есть подпись.

Это будет важно в дальнейшем изложении темы этой главы. А пока что поговорим о доверенных третьих лицах.

## Доверенные третьи лица

Специалисты по криптографии определяют доверенное третье лицо как персону, которой доверяют все участники протокола, помогающую придать протоколу завершенность и безопасность. Мой друг из NSA однажды заметил (определение, отличающееся замечательной ясностью): «Каждый, кто вас знает, может нарушить вашу секретность и не быть пойманным». Вполне достаточно, эти два определения в главном совпадают.

Помните различные протоколы доверенных третьих лиц из главы 7? Любая коммерция, исключая прямой бартер, тем или иным образом использует доверенных третьих лиц. Даже сделки с оплатой наличными: продавец верит, что прави¬

тельство примет деньги, которые он получил от покупателя. Когда же сделки включают в себя более интересные финансовые инструменты — чеки, кредитные карты, дебетовые карты, командировочные удостоверения, — и продавец и покупатель полагаются на то, что банк или финансовая компания будут вести себя правильно. Предприниматель и клиент не обязательно доверяют друг другу, но доверенное третье лицо должно успешно посредничать между ними. Все заканчивается очень быстро, если компания кредитных карт начинает капризно отвергать карты некоторых владельцев.

Адвокаты играют роль доверенных третьих лиц в персональных делах, выполняя волю свих клиентов. Когда некто говорит своему похитителю: «Если вы убьете меня, то мой адвокат пошлет копии доказательств в ФБР, CNN и *New York Times,* он использует своего поверенного в качестве доверенного третьего лица. Адвокаты не думают шутить, профессия делает их превосходными доверенными лицами.

Судебная система в целом может рассматриваться как доверенное третье лицо, гарантирующее, что контракты законны и дела ведутся правильно. Рассмотрим такой протокол честного контракта: Алиса и Боб ведут переговоры и подписывают контракт. Если один из них почувствует, что другой не соблюдает контракт со своей стороны, он или она обратятся к доверенной третьей стороне — к судье. Судья выслушает доказательства с обеих сторон и затем свершит правосудие.

Это работает, если они оба верят, что судья честен. В тех случаях, когда юридическая система коррумпирована или ее представители некомпетентны, мы видим крайне малую степень доверия контрактам и совершенно другую систему правил ведения коммерции.

Современная жизнь полна примеров существования доверенной третьей стороны. Магазины, торгующие по накладным, неважно, реальные или в Сети, суть доверенные третьи лица. Также и аукционы. А каждый, кто покупает что-либо через службу доставки? Служба доставки — доверенное третье лицо. То же самое — почтовые переводы. Нотариально заверенный акт подтверждает подлинность личности людей, подписавших законные документы, и обеспечивает проверку доказательств в случае спора. ООН посылает «наблюдателей» в качестве доверенных лиц в те страны мира, где противоборствующие стороны не доверяют друг другу. В Сети появляется служба условного депонирования, действующая как доверенная третья сторона между продавцами и покупателями в случае сделок с привлечением крупных денежных сумм.

В США существует целая индустрия доверенных третьих лиц, проводящих сделки с недвижимостью. Эти компании действуют как доверенные третьи лица между всеми сторонами, вовлеченными в сделки с недвижимостью: между продавцом и покупателем, банком продавца и банком покупателя, агентом по недвижимости продавца и агентом по недвижимости покупателя. Все эти стороны полагаются на агентство по недвижимости в том, что сделка будет завершена успешно.

Доверенные третьи лица играют особо важную роль в электронном мире. В мире, в котором отсутствует контакт лицом к лицу, а порой и голосовой контакт, в котором вы вынуждены полагаться на криптографию и устрашающе ненадежную компьютерную безопасность, это единственное реальное подтверждение, которое вы можете получить.

Помните целую систему открытых ключей, которую мы с вами обсуждали в главе 6? Алиса хочет послать зашифрованное сообщение, используя открытый ключ Боба, и для этого просматривает базу данных открытых ключей, чтобы найти его. Она получает *сертификат открытого ключа* Боба. Это сообщение, подписанное кем-то еще, которое удостоверяет, что данный ключ действительно принадлежит Бобу. Персона, которая подписала его, — *это доверенное третье лицо*.

Доверенные третьи лица — это рычаг системы безопасности, поскольку обращение к ним присуще любой системе, которой требуются гарантии безопасности. В плохо сконструированных системах доверенным третьим лицам права предоставляются без учета возможных неожиданностей. Блестяще сконструированные системы вручают мандат доверия по закону.

## Удостоверения

Откройте ваш бумажник. Внутри вы обнаружите все виды удостоверений. У вас есть банковская карта. Это удостоверение, предоставленное вам вашим банком; оно необходимо, чтобы подтвердить подлинность вашей личности, когда автоматическая система выдает вам наличные. Вы имеете кредитную карту — удостоверение, врученное вам банком, и можете расплачиваться с ее помощью. У вас есть водительские права, выданные правительством. Они доказывают, что вы имеете все привилегии водителя.

Компании в зарубежных странах, дающие в аренду автомобили, рассматривают это удостоверение как подтверждение вашей способности управлять машиной, но полиция обычно видит в нем лишь легкий путь узнать ваше имя и адрес: они сверяют ваши права со своей базой данных. (Есть несколько историй, когда водительские права использовались в качестве удостоверения, прежде чем вам разрешали расплатиться чеком; отношения с доверенными третьими лицами базируются на предпосылке, что правильная информация об идентификации помогает судебному расследованию.) У вас также есть карта клиента авиационной компании, читательский билет, членский билет какого-нибудь спортивного клуба и бог знает что еще. Если у вас есть паспорт — это еще одно удостоверение.

Все эти удостоверения предоставлены различными доверенными третьими лицами, и каждое из них действует в той области, в которой вручившее его лицо является доверенным. Удостоверения не являются взаимозаменяемыми. Банку доверяют выдачу кредитных карт людям, которые имеют счета. Карта вместе с личным идентификационным номером и актуальной базой данных, дающей возможность видеть баланс на вашем счете, позволит вам снять деньги со счета. Водительские права, поскольку это удостоверение, выданное государственным органом, могут также подтвердить ваш возраст при посещении бара. Бар принимает поручительство государства при проверке возраста. (Я видел бары, которые принимали в качестве доказательства возраста водительские права, выданные государственными органами удостоверения личности и заграничные паспорта, но не паспорта США. Это лишено смысла.) Но если вы хотите заказать что-нибудь, водительские права вряд ли будут полезны. Бар доверяет государству подтвердить вашу дату рождения, но не вашу платежеспособность.

Каждое из доверенных третьих лиц имеет свои собственные правила, которым следует при вручении удостоверений. Чтобы получить паспорт, вы должны предоставить доказательства наличия у вас гражданства и подлинности вашей личности. Чтобы получить водительские права, вы обязаны сдать экзамен и предоставить доказательства того, что живете в том штате, в котором вы хотите их получить. Компания, выпускающая кредитные карты, соберет о вас кучу личной информации, проверит ее в некоей большой базе данных, и лишь после этого вы получите удостоверение. Удостоверение может поначалу обеспечивать небольшой кредит, но если вы и дальше будете строить отношения как постоянный пользователь, он будет возрастать.

Эти физические удостоверения также иллюстрируют проблему аннулирования. Что случится, если ваше удостоверение будет аннулировано? Когда управляющий банка (Master-Card) аннулирует вашу кредитную карту, он не может залезть в ваш бумажник и вытащить ее оттуда. Так что он «аннулирует» ее где-то в базе данных: он просто пометит соответствующий номер карточки как недействительный. Это работает до тех пор, пока тот, кто принимает кредитные карты, сверяется с базой данных. Если вы находитесь в какой-нибудь глубинке, где нет даже телефона, может не найтись способа подтвердить действительность вашего удостоверения.

Другой способ обходиться с аннулированными удостоверениями — это ограничить период времени, когда они могут использоваться, не будучи вновь подтвержденными. Почти все удостоверения имеют срок годности, даже такие, как читательские билеты. (Единственный контрпример, я думаю, знаки принадлежности к некоей корпорации. Это «глухой» номер. Хотя намного более вероятно, что вы смените работу, чем то, что вы разучитесь водить.) Удостоверения недействительны по истечении срока годности, что знает каждый, кто неосторожно пытался заплатить по просроченной кредитной карте или (не дай Бог) вернуться в США с просроченным паспортом. Если срок годности вашего удостоверения истек, вы должны получить новое. Иногда вы обязаны явиться за ним лично (как в случае с паспортом), иногда вам посылают его автоматически, например новую кредитную карту.

Срок годности обеспечивает безопасность. Негодные удостоверения могут существовать некоторое время, но в конце концов окажутся просроченными. Компания кредитных карт должна хранить записи об аннулированных картах только в течение этого срока, так как по его истечении они в любом случае окажутся непригодными. Доверенное лицо, которое выпускает эти удостоверения, может регулировать срок годности, чтобы обеспечить наиболее удобный режим. Ваша первая кредитная карта будет действительна в течение полугода или года, даже в том случае, если вы не оправдываете доверия. Спустя какое-то время ваша карта может обновляться раз в три года. Водительские удостоверения (по крайней мере в Иллинойсе) устаревают через четыре года. Паспорт США действителен в течение 10 лет. Можно представить себе, что в тех сферах, где мошенничество является обычным делом, удостоверения должны обновляться каждую неделю, или каждый день, или каждый час. Это большая головная боль, если в нашем распоряжении только ручка и бумага, но это прекрасно работает в киберпро¬странстве.

Так что мы нуждаемся в удостоверениях, работающих в киберпространстве. Мы хотим иметь цифровой эквивалент кредитных карт, удостоверений, подтверждающих возраст, знаков принадлежности к корпорации, читательских билетов, членских билетов и т. п.

# Сертификаты

Сертификат — это удостоверение... один из его видов. Это способ идентифицировать вас, но безотносительно к тому, что вы представляете собой в реальности. И это означает для любого, что можно доверять... может быть. Это определенно не то же самое, что открытый ключ.

Я думаю, нужно начать с начала.

Помните главу 6 и шифрование с применением открытого ключа? Алиса использует это шифрование для цифровой подписи. Она заверяет документы своим закрытым ключом и посылает подписанный документ Бобу. Теперь Боб нуждается в открытом ключе Алисы, чтобы проверить ее подпись. Как он получит его и в каком случае он может быть уверен, что ключ действительно Алисин?

На ранней стадии развития шифрования с использованием открытого ключа предполагалось, что будут существовать обширные базы данных, содержащие открытые ключи, подобно телефонным книгам. Боб мог бы посмотреть Алисино имя в такой базе данных и отыскать соответствующий этому имени открытый ключ.

Хорошо, но если все открытые ключи будут храниться где-то в большой базе данных, что можно сказать о безопасности этой базы данных? Злоумышленник способен натворить кучу бед, если сможет заменить один открытый ключ другим. Он может создать новый открытый ключ, подписать им пачку чеков и затем поместить его в базу данных рядом с именем Алисы. Таким образом, она подписала все эти чеки. Если Боб использует открытый ключ Алисы, чтобы зашифровать сообщение для нее, злоумышленник в силах заменить открытый ключ Алисы на свой; тогда секретное сообщение для Алисы сможет расшифровать взломщик, но не она сама.

Мы должны быть в состоянии защитить базы данных открытых ключей, но пригодна ли сама идея обеспечить свободный и широкий доступ к открытым ключам? Надо признать, что это не работает.

Сертификаты являются решением обозначенной проблемы. Сертификат представляет собой связь между открытым ключом и подтверждением подлинности. Пользующийся всеобщим доверием объект — назовем его Богом — берет имя Алисы и ее открытый ключ, связывает их и затем подписывает все это. Теперь Боб может не беспокоиться. Он получил Алисин сертификат открытого ключа — его не очень заботит, откуда — и проверил подпись Бога на нем. Боб доверяет Богу, так что, если он удостоверился в подлинности подписи, он знает, что открытый ключ принадлежит Алисе, а не какому-то обманщику. Проблема решена; мир теперь безопасен для электронной коммерции.

Однако не совсем. Заметим, что в действительности мы не решили проблему. Все, что мы сделали, возвращает нас к первоначальному вопросу: «Как Боб узнает, что открытый ключ Алисы действительно принадлежит ей?» Или изменим фор¬

мулировку: «Как Боб узнает, что открытый ключ Бога действительно его ключ?». Боб проверил подлинность подписи Бога на сертификате, прежде чем начал использовать Алисин ключ, так что ему необходим был открытый ключ Бога. И где он его мог получить?

Но мы все же решили кое-что. Вероятно, Боб хочет взаимодействовать со множеством людей, не только с Алисой. И если Бог подписал каждый сертификат, мы свели проблему Боба к меньшей проблеме: теперь вместо подтверждения открытого ключа каждого нужно подтвердить лишь один ключ — Божий. Но эту проблему мы обсудим позже.

Настоящий сертификат представляет собой нечто немного более сложное. Он содержит информацию о персоне (имя, возможно, место работы, возможно, электронный адрес и другие сведения), информацию о сертификате (кем он выпущен, каков его срок годности), информацию об изготовителе сертификата или о том, кто его подписал (кто он, какой алгоритм он использовал для подписания сертификата и информацию относительно открытого ключа (какой алгоритм))... и собственно открытый ключ.

Основная идея состоит в том, что Алиса некоторым образом получает подписанный Богом сертификат открытого ключа. Либо она создает пару открытый ключ/закрытый ключ и отправляет открытый ключ Богу, который возвращает ей сертификат открытого ключа, либо Бог сотворит пару открытый ключ/закрытый ключ для Алисы и пошлет ей закрытый ключ и сертификат открытого ключа. (Здесь мы сталкиваемся с проблемой безопасности *этого* обмена, но пока что не берем ее в голову.)

Это все работает великолепно до тех пор, пока Алиса не потеряет свой закрытый ключ. Может быть, кто-то украдет его. Может быть, она забудет его. (Или, что более правдоподобно, ее компьютер «полетит» и не останется резервной копии.) Боб собирается попытаться послать ей сообщение, зашифрованное этим потерянным ключом. Или, хуже того, Боб попытается проверить подлинность подписей, созданных после того, как некто украл ключ. Что мы будем делать тогда?

Мы обратимся к Богу, и он аннулирует Алисин сертификат. Он объявит старый сертификат недействительным, неправильным и непригодным. Как он это сделает? Бог не может обшарить каждый уголок в Сети и уничтожить каждую копию сертификата. (Конечно, может быть, Господь Бог и всемогущ, но это только аналогия.) Возможно, он даже не знает, что у Боба есть такая копия.

Так что Бог включает Алисин сертификат в *список аннулированных сертификатов,* или *CRL (certificate revocation list)*. (Вспомним, как 20 лет назад торговцы публиковали списки номеров недействительных кредитных карт. Это CRL.) Бог выпускает CRL регулярно (компании кредитных карт делают это раз в неделю), и это уж дело Боба — удостовериться, что Алисин сертификат не находится в последнем списке, прежде чем он использует его. Он должен также убедиться, что сертификат не просрочен и что он в действительности принадлежит Алисе.

Каким образом Боб может осуществить последнее? Он сравнит имя Алисы с указанным на сертификате. Если они совпадают, сертификат принадлежит ей. Это звучит просто, однако ж не работает.

Это отдельная проблема. Во-первых, никто не может выступать в роли Бога. Или, более точно, не существует организации или объекта, с которым может дого¬

вориться каждый и чье правосудие не подлежит сомнению. Во-вторых, Алиса может не иметь уникального имени, относительно которого все согласились.

Разберемся со всем по порядку. Сначала первая проблема. Вспомним, чтобы система в целом работала, Алиса должна получить свой сертификат от кого-либо, кому и она, и Боб доверяют. В действительности существует иерархия доверенных лиц, определяющая ценность выданного сертификата. Военная организация, возможно, лучший пример такого рода системы. Командир взвода подписывает сертификат каждому бойцу в своем взводе. Командир дивизии заверяет сертификаты каждому командиру взвода в его подчинении. Генерал армии подписывает сертификаты своим дивизионным командирам. И так далее, до главнокомандующего.

Теперь у Алисы есть целая цепь сертификатов: от главнокомандующего — к генералам армий, от них — к дивизионным командирам, от них — к командирам взводов, и наконец, от командира ее взвода — к ней самой. Она хранит их все и представляет их Бобу. Если она и Боб числятся в одном взводе, у Боба также есть сертификат, выданный командиром взвода. Он знает, как должен выглядеть действительный сертификат, так что может непосредственно проверить и подтвердить ценность Алисиного сертификата. Если они в одной дивизии, но в разных взводах, они оба обладают одинаковыми сертификатами от дивизионного командира. Боб может использовать для подтверждения сертификат Алисиного командира взвода и затем сертификат Алисы. Поскольку Алиса и Боб числятся в одних войсках, каждый из них включен в свою цепь командования. Возможно, потребуется даже сертификат от главнокомандующего, который в данном случае и есть Бог.

Эта система замечательно работает у военных, но гораздо хуже в гражданском обществе. В Интернете сертификаты используются для поддержки множества протоколов: IPsec (Internet Protocol security, интернет-протокол безопасности) и различные системы VPN (Virtual Private Network, виртуальная частная сеть), SSL (Security Socket Layer, протокол безопасности сокета), несколько протоколов электронной коммерции, некоторые протоколы проверки входа в систему. Соответствующие сертификаты выдаются пользователям некоторой инстанцией, называемой *бюро сертификации (certificate authority, CA)*. CA может быть корпоративной организацией. Правительство также способно выступать в этом качестве. Оно может быть частной компанией, которая сделала своим бизнесом изготовление сертификатов для пользователей Интернета.

Центры сертификации также нуждаются в сертификатах. (Вспомним об иерархии.) Сертификаты выдаются CA другими сертифицирующими органами (возможно, VerySign). В конечном счете мы получаем Бога в этой системе, точнее, целый пантеон Богов. Сертифицирующие органы на самом верхнем уровне имеют *корневые сертификаты:* они не подписаны кем-либо еще. Такие сертификаты включены в программное обеспечение, которое вы покупаете: ваш браузер, обеспечение частной виртуальной сети и т. д. Это все называется *инфраструктурой открытых ключей (public-key infrastructure, PKI)*. Она работает, но только частично.

Вторая проблема: имя Алисы.

В стародавние времена (примерно в середине 80-х) каждый мечтал о мире, в котором каждая персона, каждый процесс, каждый компьютер, каждый механизм связи — все, что связано с цифровыми коммуникациями,.— имели бы уникальное

имя. Эти имена хранились бы в широко распространенной базе данных, доступной множеству людей в различных регионах. Этот проект был назван X.500.

Вообще говоря, сертификаты связывают открытый ключ с уникальным именем (называемым *отличительным именем* в терминологии X.500), но, возможно, стоит обсудить, насколько полезна такая связь. Представьте себе, что вы получили сертификат, принадлежащий Джоан Робинсон. Вы, возможно, знаете лично только одну Джоан Робинсон, но сколько людей с таким именем известны центру сертификации? Как вы удостоверитесь, что конкретный сертификат Джоан Робинсон, полученный вами, принадлежит вашей подруге? Вы можете получить ее открытый ключ лично от нее, или она лично подтвердит передачу, но более вероятно, что вы примете сертификат по электронной почте и должны просто поверить, что это «правильная» Джоан Робинсон. Обобщенное имя на сертификате, вероятно, будет дополнено некоторой другой информацией, благодаря которой станет уникальным среди имен, получивших сертификат от одного CA.

Вы знаете эту дополнительную информацию о своей подруге? Знаете ли вы, какой CA выдал ей сертификат?

Вспомним аналогию с телефонным справочником. Если вы захотели найти открытый ключ Джоан Робинсон, вы просмотрели каталог, разыскали ее открытый ключ и послали сообщение, предназначенное только для ее глаз, используя этот открытый ключ. Это могло работать в случае с телефонным каталогом Стэнфорд¬ ского отделения информационного обеспечения в 1976 году, но сколько Джоан Робинсон присутствуют в телефонном справочнике Нью-Йорка, и насколько это число меньше предполагаемого числа Джоан Робинсон в гипотетическом телефонном справочнике Интернета?

Мы вырастаем в маленьких семьях, где имена служат идентификаторами. К тому времени как нам исполняется пять лет от роду, мы усваиваем этот урок. Имена работают. Это неверно для большого мира, но вещи, впитанные в младенческом возрасте, мы не забываем никогда. В данном случае мы должны внимательно продумать все относительно имен, а не полагаться слепо на опыт пятилетнего ребенка, отложившийся в нашем сознании.

Идея также предполагает, что Алиса и Боб находятся в каких-то взаимоотношениях в физическом мире и хотят перенести эти отношения в киберпространство. Помните времена, когда «киберпространство» было термином из научной фантастики, а все взаимоотношения, о которых мы говорили, — деловые, социальные, банковские, коммерческие — формировались исключительно в мире из плоти и крови? Теперь люди встречаются в Сети и формируют там свои взаимоотношения все время. Иногда им удается свидеться лично через много лет после того, как они уже стали друзьями; в некоторых случаях они так и не входят в личный контакт. В этом прекрасном новом мире система, сконструированная только для того, чтобы проецировать взаимоотношения реального мира в киберпространство, выглядит ограниченной.

## Проблемы с традиционными PKI

Инфраструктуры открытых ключей и центры сертификации несут в себе массу других проблем. Например, что может значить, когда бюро сертификации объяв¬

ляет кого-либо заслуживающим доверия? Согласно литературе по криптографии, это означает лишь, что этот некто правильно обращается со своим собственным закрытым ключом. Это не говорит, что вы можете всегда доверять сертификату, полученному от CA для какой-то определенной цели: оплатить небольшую сумму или подписать заказ на миллион долларов.

Кто дает CA полномочия проводить подобные авторизации? Кто делает их доверенными лицами? Многие сертифицирующие органы обходят вопрос о том, что им никто не давал власти проводить авторизацию путем выпуска идентификационных сертификатов. Каждый может назначать имена. Мы делаем это все время. Инстанция, подтверждающая сертификаты, идет на риск, если она использует идентификационные сертификаты таким образом, что они подразумевают некий вид авторизации. В основном сертификаты только защищают вас от тех, с кем продавцы PKI не захотели даже иметь дела.

И слово «полномочия» («authority») имеет несколько значений. CA могут обладать авторитетом для выпуска сертификатов, но обладают ли они властью в отношении их содержимого? Например, сертификат сервера SSL содержит два массива данных, представляющих потенциальный интерес для целей безопасности: имя держателя ключа (keyholder) (обычно имя корпорации) и имя сервера в системе DNS (Domain Name System, система доменных имен). Существуют органы, обладающие властью назначать имя в системе DNS, но ни один из сертифицирующих органов SSL, представленных в списках, которые можно видеть в популярных браузерах, таковым не является. Это означает, что имя DNS в сертификате не является авторитетно установленным. Есть органы, назначающие корпоративные имена. Такое имя должно регистрироваться, когда некто хочет получить лицензию на какой-либо бизнес. Однако ни один из CA SSL, опубликованных в популярных браузерах, не обладает полномочиями делать это. В добавление ко всему, если сервер получает сертификат SSL-сервера, он имеет разрешение работать по протоколу SSL. Кто дал полномочия сертификационному органу SSL контролировать подобные разрешения? И является ли контроль за этими разрешениями необходимым? Какой вред будет причинен, если несертифицированный сервер будет использовать кодирование? Никакого.

Некоторые центры сертификации, учитывая тот факт, что они не властны над содержимым сертификата, выделили структуру, называемую *бюро регистрации (Registration Authority, RA),* которая должна отвечать за содержание. Идея состоит в том, что RA ответственен за проверку информации, находящейся в сертификатах, а CA отвечает за их выдачу.

Модель RA+CA, безусловно, менее безопасна, чем система, при которой CA находится в составе полномочного органа (то есть RA). Модель RA+CA позволяет некоторому объекту (CA), который не является полномочным относительно содержания, подделать сертификат с соответствующим содержанием. Конечно, сертифицирующий орган подписывает контракт, в котором обещает так не делать, но это не исключает подобную возможность. Между тем, поскольку безопасность всей модели зависит от безопасности обеих ее частей и взаимодействия между ними (они должны каким-либо способом взаимодействовать), RA+CA менее безопасна, чем каждая из частей, безотносительно того, насколько силен CA и насколько хорош контракт, который с ним подписан. Конечно, модель, в которой CA находится

в составе органа-распорядителя (не со стороны продавца), препятствует некоторым способам ведения бизнеса продавцами в инфраструктуре PKI.

Другую проблему представляет защита закрытого ключа. Вспомним: чтобы вся система цифровых подписей работала, вы должны быть уверены, что ваш закрытый ключ известен только вам. Отлично, но как защитить его? Можно быть почти уверенным, что у вас нет безопасной вычислительной системы, с контролем физического доступа, экранированием TEMPEST, «пространством» сетевой безопасности и другими средствами защиты; вы храните свой закрытый ключ на обычном компьютере. Таким образом, он является объектом для вирусов и иных разрушительных программ. Даже если ваш закрытый ключ в безопасности на вашем компьютере, находится ли этот компьютер в закрытой комнате, снабженной системой видеонаблюдения, уверены ли вы, что никто, кроме вас, никогда не воспользуется им? Если он защищен паролем, то насколько трудно «вычислить» этот пароль? Если ваш ключ хранится на смарт-карте, насколько она устойчива к атакам? Если он содержится на действительно устойчивом к атакам устройстве, то не может ли инфицированный компьютер найти способ «убедить» это заслуживающее доверия устройство подписать то, что вы не хотели бы подписывать?

Здесь дело преимущественно касается термина *безотказность (nonrepudation)*. Подобно «доверенному» объекту, этот термин взят из академической литературы по криптографии. Здесь он используется в специфическом значении, описывая алгоритм создания цифровой подписи, который невозможно разрушить, так что никто не может подделать вашу подпись. Продавцы PKI замыкаются на этом термине и используют его в юридическом смысле, лоббируя законы, согласно которым, если кто-то использует свой закрытый ключ, то он в любом случае не может отказаться от своей подписи. Другими словами, в соответствии с некоторыми законами об использовании цифровой подписи (например, законами штатов Юта и Вашингтон), если ключ вашей подписи сертифицирован правомочным CA, вы отвечаете за все, что с помощью этого ключа делается. Им нет дела до того, кто сидел за клавиатурой компьютера или какой вирус создал эту подпись: вы ответственны по закону.

Считается, что работает следующий механизм: если вы подозреваете, что ваш ключ ненадежен, вы «отправляете» его в CRL. Все, что подписано этим ключом в дальнейшем, автоматически считается недействительным. Это кажется невероятным, но такая система изначально порочна. Боб хочет знать, что ключ Алисы вне подозрений, прежде чем примет ее подпись. Злоумышленник не собирается объявлять о том, что он взломал Алисин ключ. Таким образом, Алиса может понять, что ее ключ взломан, только когда она получит от Боба сообщение, демонстрирующее доказательства подделки ее подписи. В большинстве случаев это может произойти, только если Боб примет ее подпись.

Контраст с этим являет собой практика защиты кредитных карт. Согласно правилам, регулирующим заказы по почте и по телефону, если вы возражаете против какого-либо пункта в счете вашей кредитной карты, вы имеете право отвергнуть его — сказать, что вы не покупали этого, — и продавец должен доказать, что вы на самом деле это купили.

В компьютере существуют различные уязвимые места, которые должны проверяться. Проверка сертификата не имеет отношения к секретному ключу, он под¬

утверждает только открытый ключ. Но чтобы проверить сертификат, нам необходим один или несколько «корневых» открытых ключей: открытые ключи сертифицирующих органов. Если злоумышленник сумеет прибавить свой открытый ключ к этому списку, он может выпустить затем свои собственные сертификаты, и обращаться с ними будут точно так же, как и с законными. Они могут быть даже более чем законными в любом другом отношении, за исключением того, что они содержат открытый ключ злоумышленника вместо корректного открытого ключа.

Некоторые продавцы PKI объявляют, что их ключи находятся в *корневых сертификатах* и вполне безопасны. Такие сертификаты подписаны ими самими и не предполагают следующего уровня безопасности. Единственным решением будет проведение всех сертификационных проверок на компьютерной системе, защищенной как от вторжения враждебного кода, так и от физического вмешательства.

И наконец, как бюро сертификации идентифицируют владельца сертификата? Будет ли сертификат использоваться только как идентификатор или для другой специфической авторизации, но CA должен идентифицировать претендента перед вручением ему сертификата.

Некоторые кредитные организации подумывают о том, чтобы заняться сертификационным бизнесом. Помимо всего прочего они имеют обширную базу, содержащую данные о множестве людей, так что логично предположить, что они способны с легкостью установить подлинность личности. Если они хотят подтверждать подлинность по сети, они в состоянии сделать это, обеспечив совместный с субъектом доступ к секретной информации и безопасный канал, по которому конфиденциальную информацию можно передавать. SSL обеспечивает безопасный канал.

Помешать подобным кредитным организациям выполнить свою роль может то, что у них не получится сделать совместный доступ секретным. Другими словами, не существует безопасного идентификатора, который мог бы использоваться для запуска всего процесса, потому что такие организации участвуют в продаже имеющейся у них информации другим людям. Хуже того, поскольку кредитные бюро делают такое «доброе» дело, как сбор и продажа информации о людях, другие, кто предположительно имеет информацию о субъекте, возможно, подвергаются сильному давлению, чтобы они помогли найти те секретные данные о субъекте, которые еще не известны этому кредитному бюро.

Между тем, каким-либо образом идентифицировав претендента, как CA проверит, что претендент действительно контролировал закрытый ключ, соответствующий открытому ключу, подтвержденному сертификатом? Некоторые сертифицирующие органы даже не рассматривают эту часть процесса. Другие могут требовать, чтобы претендент подписал некоторый документ тут же на месте, в присутствии CA.

Сертификаты не напоминают некий магический эликсир безопасности, капля которого, добавленная к вашей системе, сделает ее безопасной. Сертификаты должны использоваться правильно, если вы хотите безопасности. Осуществляется ли эта практика с соблюдением твердых гарантий безвредности или это только ритуалы или имитация поведения кого-то еще? Многие случаи из подобной практики и даже некоторые стандарты при ближайшем рассмотрении оказывались скопированными с таких образцов, которые с самого начала не содержали реального ответа.

Как вычисляется время жизни ключа? Почему продавцы используют один год — потому, что это общепринято? Ключ имеет криптографическое время жизни. Он также имеет время жизни до воровства, являющееся функцией от уязвимости подсистемы, в которой он хранится, состояния ее физического и сетевого окружения, привлекательности ключа для взломщика и т. д. Исходя из этого можно вычислить вероятность потери ключа как функцию времени и эксплуатации. Делают ли продавцы эти вычисления? Какой порог вероятности выбирается, чтобы считать ключ недействительным?

Поддерживает ли продавец аннулирование ключа и сертификата? Списки отозванных сертификатов встроены в некоторые сертификационные стандарты, но многие реализации избежали этого. Но если CRL не используются, то как осуществляется упразднение? Если отмена действия поддерживается, как предотвратить использование ключа, когда стало известно, что он отозван? Может ли случиться так, что он будет отозван задним числом? Причем так, что владелец сертификата может отрицать, что именно он сделал эту подпись некоторое время назад? Если так, датируются ли подписи таким образом, чтобы каждый мог отличить действительные подписи от недействительных? Осуществляется ли контроль даты службами безопасности?

Какую длину имеют сгенерированные открытые ключи и почему выбрана эта длина? Поддерживает ли продавец более короткие и слабые ключи RSA только потому, что они быстрее, или более длинные ключи, поскольку кто-то где-то ему сказал, что они безопасны?

Требует ли правильное использование сертификатов некоторых действий от пользователя? Выполняет ли он эти действия? Например, когда вы устанавливаете соединение SSL с помощью своего браузера, присутствует видимая индикация, что работает протокол SSL и связь зашифрована. Но с кем вы говорите по безопасному каналу? До тех пор пока вы не прочтете сертификат, полученный вами, вы не знаете.

## PKI в Интернете

Большинство людей взаимодействуют с PKI единственно по поводу использования SSL. Технология SSL обеспечивает безопасность транзакций во Всемирной паутине, и некоторые продавцы PKI указывают на это, рекомендуя эту технологию для электронной коммерции. Это фальшивый аргумент; хотя никто не призывает людей, торгующих через Интернет, не использовать SSL.

SSL позволяет зашифровать операции с кредитными картами в Интернете, но не является источником безопасности для участников транзакции. Безопасность основывается на процедурах, выполняемых компанией кредитных карт; потребителю должно быть предоставлено право отвергнуть любой пункт, вызывающий сомнение, прежде чем он оплатит счет. SSL защищает пользователя от прослушивания, но не может защитить его ни от взломщика, если тот проникнет на веб-сайт и украдет файл, содержащий номера кредитных карт, ни от жуликоватого сотрудника, если тот следит за номерами кредитных карт в компании.

Предполагалось, что PKI будут обеспечивать идентификацию, но они не делают даже этого.

Пример первый: компания F-Secure (более формальное название Data Fellows) продает программное обеспечение через свой веб-сайт, находящийся по адресу www.datafellows.com. Если, намереваясь купить какой-либо продукт, вы щелкните в соответствующем месте, то будете перенаправлены на www.netsales.net, с которым устанавливается соединение по протоколу SSL. Владельцем сертификата SSL является «NetSales, Inc., Software Review LLC» в Канзасе. Главные представительства F-Secure расположены в Хельсинки и в Сан-Хосе. Согласно любым правилам PKI, никто не имеет права делать бизнес через этот сайт. Полученный сертификат принадлежит не той компании, которая на самом деле торгует программным обеспечением. Это в точности то, что мы называли атакой посредника, и это именно то, что PKI должны предотвращать.

Пример второй: я посетил www.palm.com, чтобы купить кое-что для моего PalmPilot. Когда я собирался проверить соединение, то был перенаправлен на https://palmorder.modusmedia.com/asp/store.asp. Сертификат зарегистрирован на имя Modus Media International; налицо скандальная попытка обмануть пользователей, которую я обнаружил, поскольку внимательно проверяю сертификаты SSL. Ну уж нет!

Кто-нибудь еще поднимет тревогу в таких случаях? Кто-нибудь не станет покупать продукт через Интернет только потому, что имя на сертификате не совпадает с именем на сайте? Кто-то, кроме меня, хоть что-нибудь заметит?

Сомневаюсь. Правда заключается в том, что сертификат, выданный VerySign, дает возможность осуществить атаку посредника, но никому нет до этого дела. Так или иначе я сделал покупку, поскольку безопасность обеспечивается правилами использования кредитных карт, а не протоколом SSL. Моя максимальная ответственность за украденную кредитную карту измеряется суммой 50 долларов, и я могу отвергнуть любую сделку, в которой нечестный торговец попытается обмануть меня. Так как все это используется, вместе с тем, что средний пользователь не беспокоится о проверке подлинности сертификатов, и не существует механизма аннулирования, то SSL — это просто (очень медленный) метод Диффи-Хеллмана обмена ключами. Цифровые сертификаты не обеспечивают реальной безопасности электронной коммерции; это полный обман.

# Глава 16. Уловки безопасности

Эта глава представляет собой коллекцию различных трюков и технических приемов, которые реально больше в безопасности не используются.

## Правительственный доступ к ключам

«Отлично, дела обстоят следующим образом: мы — правительство, и наше дело предупреждать преступность. Это нелегко, криминальным элементам свойственно идти окольными путями. Эти преступники, страшные преступники: наркодельцы, террористы, торговцы детской порнографией, фальшивомонетчики, — используют криптографию, чтобы защитить свои коммуникации. Мы беспокоимся, что наши перехваты, осуществляемые на законных основаниях, больше не эффективны; все эти страшные преступники скрываются. Мы хотим иметь возможность дешифровать сообщения каждого; может случиться, что он окажется криминальным элементом. Мы желаем, чтобы все вы сделали копии ваших ключей шифрования и послали их в полицию (или лицу, которому полиция доверяет), только на случай, если вы окажетесь преступником. И конечно, мы не доверим вам сделать это самостоятельно — поскольку мы собираемся выполнить это автоматически с помощью продуктов для шифрования, которые вы покупаете».

Надо признать, это недоброе описание позиции ФБР по вопросу о секретных ключах, но зато точное. С 1993 года администрация Клинтона и ФБР пытались вынудить американскую общественность принять идею, что нужно дать правительственным организациям доступ к частной информации. Они пытались привлечь корпорации к тому, чтобы они предусматривали эту возможность в своих продуктах, убедить пользователей, что эти меры предпринимаются в их высших интересах, и когда встретили сопротивление в Соединенных Штатах, продолжали оказывать давление на другие страны, вынуждая их принять такую же политику. Они даже угрожали объявить криптографию незаконной. Это очень спорный вопрос.

На первый взгляд кажется, что жалобы ФБР обоснованы. Преступники используют криптографию, чтобы скрыть доказательства, которые по закону могут быть использованы против них в суде: они зашифровывают компьютерные файлы, они шифруют телефонные переговоры и радиосвязь. Но положительные стороны использования криптографии намного перевешивают негативные факторы, и прежде всего то, что криптография чаще применяется, чтобы остановить преступников, чем в помощь им. Рон Ривест однажды сравнил криптографию с перчатками. Это

правда, производя перчатки, общество облегчает преступникам задачу скрыть свои отпечатки пальцев. Но никто никогда не предлагал объявить перчатки незаконными.

Существует множество названий для обозначения этой идеи. Первый термин, предложенный правительством, был *«депонирование ключей» («key escrow»),* так как мастер-ключ в «Клиппер-чипе» (Clipper chip)[1] предполагал, что сеансовый ключ будет «сдаваться на хранение» (hold «in escrow») и в дальнейшем передаваться органам охраны правопорядка. Если люди переставали покупать систему шифрования «о депонированием», название продукта изменяли, чтобы снова сделать его приемлемым. Сегодня используются термины *«восстановление ключей» («key recovery»), «шифрование с участием доверенной третьей стороны» («trusted third party encryption»), «исключительный доступ» («exceptional access»), «восстановление сообщений» («message recovery»)* и *«восстановление данных» («data recovery»).* Мне нравится GAK *(government access to keys, правительственный доступ к ключам).*

GAK-системы имеют *«черный* ход». Другими словами, они обеспечивают некую форму доступа к зашифрованным данным помимо нормального процесса дешифровки. В проекте «Клиппер-чип» предлагалось называть этот «черный ход» *областью доступа правоохранительных органов (Law Enforcement Access Field, LEAF).* (Первоначально она называлась областью использования органов охраны правопорядка, пока кто-то не указал на то, что это название неблагозвучно.)

«Черный ход» в GAK-системах предназначается для применения правительственными структурами (такими как полиция). Они работают различными способами: ранние GAK-системы оперировали с хранилищем закрытых ключей правительства США, более поздние самостоятельно определяют закрытые объекты. Другие системы используют программные агенты для «депонирования» или для восстановления ключей, что дает возможность узнать ключи частных зашифрованных сеансов связи или хранящихся файлов. Некоторые системы распределяют работу по восстановлению ключей между несколькими агентами. Имеются различные варианты, но все GAK-системы содержат два существенных элемента. Первый — механизм, внешний по отношению к первичной системе, посредством которого третья сторона может получить секретный доступ к открытому тексту, который был зашифрован. И второй — существование высокочувствительного метода секретного восстановления ключа (или коллекции ключей), который должен сохраняться в тайне в течение продолжительного времени. С точки зрения полиции, GAK-системы необходимы, чтобы предоставлять полиции своевременный доступ к открытому тексту, не сообщая об этом пользователям. Системы такого типа, по мнению администрации Клинтона и ФБР, решают проблему использования шифрования преступниками, делая доказательства их преступлений очевидными.

---

[1] Проект «Клиппер-чип» был объявлен администрацией президента США в 1994 году и должен был положить начало внедрению в США «Стандарта шифрования с депонированием ключа». Главный замысел проекта состоял в том, чтобы по решению суда предоставить правоохранительным органам беспрепятственный доступ к шифруемой с помощью «Клиппер-чипа» информации. Для этого в чипе используется алгоритм Skipjack с двумя ключами. Знание одного из ключей (мастер-ключа) достаточно для того, чтобы было возможно дешифровать любое сообщение, зашифрованное с помощью «Клиппер-чипа».

К несчастью, решение хуже самой проблемы. Восстановление данных производится с легкостью, поскольку это делается в высших интересах пользователя. Действия пользователей подобны автоматическому созданию резервной копии; они не должны помнить, что сделали копию. (Подождите, пока я не сделаю копию этого манускрипта.) Но GAK также связан с коммуникациями — телефонными разговорами и почтой, — которые не имеют отношения к копированию данных. Хранимые данные имеют огромную ценность: если вы теряете их, то нет способа их восстановить. Данные, передаваемые при соединении, не имеют ценности: если вы потеряли их, вы можете повторить передачу.

Способы осуществления GAK различны и достаточно сложны, поскольку он должен работать, несмотря на враждебность пользователей. Требования, предъявляемые ФБР к доступу, — скорость, секретность, полнота — вынуждают пользователей усиливать меры безопасности. Если я шифрую сообщение электронной почты, я должен доверять шифрованию — как на своей стороне, так и на стороне получателя. Наличие GAK означает, что я должен быть уверен в целой инфраструктуре, предназначенной для хранения ключей: шифрование, базы данных, полиция, народ. Цена построения такой инфраструктуры может быть огромной, в соответствии с уровнем возможного риска.

Этот риск неотделим от идеи использования GAK и не зависит от особенностей технологии. Все GAK-системы предполагают существование высокочувствительного и доступного ключа или коллекции ключей, которая должна сохраняться в секрете в течение продолжительного времени. Такие системы обязаны обеспечивать быстрый доступ к информации сотрудникам правоохранительных органов без ведома владельцев ключей. Эти основные требования делают задачу полного восстановления ключа трудной и дорогостоящей, возможно, слишком небезопасной и слишком дорогостоящей для многих приложений и многих пользователей.

Рассматривая различные варианты GAK, вы можете выбирать между более высокой ценой и более высоким риском. Хотя это может оказаться возможным — использовать отдельную GAK-систему относительно безопасным образом, для достижения этого результата пользователь заплатит неимоверную цену. С другой стороны, простые и недорогие GAK-системы могут создать угрозу для безопасности. Например, плохо реализованная система для восстановления ключей, обслуживаемая малоквалифицированным и низкооплачиваемым персоналом, с низким уровнем физической секретности и без надежного страхования, скорее всего, будет дешевле хорошо организованного центра. Но она также будет менее аккуратно обращаться с ключами.

Интересно, что безопасность и цена зависят также от конструкции системы. Например, представьте себе механизм, в котором сеансовые ключи отсылаются в центр по восстановлению, зашифрованные общедоступным открытым ключом, хранящимся в центре. Эта система относительно проста для конструирования и выполнения, но она является одной из худших с точки зрения безопасности. Все зависит от ключа системы восстановления, которым шифруются все остальные ключи. Если этот ключ взломан (или подделан), все ключи в системе могут быть взломаны. Конечно, некоторые коммерческие системы основываются на почти таком же механизме.

GAK-системы существенно менее безопасные, более дорогие и более сложные в эксплуатации, чем подобные им системы, не предусматривающие возможность восстановления ключей. Чтобы сделать их работающими, надо выдвинуть требование считать незаконными продукты, не включающие GAK. Более того, для этой схемы требуется построение безопасной инфраструктуры такого поразительного масштаба и сложности, создание которой превышает существующие сегодня возможности в этой области и может внести безусловно неприемлемый риск и дороговизну.

## Безопасность баз данных

Безопасность баз данных — более сложная вещь, чем обычно думают. В простейшем случае все легко: Алиса имеет доступ к персональной базе данных, Боб не имеет. В более трудных случаях все сложнее — Алиса имеет доступ к части базы данных, относящейся к медицинскому страхованию, а Боб имеет доступ к персональной базе данных по зарплате, — но коммерческие базы данных управляются с этим достаточно успешно. Действительно сложная задача — обеспечить анонимность при работе с базой данных, позволяя людям использовать итоговую информацию, — удивительно трудна.

Первый трудный случай. База данных может быть настроена таким образом, чтобы определенные пользователи могли видеть только определенные поля. Всем пользователям должно быть разрешено видеть ряд полей общего доступа (имя служащего, номер служащего), но только некоторым пользователям разрешается видеть специальные поля (информация о медицинском страховании, зарплата). Это все обычные проблемы компьютерной безопасности, разрешаемые с помощью протоколов идентификации и списков контроля доступа.

Намного более сложно иметь дело с ситуацией, когда у Алисы есть право делать запросы и смотреть статистические данные, но нет права на просмотр информации по каждому отдельному случаю. Это одна из проблем «вывода»; Алиса может вывести информацию об отдельных случаях, но делая запросы относительно групп.

Пример: Алиса запрашивает у базы данных итоговую информацию по определенным группам. Если она сможет получить ответ на запрос такого сорта — сообщить обо всех наркоманках, в возрасте от 35 до 45 лет, у которых один из родителей диабетик, имеющих определенный почтовый адрес, — скорее всего, она получит возможность выделить отдельные случаи.

Возможное решение этой проблемы — вычищать данные, прежде чем они попадут в чьи-нибудь руки. Данные о переписи населения США 1960 года, например, засекречены подобным образом. Для статистического анализа можно получить только одну запись на тысячу, и в этих записях удалены поля, содержащие имена, адреса и другая важная информация. Бюро переписи также применяет ряд других трюков: данные, имеющие экстремальное значение, подавляются, в систему добавляется шум. Эти методы защиты сложны, и, несмотря на это, достаточно тонкие атаки все же могут быть проведены. Если вы хотите получить сведения об одной богатой семье, живущей по соседству, может оказаться возможным вывести эти данные, если вы сделаете некоторые разумные предположения.

Другое допустимое решение — наложить ограничение на типы запросов, которые пользователь может делать к базе данных. Это также трудно сделать правильно. В одной известной исследовательской работе автор, вычисляя жалованье своего босса, основывался на вполне законных запросах к базе данных переписи 1970 года, несмотря на контроль, введенный специально для того, чтобы предупреждать подобные вещи. Информационная система Национального здравоохранения Новой Зеландии пыталась разрушить подобные виды атак, не выдавая информацию по группам, состоящим менее чем из 6 человек.

Атаки все равно остаются возможными. Алиса собирается узнать виды разрешенных запросов, и лучшее, что она может сделать, — сформировать некий математический подход для вывода нужной ей информации из информации, к которой она имеет доступ. И подобные вещи усугубляются, если Алисе разрешено добавлять данные в базу и удалять их. Если она хочет получить информацию относительно отдельного человека, она могла бы вставить пару сотен записей в базу и затем сделать общий запрос по людям, включая тех, которых она добавила, плюс ее цель. Так как ей известны все данные, которые она добавила, она может «вычислить» данные того, кто ей нужен. Целый ряд атак основывается на означенной идее.

В этой области проводились активные исследования в 80-х годах, но сейчас их меньше. (Новые правила конфиденциальности медицинской информации могут привести к их возрождению.) Хотя проблемы так и не решены.

# Стеганография

Стеганография[1] — это наука скрывать сообщения внутри сообщений. Геродот рассказывает о практике древних греков, когда секретное сообщение записывали с помощью татуировки, нанесенной на бритую голову посланца, а потом он снова отращивал волосы, перед тем как переправить это сообщение через вражескую территорию. (Длительность этой связи измерялась месяцами.) Использование невидимых чернил — более современная технология. Микрофотоснимки были изобретены немцами во время Первой мировой войны и оставались в моде в течение многих лет. Шпионы делали негативное изображение достаточно маленьким, чтобы можно было вырезать его и поместить между строчек книги. Шпион мог пронести такую книгу везде, не опасаясь, что кто-либо обнаружит микрофотографии, скрытые на одной из ее многочисленных страниц.

В компьютерном мире стеганография используется для скрытия секретных сообщений в графике, картинках, движущихся изображениях или в звуке. Отправитель скрывает сообщение, используя наименьшие значащие биты[2] файла одного из этих типов — качество слегка ухудшается, но если вы сделали все правильно, это с трудом можно заметить — получатель на другом конце извлекает его. Неко¬

[1] *Стеганография* в переводе с греческого — *тайнопись. — Примеч. перев.*

[2] Наиболее распространенным, но наименее стойким является метод замены наименьших значащих битов (LSB-метод). Он подстраивается под погрешность дискретизации, которая всегда существует в оцифрованных изображениях или аудиофайлах и видеофайлах. Модификация младших битов в большинстве случаев не вызывает значительной трансформации изображения и не обнаруживается визуально. — *Примеч. перев.*

торые коммерческие и свободно распространяемые программы предлагают стеганографию либо отдельно, либо в составе целого пакета средств обеспечения безопасности связи.

Стеганография обеспечивает степень секретности более высокую, чем криптография. Если Алиса захочет послать Бобу секретное сообщение, она может использовать одну из популярных программ шифрования для электронной почты. Однако заинтересованное лицо может перехватить сообщение, и хотя оно не сумеет прочитать его, но будет знать о самом факте отправки сообщения. Стеганография позволяет Алисе сообщаться с Бобом секретно; она может скрыть свое письмо в GIF-файле, содержащем изображение пары жирафов. Перехватчица не поймет, что Алиса послала Бобу секретное сообщение. Для обеспечения еще более надежной защиты Алиса может зашифровать сообщение, прежде чем скрыть его.

Чем дальше, тем лучше. Но в действительности системы работают не так. Наша перехватчица не глупа; как только она увидит картинку с жирафами, у нее могут возникнуть подозрения. Почему бы это Алисе посылать Бобу картинку с двумя жирафами? Боб коллекционирует жирафов? Или он художник-график? Посылали ли Алиса и Боб одно и то же изображение жирафов друг другу в течение последних недель? Упоминали ли об этой картинке в другой корреспонденции?

Самого по себе применения стеганографии недостаточно. Алиса и Боб должны скрыть факт, что они передавали что-либо, кроме безвредных фотографий. Это будет работать только в том случае, если стеганография станет использовать существующие образцы связи (communications patterns). Я в жизни не посылал и не получал GIF-изображений. Если кто-то неожиданно пошлет мне одно, это может навести дотошного исследователя на мысль, что в нем скрыто стеганографическое сообщение. Если Алиса и Боб регулярно обмениваются подобными файлами, перехватчик не сможет узнать, какое сообщение содержит в себе — если какое-либо содержит — скрытую информацию. Если Алиса и Боб изменят свои образцы связи для скрытых сообщений, система перестанет работать. Перехватчица их «вычислит».

Это важно. Стеганографию иногда рекомендуют использовать для секретной связи при репрессивных режимах, когда простой факт посылки зашифрованного сообщения может рассматриваться как подрывная деятельность. Это плохой совет. Модель, в которой постоянно присутствует угроза, предполагает, что вы под подозрением и должны выглядеть невинным в случае проверки. Это трудно. Вы можете применить программу стеганографии, которая доступна и перехватчице ваших сообщений. Она будет иметь копию этой программы. Она будет настороже, предвидя возможность появления стеганографических сообщений. Не используйте простое изображение, которое было в программе, когда вы установили ее: перехватчица быстро распознает его. Не используйте одно и то же изображение снова и снова: перехватчица увидит различия, указывающие на то, что это скрытое сообщение. Не используйте картинки, загруженные из Сети: перехватчица легко может сравнить картинку, которую вы послали, с исходной, которую вы загрузили. (Вы можете предположить, что она следит за загрузкой или осуществит поиск в Сети, чтобы найти такое же изображение.) И лучше, если вы придумаете правдоподобную историю о том, почему вы «гоняете» жирафов туда и обратно. И эта история должна быть придумана до того, как вы начнете посылать стеганографические сообщения, или вы ничего реально не выиграете.

Программы стеганографии существуют, чтобы скрывать файлы на вашем жестком диске. Это может работать, но вам по-прежнему нужна жизнеспособная история. Таким образом, есть определенные достижения по сравнению с простым шифрованием — по крайней мере, в свободных странах вы сможете доказать, что полиция не имеет реальных оснований для обвинения, — но вы должны тщательно все обдумать.

## Скрытые каналы

Одна из проблем, связанных с применением стеганографии, — ширина полосы пропускания. Легко скрыть несколько битов информации; спрятать целое сообщение электронной почты намного труднее. Рассмотрим пример совершенно разумного использования стеганографического канала передачи данных: Алиса и Боб должны обсудить, является ли некое отдельное действие «безопасным» или «угрожающим». Это один бит информации. Они регулярно обмениваются рецептами по электронной почте и договорились, что ключевая фраза «продублируй рецепт» будет индикатором сообщения. Если в послании сказано, что рецепт может быть продублирован, действие безопасно. Если же в нем говорится, что рецепт не может быть продублирован, соответствующее действие опасно. Любой рецепт без ключевой фразы не содержит скрытого сообщения.

Этот вид систем работает, поскольку секретное послание много-много меньше, чем скрывающее его сообщение, и в общем случае называется *скрытым каналом (subliminal channel)* (похоже на тайный канал, описанный в главе 8). Скрытые каналы так же стары, как компьютеры, и всегда использовались недобросовестными программистами для «скачивания» информации без согласия пользователей. Представьте, что вы программист и делаете отчет по клиентам банка, и вы хотите запустить свои руки в картотеку индивидуальных номеров (PINs). Вас не уполномочили проверять реальные данные, но доверили вам написать код для получения отчета по базе, содержащей PINs. И вы можете посмотреть отчеты, которые были сделаны раньше. Программа создания отчета добавляет пробелы после данных каждого клиента, от 0 до 9, в соответствии с одной цифрой его PIN. Пусть теперь построитель отчета использует первую цифру в первый день, вторую цифру во второй день, и так далее, пока цикл не будет завершен и мы не возвратимся к первой цифре. Вот именно. Если программист сможет приложить руку к созданию электронного отчета в течение четырех дней, он справится с восстановлением всех индивидуальных номеров. (Действительно, он имеет четыре возможных варианта для каждого номера, в зависимости от того, какая цифра использовалась построителем отчета. Легко понять, что к чему.) Ни один из тех, кто будет смотреть отчеты, не увидит в них ничего злонамеренного, и пока они не проверят код, используемый для создания отчета (а как часто это случается?), никто не узнает, что индивидуальные номера раскрыты.

Есть история о солдате, которому не разрешали говорить, где он служит. У него не было среднего инициала, и он послал серию писем своей подруге, используя в подписи различные средние инициалы; таким образом он дал знать, где находится.

Теперь, когда вы имеете представление об общей идее, вы можете подумать обо всех возможных способах внедрения скрытых каналов в документы: выборе шрифтов и размерах шрифтов, размещении данных и графики на странице, использовании различных синонимов в тексте и т. д. Многие протоколы шифрования позволяют воспользоваться выбором параметров в целях создания скрытого канала: выбором случайных битов для дополнения или неиспользованных битов полей. До тех пор пока вы не слишком жадничаете и согласны черпать информацию чайной ложечкой, несложно организовать скрытый канал в системе.

Вы можете организовать утечку всего что угодно. Индивидуальные номера — хороший пример. Другой пример — ключи шифрования. Создание устройства для шифрования, в котором информация о ключах утекает по скрытому каналу, — замечательный способ атаковать кого-нибудь.

Скрытые каналы, внедренные недобросовестными программистами, обнаруживались во всех видах программного обеспечения спустя какое-то время. Разведывательные организации, подобные NSA, долгое время подозревались во внедрении скрытых каналов, по которым идет утечка информации о ключах криптографического оборудования, проданного иностранным правительствам. Недавний скандал, в котором фигурировала шведская компания Crypto AG, подтверждает это. Побочные каналы, обсуждавшиеся в контексте главы 14, где речь шла об аппаратных средствах сопротивления вторжению, могут рассматриваться как действующие скрытые каналы.

Заметим, что для скрытых каналов существует та же проблема, что и для стеганографии — каждый, кто удосужится проверить программное обеспечение, может обнаружить скрытый канал. Но внедренные в сложное программное обеспечение, а еще лучше в аппаратное обеспечение, они могут оставаться незамеченными в течение долгого времени.

## Цифровые водяные знаки

Мы говорили об интеллектуальной собственности в главе 3. К слову сказать, компании, подобные компании Диснея, собираются торговать «в розницу» своей интеллектуальной собственностью — музыкой, видеоклипами, фотоснимками, чем угодно — в цифровой форме. Они не хотели бы, чтобы люди копировали *Русалочку* ( *The Little Mermaid)* и свободно распространяли ее через Интернет. Они не хотели бы, чтобы люди крали части различных изображений — даже простейшего изображения Микки-Мауса — и использовали их без уплаты определенного вознаграждения автору. Они хотят осуществлять контроль над своей собственностью.

Цифровые водяные знаки — один из способов достижения этой цели. Их можно считать скрытым каналом или одним из приложений стеганографии. Идея состоит в том, чтобы вложить секретную информацию в материал, и тогда нетрудно будет определить, кто его законный владелец. Это похоже на бумажные водяные знаки: бумага с водяными знаками распространяется везде и передается от человека к человеку, но любой может посмотреть ее на свет и увидеть водяные знаки.

В действительности есть два различных термина. Создание *водяных знаков (watermarking)* позволяет идентифицировать неизменяемую информацию, а ис¬

пользование *отпечатков пальцев (fingerprinting)* дает возможность идентифицировать конкретного покупателя.

Например, водяной знак *Русалочки* будет говорить нам нечто вроде: «Собственность Диснея», тогда как цифровой отпечаток пальца на этом фильме скажет нам следующее: «Куплено Алисой 1 января 2001 года».

Цифровые водяные знаки и отпечатки пальцев всем хороши, кроме одного. Скопируйте бумагу, и водяные знаки пропадут. Скопируйте цифровой файл, и водяные знаки перейдут на копию. Может быть, мы не в силах предотвратить копирование, считают в компании Диснея, но мы по крайней мере можем указать пальцем на каждого, кто сделал копию. И я видел водяные знаки на множестве вещей: графике, изображениях, аудио, видео... даже в данных биржевых телеграфных аппаратов и компьютерных программ.

Таким образом, в зависимости от того, какие данные вы заложили в водяные знаки, вам предоставлена одна из двух возможностей. Во-первых, можно определить, кто обладает авторскими правами. Во-вторых, если каждая проданная копия *Русалочки* будет снабжена водяными знаками, указывающими на имя и адрес того, кто ее купил, станет возможным установить также и покупателя: тогда, если копия появится в Интернете, компания Диснея сможет найти виновного.

Замечательная идея, только не работает.

Проблема в том, что для того, чтобы Дисней смог найти внедренные в копию *Русалочки* водяные знаки, должна существовать возможность их обнаружения. Но если Дисней может их найти, пират может сделать то же самое. Компании, продающие подобные вещи, постараются объяснить вам, что их схема создания водяных знаков не позволяет удалить эти знаки по той или иной технологической причине.

Но это неправда. Может быть, так же как в случае со скрытым каналом, невозможно найти хорошие водяные знаки, если вы не знаете точно, куда смотреть. Но в отличие от скрытого канала, механизм обнаружения в данном случае в конце концов станет известным. Либо информация просочится в сообщество хакеров, как это обычно бывает, либо она станет доступной общественности при первом же случае судебного разбирательства. Механизмы создания водяных знаков в итоге все равно становятся известными, и когда это происходит, можно осуществить обратный процесс и удалить знаки с изображения.

Это может оказаться нелегким делом. Разные хитроумные трюки способны затруднить этот процесс, но все же это не является невозможным. И сообразительный хакер вполне в состоянии написать программу, автоматически удаляющую водяные знаки, как только он поймет, как это работает.

Другим слабым местом водяных знаков является то, что они не решают стоящей перед ними задачи. Создание водяных знаков должно позволить компании указать на ее цифровую собственность, которая должна оставаться неприкосновенной и сказать: «это мое». Осуществить это достаточно сложно, и водяные знаки не могут воспрепятствовать изменению цифровых данных, являющихся чужим имуществом. Также водяные знаки не обязательно укажут точно на нарушителя авторского права. Представьте, что каждая копия *Русалочки* снабжена водяными знаками с информацией о покупателе. Но как продавец установит личность покупателя? Это возможно только в том случае, если имеются защищенные от подделки документы. Однако нет способа воспрепятствовать злоумышленнику нанять за

10 долларов бездомного пьяницу, чтобы тот зашел в видеосалон и купил для него фильм. Таким образом, он получит копию с внедренными водяными знаками, указывающими на того, кто, скорее всего, не слишком беспокоится, что Дисней его опознает, и с кого нечего взять, если Дисней надумает предъявить к нему претензии.

Наличие водяных знаков может вызвать чувство вины у бабушки, когда она копирует *Русалочку* для своих внуков. Но нельзя помешать тайваньским пиратам стереть водяные знаки и выбросить на черный рынок полмиллиона копий фильма. Водяные знаки также не могут воспрепятствовать кому бы то ни было использовать подставное лицо, чтобы законным образом приобрести копию и ни о чем больше не беспокоиться.

## Защита от копирования

Эту проблему легко описать, но гораздо труднее решить. Компании, производящие программное обеспечение, хотят, чтобы люди покупали их продукцию; их приводит в бешенство, когда кто-нибудь делает копии коммерческой программы, которая стоит сотни долларов, и раздает ее своим друзьям. (Правда, теперь им это даже нравится в каком-то смысле. Они понимают, что если некто, вовсе не собиравшийся приобрести законную копию, получит возможность попользоваться программой и придет от этого в восторг, всяко он или его босс в итоге приобретут эту программу, а не продукт конкурентов. При распространении программы WordPerfect большую роль сыграла эта схема, которая во многом способствовала росту ее популярности.) Сказанное в первую очередь относится к распространению компьютерных игр и к распространению программ в тех странах, где слабо обеспечивается защита авторских прав: в этих случаях большинство пользователей скорее предпочтут приобретение пиратской копии покупке программы законным образом. (С подобной проблемой сталкиваются продавцы книг, кинофильмов, видео и т. д. — им не выгодно, чтобы их продукцию копировали.)

Существует много решений — встроенный в программное обеспечение код, который препятствует копированию, код, который обращается к некопируемым фрагментам оригинального диска, аппаратные средства, которые приводят в действие программные механизмы, — и я не буду здесь вдаваться во все детали. Они все обладают одним существенным недостатком: дело в том, что практически невозможно защитить от копирования программное обеспечение многоцелевого компьютера.

Если мы имеем дело с компьютерным пользователем Джо Середнячком, то любая система защиты от копирования работает. Он сможет скопировать обычные файлы, следуя указаниям, но ему не придет в голову взломать сложную схему зашиты от копирования. Если же мы имеем дело с Джейн Хакер, никакая система защиты от копирования не сработает.

Дело в том, что Джейн управляет работой своего компьютера. Она может запустить программы обнаружения ошибок, восстановить исходный код, проанализировать программы защиты. Если она достаточно сообразительна, она сможет разобраться в программном коде и изменить коды программ защиты. Производители программного обеспечения не сумеют сделать ничего, чтобы остановить ее: все,

что они могут сделать, это усложнить ее задачу. Но это только послужит вызовом для Джейн.

Существует множество таких Джейн, которые выбрали своим хобби взломы подобных систем. Они «болтаются» по Сети, продавая нелегальное программное обеспечение. Другие делают это для получения прибыли. Они работают в Китае, Тайване и еще где угодно, взламывая системы защиты от копирования и перепродавая программное обеспечение в десять раз дешевле розничной цены. Они способны разрушить даже самый сложный механизм защиты. Урок, который мы должны вынести из того, что эти люди существуют, — это понимание, что любая схема защиты от копирования может быть взломана.

*Защитная заглушка* — это последнее слово техники в искусстве создания систем защиты от копирования. Она представляет собой аппаратное средство, подсоединенное к компьютеру, обычно к параллельному порту. (Конфликты с другими механизмами, подсоединенными к параллельному порту, или с другими заглушками проявляются лишь в редких случаях.) В процессе выполнения программ часто используется обращение к заглушкам; например, через каждую тысячу нажатий на клавиатуру или щелчков кнопкой мыши; или когда пользователь пытается сохранить данные; или каждый раз, когда он выбирает вид оружия в компьютерной игре. Если заглушка не отвечает на запрос или отвечает некорректно, программное обеспечение прерывает свою работу. Или, что более эффективно, продолжает работать, но при этом выдает неправильные результаты. (В версии Autodesk's 3D Studio 1992 года заглушка использовалась для создания в памяти таблицы, которая требуется для правильного отображения трехмерной геометрии. Удаление заглушки приводило к тому, что программа работала неправильно в течение нескольких часов: сначала это происходило почти незаметно, но чем дальше, тем хуже. Autodesk был вынужден в спешном порядке давать ответы на множество запросов незарегистрированных пользователей, жалующихся на странную ошибку в этой версии.)

Запросы к заглушкам всегда зашифрованы, и сами они защищены от аппаратного перепрограммирования множеством различных хитростей. Тем не менее программы, использующие заглушки, по сей день регулярно взламываются без применения ухищрений, направленных на преодоление криптографии или системы сопротивления вторжению.

Как это происходит? Хакеры вне зависимости от того, устояла заглушка или нет, внедряются в код и удаляют из него все обращения к ней. Это кропотливая работа: хакеры должны просмотреть весь код строку за строкой, функцию за функцией, вызов за вызовом. Им необходимо подключить логический анализатор к заглушке и проследить адреса обращений к ней. Сложная программа может содержать десятки мегабайтов кода. Но вспомним, что было сказано в главе 2, и мои рассуждения о том, чем Интернет отличается от реального мира: лишь первый взломщик должен добиться успеха, взломанной версией программы смогут пользоваться все остальные.

Несмотря на успехи пиратов, компании-производители не прекращают попыток защитить свою продукцию от копирования. Версия Quake 1996 была выпущена на зашифрованном компакт-диске: позволено было использовать ее практически бесплатно, если позвонить в компанию и приобрести пароль. Однако и она была взломана, так же как и все другие популярные программы, имевшие защиту от копирования.

Взломанные программы называются *warez,* и каждый может самостоятельно собрать целую их коллекцию, блуждая по Интернету. Вы не найдете соответствующие руководства, но в вашем распоряжении есть множество полезных книг.

Производители средств защиты от копирования стараются сохранить свои позиции, ссылаясь на возможности новых технологий в этой области. Они уверяют, что серийный номер микропроцессора является гарантией того, что законно приобретенная программа будет установлена на одном-единственном компьютере. Они рассуждают о возможности применения кодирования с использованием особенностей конкретной материнской платы. Все это неправда. Это может удержать Джо Середнячка от распространения программ, имеющихся в его распоряжении, но не остановит Джейн Хакер от того, чтобы взломать программу и предоставить ее версию для всеобщего пользования.

Двоякая природа рисков в этом случае точно такая же, как и рассматривавшаяся в связи с проблемой водяных знаков. Посмотрите, что происходит в видеоиндустрии: пиратство было гораздо менее распространено, когда видеомагнитофоны были еще в диковинку. Тому есть две причины. Во-первых, изящная защита от копирования не позволяла Джо Середнячку использовать для этих целей ни один из существовавших видеомагнитофонов. Во-вторых, розничная цена видеозаписей была настолько низка, что все ухищрения Джейн Хакер были экономически не оправданны.

Что действительно интересно в связи с проблемой защиты от копирования и пиратства, так это идея о том, что проблема на самом деле не существует. Не столь важно защитить от копирования коммерческий продукт. В условиях конкуренции ключевым моментом является распространение продукта на рынке. Многие компании руководствуются принципом: пираты не причинят нам ущерба, если наша продукция не пользуется спросом. Это вроде того, как если кто-то принимает наши телепередачи вне нашей сети. Почти все те, кто крадет наши программы, не в состоянии заплатить за них. Однако когда эти пираты окажутся перед выбором, они будут покупать нашу продукцию, а не продукцию конкурентов. Пиратство оказывается неожиданным средством рекламы.

Компания Microsoft имела в виду именно это, когда переводила свои программы на китайский язык и распространяла их в этой стране. Было очевидно, что программы будут взламываться, но потери будут составлять меньше одной десятой со всех продаж. Часто цитировали слова одного из сотрудников Microsoft, Стивена Бальмера: «Готовность к тому, что ваши программы будут взламывать, означает, что вы понимаете — это будут ваши программы, а не конкурентов. Важно, чтобы в развивающихся странах на рынке были широко распространены краденые программы». Когда Китай войдет в число свободных стран, он будет ориентироваться на продукцию Microsoft. До тех пор Microsoft ничего не теряет. В этом состоит стратегия бизнеса[1].

---

[1] В настоящее время (2002 год) китайские власти всерьез озабочены монополией Microsoft на национальном рынке офисного программного обеспечения. Около двух десятков компаний и вузов представили совместную разработку «Янфань», которая проходит тестирование в госорганизациях Китая. Предполагается, что альтернатива Windows будет иметь уровень устойчивости и функциональности Windows 98 и совместимость с Microsoft Office. — *Примеч. ред.*

# Уничтожение информации

Очень часто возникает необходимость уничтожения информации. Если вы хотите удалить секретный файл со своего компьютера, вам хочется быть уверенными в том, что никто впоследствии не сможет восстановить его. Если вы используете секретный ключ шифрования связи — телефонных переговоров, например, — вам хотелось бы, чтобы ключ был уничтожен по окончании разговора и никто не смог бы воспользоваться им.

Уничтожение информации оказывается делом гораздо более сложным, чем это можно себе представить.

Когда вы удаляете файл с магнитного диска (жесткого, гибкого диска), данные в действительности не уничтожаются. (Поэтому и возможно их восстановление с помощью утилиты unerase.) Файл попросту помечается как «удаленный», и впоследствии соответствующие биты перезаписываются новыми данными. Единственный способ полностью удалить файл с магнитного диска состоит в том, чтобы записать на его место другой файл. Некоторые утилиты удаления именно это и делают.

Менее известно, что существуют технологии, позволяющие восстановить удаленные данные, даже если на их место были записаны новые. Я не буду вдаваться в премудрости этой науки, но вы можете представить, что перезапись некоего фрагмента является попросту записью в его начало. Некоторые данные, расположенные ниже, сохраняются. И когда вы вновь производите перезапись, сохраняются фрагменты двух предыдущих записей, и т. д. Технология, известная под названием «магнитная микроскопия», позволяет восстановить данные после многократной перезаписи. Сколь много может быть перезаписей, точно не известно, некоторые утверждают, что до десяти.

Эти «микроскопы» достаточно дороги (хотя любительские версии дешевеют), и проверки такого рода доступны только государственным структурам. Если у вас возникает беспокойство в связи с интересом государственных служб, единственным действенным способом уничтожения информации на магнитном диске будет стереть его в порошок.

Уничтожить данные, оказавшиеся в аппаратуре, нелегко. Как SRAM, так и DRAM сохраняют некоторые следы данных после отключения питания. Биты, содержащиеся в RAM, могут быть восстановлены путем определения изменений содержимого ячейки памяти. Изменением температуры чипа и подаваемого напряжения можно достичь восстановления удаленных данных. Существует множество физических методов, которые применимы для этих целей.

Оборудование для военной криптографии в США построено таким образом, чтобы уничтожать, или *«обнулять»,* ключи в случае вторжения. Это нелегкая задача, и тому есть две причины: удалить данные непросто, а кроме того, нужно еще успеть сделать это вовремя. Были разработаны специальные датчики, сигнализировавшие о попытках проникновения внутрь оборудования. Они реагируют на изменение различных параметров: напряжения и силы тока, освещенности и температуры. Однако если нападающий знает, что представляют собой эти датчики, он в состоянии их обойти. (Он может работать при таком освещении, к длине волны которого датчик нечувствителен, или может медленно изменять температуру,

чтобы обмануть его, а также использовать множество других приемов.) Опять-таки, это проблема в первую очередь государственных структур, и она очень сложна.

Одна из основных трудностей состоит в том, что устройство должно обеспечивать полную сохранность ключей в нормальных условиях и полностью удалять их, не оставляя следов, в чрезвычайных ситуациях. Если используется хорошая технология обеспечения сохранности ключей, неизбежно возникают трудности С их уничтожением. Решить сразу обе противоречащие друг другу задачи весьма непросто.

Коммерческие системы сталкиваются с этой проблемой в тех случаях, когда требуется, чтобы владелец устройства, например смарт-карты или приемника передач платного телевидения, не мог добраться до его секретов. Я уже говорил о системах сопротивления вторжению и способах их преодоления. Техника «обнуления» позволяет обезопаситься от нападений этого рода. Однако существуют способы борьбы и с «обнулением». В основном коммерческие системы не приобретают права на использование «обнуления» (мне известно только одно устройство коммерческого назначения, имеющее государственный сертификат «обнуления» FIPS 140-3), поскольку стоит это весьма дорого.

# Глава 17. Человеческий фактор

Обеспечить компьютерную безопасность трудно (может быть, даже невозможно), однако представьте на минуту, что нам это удалось сделать. Где необходимо, применяется мощная криптография, протоколы безопасности безупречно выполняют свои функции. В нашем распоряжении имеются как надежное оборудование, так и надежное программное обеспечение. Даже сеть, в которой мы работаем, совершенно безопасна. Чудесно!

К несчастью, этого еще недостаточно. Сделать что-либо полезное эта замечательная система может лишь при участии пользователей. И это взаимодействие человека с компьютером таит в себе наибольшую угрозу из всех существующих. Люди часто оказываются самым слабым звеном в системе мер безопасности, и именно они постоянно являются причиной неэффективности последних.

Когда я начинал давать консультации различным компаниям по вопросам криптографии, я обещал потенциальным клиентам, что смогу более или менее надежно защитить их данные, однако использование этих данных людьми будет представлять постоянную угрозу безопасности. С годами я стал более циничен и говорю будущим клиентам, что в отношении безопасности математический аппарат безупречен, компьютеры же уязвимы, сети вообще паршивы, а люди просто отвратительны. Я изучил множество вопросов, связанных с обеспечением безопасности компьютеров и сетей, и могу утверждать, что не существует решения проблемы человеческого фактора. Обезопасить что бы то ни было от воздействия человека вообще очень трудно.

Люди не понимают, что такое компьютер. Он представляется им загадочным «черным ящиком», в котором что-то происходит. Они доверяют его сообщениям и хотят лишь одного — чтобы их работа делалась.

Люди не понимают, что такое опасность, может быть, за исключением разве только случаев явной угрозы. Они запирают двери и задвигают шпингалеты на окнах. Идя по темной аллее, они стараются убедиться в том, что их никто не преследует. Но они не осознают скрытую опасность и не задумываются о том, что в сверток может быть заложена бомба или что продавец в уютном ночном магазинчике может продавать на сторону номера кредитных карт. И почему, собственно, они должны беспокоиться? Такое ведь почти никогда не случается.

Средства компьютерной безопасности работают в цифровом мире. Перевести информацию в царство цифр непросто, а сохранить ее там попросту невозможно.

Помните «безбумажный офис» прошлого года?[1] Информация никогда не остается в компьютере и постоянно переносится на бумагу. С точки зрения взломщика, информация, хранящаяся в бумажных папках, ничуть не хуже той, что содержится в компьютерных папках[2]. Бумаги, выброшенные в корзину, часто представляют большую ценность, нежели содержащиеся в компьютере сведения. Их легче похитить и труднее пропустить что-либо важное. Компания, в которой тщательно шифруются все данные, но оставляются незапертыми кабинеты или не уничтожаются выброшенные в корзину бумаги, открывает себя для нападений.

Я намерен рассмотреть шесть аспектов проблемы человеческого фактора.

- Как люди воспринимают опасность.
- Как люди относятся к редко происходящим событиям.
- Как люди доверяют компьютерам, и почему это может быть столь опасно.
- Почему бесполезно просить людей принимать разумные меры безопасности.
- Какую опасность представляют внутренние враги.
- Что такое манипулирование людьми, и почему нападающим бывает легко получить секретные сведения.

Это все выглядит не слишком красиво.

# Риск

Люди не умеют анализировать риски. Они не понимают, насколько это плохо, когда их система уязвима для нападений. В случае нападения они не приходят к обоснованному заключению о том, на что это похоже. Они не способны рассмотреть проблему безопасности и принять разумное решение о том, что же следует делать в этой ситуации.

Проблема состоит не только в недостатке информации, часто опасность оценивают совершенно неправильно, располагая при этом достаточными сведениями.

[1] Билл Гейтс, «Бизнес со скоростью мысли»: «Полностью электронная рабочая среда обычно называется *"безбумажным офисом"* — термин этот существует по крайней мере с 1973 года. Тогда это было мечтой. Не будет больше кип бумаги, в которых невозможно найти нужный документ. Не будет груд книг и отчетов, в которых приходится копаться в поисках маркетинговой информации или сведений о продажах. Не будет неправильно адресованных форм, потерянных счетов, многократного ввода одних и тех же данных, отсутствующих подписей и проволочек, вызванных недостающими документами. В наши дни все необходимое для реализации этой мечты имеется. Графические компьютерные среды и усовершенствованные аналитические инструменты значительно упрощают интеграцию данных различных типов. Мощные, объединенные в сети персональные компьютеры стали повсеместным атрибутом офисной обстановки. Интернет соединяет между собой ПК, разбросанные по всему миру. Тем не менее потребление бумага продолжает удваиваться каждые четыре года, 95% всей информации в Соединенных Штатах остается на бумаге, а в электронном виде хранится лишь 1%. Объем бумаг растет быстрее, чем электронная технология успевает их заменять!» Наиболее цитируемая книга по теме — «На пути к безбумажным информационным системам» (Lancaster F. W. Toward Paperless Information Systems. N. Y.: Academic Press. 1978). А наибольшая «головная боль» — название этой главы. Один из разделов книги Поля Страсмана «Информация в век электроники», посвященный безбумажному офису, также называется «Человеческий фактор». Так что не принципиально, на какой прошлый год ссылается Брюс Шнайер. — *Примеч. ред.*

[2] Игра слов: «paper files» и «computer files». — *Примеч. перев.*

Изучение этого явления показывает, что чаще всего люди неправильно оценивают риск землетрясений, авиа- и автокатастроф, пищевых отравлений, несчастных случаев при прыжках с парашютом и т. д. и т. п. Переоценивают опасность в тех случаях, когда (1) невозможно повлиять на развитие событий (например, возможность отравиться в ресторане) и (2) средства массовой информации нагнетают обстановку (например, возможность стать жертвой террористического акта). Недооцениваются опасность обыденных вещей и риск в повседневных ситуациях (например, возможность упасть с лестницы или оказаться под колесами автомобиля). Конечно, недостаток информации усугубляет проблему.

Всякое событие имеет большую или меньшую вероятность. Существует определенная вероятность, что криптография, меры компьютерной безопасности окажутся действенными, что оценка опасности будет правильной. Опасность имеет свою вероятность, так же как и безопасность.

Чтобы прояснить понятие вероятности, сыграем с Алисой в нехитрую игру. Будем подбрасывать монетку, и если выпадет орел, то в выигрыше Алиса, если решка — выигрыш наш. Но сперва мы попросим показать нам монету, так как хотим убедиться в том, что она «правильная»[1]. Пожалуйста, говорит Алиса, можете рассмотреть ее со всех сторон.

Мы бросаем монету, и выпадает решка. Это единичное событие не несет никакой другой информации, кроме той, что по крайней мере одна из сторон монеты выглядит таким образом. Итак, мы бросаем монету десять раз, и в шести случаях выпадает орел. Означает ли это, что монета «неправильная»? Возможно. Алиса спешит заметить, что если бросать «правильную» монету по десять раз, то в 38% случаев шесть раз выпадет орел. Это означает, что если взять сотню «правильных» монет и бросать каждую из них по десять раз, то 38 монет из ста шесть раз упадут орлом вверх. Трудно заподозрить здесь жульничество.

Далее, мы бросаем монету сто раз, и в шестидесяти случаях выпадает орел. Алиса напоминает нам, что если подбрасывать «правильную» монету сериями по сто раз, в 2,3% случаев она может шестьдесят раз упасть орлом вверх. Монета все еще может считаться «правильной».

Предположим, мы бросили монету миллион раз. Наиболее правдоподобный результат должен быть таков: 500 000 раз выпал орел и столько же — решка. Однако в нашем случае он оказался иным: 600 000 раз выпал орел, а решка — 400 000. Несмотря на уверения Алисы в том, что такая возможность все же существует, хоть и одна на десять миллиардов, мы предпочитаем думать, что монета утяжелена с одной стороны. Хотя наша уверенность основана лишь на оценке вероятности, поверить в то, что монета «правильная», просто невозможно.

Теперь мы, скорее всего, откажемся использовать эту монету в игре с Алисой, хоть она и протестует, утверждая, что монета «правильная». Это будет благоразумное решение, несмотря на то что формально Алиса права. Невозможно доказать, что монета утяжелена, не разрезав ее на части и не взвесив их по отдельности. Все, что мы можем сделать в этой ситуации, это продолжать собирать все более убедительные свидетельства «неправильности» монеты.

---

[1] fair coin (мат.) — симметричная монета, правильная монета. — *Примеч. перев.*

Очень часто наша уверенность основана на повторяемости событий. Мы уверены в том, что Солнце взойдет на востоке потому, что так происходит каждое утро на протяжении миллиардов лет. Это обстоятельство, без сомнения, свидетельствует о том, что вероятность иного развития событий бесконечно мала. (Сейчас мы располагаем убедительными астрономическими доказательствами, но в прежние времена люди были уверены в том, что Солнце взойдет на востоке, задолго до того, как коперниканская система вытеснила птолемеевскую.) Мы верим в то, что вода, которую мы пьем, безопасна, потому что едва ли можем припомнить случай, когда мы ею отравились. (В некоторых странах третьего мира, впрочем, это далеко не распространенное убеждение.) Мы полагаемся, что официант не снимет лишние деньги с нашей кредитной карты, поскольку раньше официанты всегда были с нами честны. И мы верим в то, что сообщение, полученное по электронной почте, пришло именно от того лица, которое значится в заголовке, поскольку в пользу этого свидетельствует весь наш предшествующий опыт.

Многие криптографические методы существуют благодаря подобной уверенности. Большинство математических моделей основано на оценке вероятности. Криптография с привлечением открытого ключа использует простые числа, и лишь в одном случае из миллиарда число может оказаться в действительности не простым. Однонаправленные хэш-функции, возможно, уникальны: вероятность того, что два различных документа будут иметь одинаковое значение хэш-функций, составляет один к $2^{80}$. Алгоритм шифрования AES предоставляет $2^{128}$ различных ключей, таким образом, вероятность того, что нападающий с первого раза подберет ключ, составляет один к $2^{128}$. Некоторых беспокоят эти цифры, и это объясняется их представлениями о том, что можно обеспечить абсолютную надежность. Однако наступление события, имеющего вероятность один к $2^{80}$, гораздо менее правдоподобно, чем, например, возможность сделать ставку при игре в рулетку на один единственный номер и выиграть пятнадцать раз подряд или дважды подряд получить наилучший набор карт при игре в бридж, или возможность того, что при игре в покер четыре раза подряд придет флеш-рояль.

В этом смысле средства безопасности работают. Большинство устройств сигнализации имеют четырехзначный код, поэтому вероятность того, что взломщик сможет отключить сигнал тревоги, составляет одну десятитысячную. Если цифровой замок имеет три набора по 36 различных комбинаций, вероятность открыть его с первой попытки составляет один на 47 000. Отпечатки пальцев не настолько уникальны, как кажется: существует вероятность, составляющая 0,1%, что постороннее лицо будет идентифицировано как имеющее права доступа. Вот все, что хотелось сказать о вероятности.

# Действия в чрезвычайных ситуациях

Опасность, таящаяся в компьютеризированных системах, состоит среди прочего в том, что они столь редко ошибаются, что люди не готовы к чрезвычайным ситуациям. Распространенное убеждение состоит в том, что компьютер не может ошибиться. На самом деле это происходит постоянно, и «плохие» хакеры рады воспользоваться этим.

Один мой приятель установил у себя дома сигнализацию. Когда она срабатывала, обслуживающая его компания должна была связаться с полицией. Приятель мог отправить сигнал, с помощью которого он сообщил бы компании о том, что это — ложная тревога (поскольку полицейским не хотелось выезжать по каждому вызову без разбора). Также имелся другой сигнал (duress code — сигнал о противозаконном принуждении под угрозой насилия) на всякий случай, который означал: «Мне в голову направлено ружье, и меня вынудили сообщить вам, что это была ложная тревога. Это неправда. Помогите!»

Однажды приятель нечаянно подал сигнал тревоги, и, конечно, уведомил компанию о том, что это была ошибка. Случайно он отправил после этого и сигнал, который мы только что описали. Он сразу осознал свою ошибку и исправил ее. Женщина, принимавшая сообщения, испытала огромное облегчение и сказала: «Благодарение Богу! Я не знала, что мне делать».

Когда сигнализация срабатывает несколько раз в неделю, даже если это ложная тревога, люди знают, что делать. Если же это случается раз в несколько лет, может оказаться, что никто не сталкивался с подобной ситуацией и не знает, как следует поступить. Самодовольные пользователи часто подвергаются нападениям. В этот момент они не отдают себе отчета в происходящем и видят причину неполадок в чем-то другом. Помните Чернобыль? «Эта красная лампочка никогда не загоралась раньше. Интересно, что это значит?..»

Именно поэтому всех нас еще в начальной школе обучают действиям по пожарной тревоге. Мы должны быть готовы к чрезвычайным ситуациям — учения помогают избежать паники и подготавливают нас к мысли, что нечто подобное может случиться. Я никогда не был на пожаре, но знаю, что делать в таком случае. Со мной, скорее всего, все будет в порядке. То же самое можно сказать о путешествиях по воздуху. Когда вдруг сверху падают кислородные маски, никто не собирается попусту отрывать пассажиров от чтения романов и заставлять выяснять, что означают эти глупые выходки... Кассир в банке тоже должен быть внимателен и в подозрительной ситуации не думать так: «Банковский компьютер велит мне выдать этому человеку миллион долларов наличными. Кто я такой, чтобы спорить с компьютером?» Диспетчер ядерного реактора должен быть готов к чрезвычайной ситуации, чтобы не размышлять, что же может означать мигание красной лампочки.

К несчастью, ложные тревоги столь часты, что люди привыкают не обращать на них внимания. «Эта красная лампочка никогда не загоралась раньше. Интересно, что это значит?..» Бывает, однако, еще хуже: «Красная лампочка постоянно вспыхивает, и это ровным счетом ничего не значит. Не стоит обращать внимания». (Вспомните сказку про пастушка, который кричал: «Волки! Волки!») Хуже всего, если надоедливую лампочку вовсе отключат. Этим объясняется эффективность атак, направленных на отказ в обслуживании, и некоторые их сценарии я приводил в главе 3.

Если нападающему удается сломать брандмауэр и перекрыть доступ к сети законным пользователям (атака, приводящая к отказу в обслуживании), они будут недовольны и потребуют, чтобы брандмауэр был отключен до тех пор, пока проблема не будет решена. Когда нападающему удастся вызвать постоянные сбои в работе засекреченной телефонной связи, люди, пользующиеся ею, потеряют терпение и станут вести переговоры по обычному телефону.

Такова человеческая природа. Людям хочется общаться друг с другом, и средства безопасности не должны, по меньшей мере, мешать им. Трудно представить себе, что люди откажутся от разговоров по телефону только потому, что зашифрованная связь не работает. Даже дисциплинированные военные не в силах удержаться от переговоров, когда не могут воспользоваться секретной связью; если уж они неспособны на это, не приходится ожидать такого геройства от простых смертных.

## Взаимодействие человека с компьютером

Мы уже говорили о том, что наиболее небезопасна та система, которая не используется. И чаще всего безопасной системой не пользуются потому, что она вызывает раздражение.

Недавно я выполнял некую работу для одной международной корпорации. Беспокойство вызывало то обстоятельство, что ее руководство часто использовало незащищенную телефонную связь: обычные линии и сотовые телефоны; нередко ему приходилось пользоваться связью в других странах. Чем я мог им помочь? Можно было использовать средства шифрования переговоров, и мы обсудили эти возможности. Качество звука было ниже, чем у обычных телефонов, и оставляло желать лучшего. Инициализация алгоритма шифрования при звонке вызывала задержку на несколько секунд. Телефонные аппараты имели чуть большие размеры, нежели самые миниатюрные и модные сотовые телефоны. Однако переговоры были бы засекречены.

Руководство было не слишком довольно. Ему хотелось иметь безопасную связь, но оно не было готово смириться с плохой передачей звука и задержкой в начале разговора. В конце концов, оно вернулось к обычным незащищенным телефонам.

Люди стремятся к безопасности, но не хотят терпеть неудобства, которые возникают при использовании соответствующих средств. Полезно было бы побеседовать с людьми, которые помнят те времена, когда двери в их жилища были впервые оборудованы замками. Таких людей еще можно встретить в сельской местности. (В городах дома запирались столетиями, а в деревне люди долгое время обходились без запоров на дверях.) Они могут рассказать, каким наказанием были для них дверные замки. Приходилось вначале искать ключ, затем вставлять его в замок и проворачивать там... и все это лишь для того, чтобы попасть в свой собственный дом. Поначалу ключи забывали или вовсе теряли — все это вызывало досаду и раздражение. Конечно, преступность существовала всегда, и дверные замки были полезной вещью, однако люди противились новшествам. Я знаю людей, которые до сих пор оставляют двери незапертыми. (Порочное убеждение: «Со мной это никогда не случится» — чрезвычайно живуче.)

То же самое можно сказать и о средствах компьютерной безопасности. Разыщите людей, которые работали на компьютерах еще в ту эпоху, когда не существовали пароли, допуски и ограничения. Спросите их, как им понравилось введение мер безопасности? Поинтересуйтесь, не пытались ли они обойти средства безопасности просто потому, что так проще работать? Даже сегодня, когда неумолимо приближается срок сдачи работы, люди не долго думая легко игнорируют сред¬

ства защиты. Они оставят открытым черный ход, облегчая посторонним проникновение внутрь здания, и выдадут свой пароль или уберут брандмауэр, лишь бы работа была выполнена. Джон Дейч (John Deutch), бывший директор ЦРУ, приносил домой секретные файлы на своем незащищенном портативном компьютере просто потому, что так было удобнее.

Безопасность — это некоторый компромисс. Средства безопасности проще в тех случаях, когда они видны пользователю и он вынужден иметь с ними дело, принимая адекватные решения, как, например, при проверке имени на цифровом сертификате. С другой стороны, пользователям не хочется иметь дело со средствами безопасности. Тот, кто разрабатывает защиту смарт-карты, тоже не хочет, чтобы средства безопасности были видны клиенту. Он знает, что люди считают меры безопасности обременительными и стараются обходить их по мере возможности, поэтому они каждый раз будут обманывать систему защиты карты. Людям нельзя доверить реализацию политики компьютерной безопасности, поскольку они столь безответственны, что оставляют незапертыми двери своих автомобилей, теряют бумажники и не могут удержаться и не выболтать кому попало девичью фамилию матери.

Люди ненадежны и не поступают должным образом. Исследования, проводившиеся в университете Карнеги-Меллона (Carnegie Mellon University) в 1999 году, показали, что большинство людей не умеют правильно использовать программу шифрования электронной почты PGP. Из двенадцати человек, участвовавших в эксперименте, восемь так и не удосужились узнать, как работает PGP 5.0. Четверо случайно отправляли незашифрованными сообщения, содержавшие конфиденциальную информацию. И это при том, что программа имеет удобный графический интерфейс (справедливости ради следует отметить, что версия PGP 6.0 и более поздние имеют лучший пользовательский интерфейс).

И конечно, от них нельзя ожидать разумных решений в вопросах безопасности. Можно было предположить, что после паники 1999 года в связи с вирусами Melissa и Worm.ExploreZip люди не станут открывать вложения, которых не ожидали увидеть в своей почте. Однако темпы распространения червя ILOVEYOU (и десятков его разновидностей) свидетельствуют о том, что люди ничему не учатся... особенно когда многие компании зарабатывают на том, что пользователи обмениваются любопытными вложениями.

Браузеры используют цифровые сертификаты для того, чтобы обеспечить безопасность соединений. Когда они получают сертификат, они не обязательно предоставляют сведения об идентификации тому, кто находится на другом конце соединения. Это имеет существенное значение для безопасности, поскольку соединение не может считаться безопасным, пока не будет известно, кто находится на другом его конце. Большинство людей не беспокоятся о том, чтобы взглянуть на сертификат, и даже не знают, что это следовало бы сделать (или как это можно сделать).

Те же браузеры могут выводить на экран предупреждения перед загрузкой ап¬плетов Java. Пользователя спрашивают, доверяет ли он некоторому веб-серверу, с которого отправляется апплет. Пользователь не имеет представления о том, стоит ему доверять ему или нет. Или не беспокоится об этом. Если некто, бесцельно блуждающий по Интернету, щелкает на кнопке, обещающей вывести на экран пля¬

шущих поросят, и получает предупреждение о возможных опасностях апплета, он сделает выбор в пользу поросят, а не стабильной безопасности своего компьютера. Если его попытаются образумить предупреждением вроде: «Апплет DANCING PIGS может содержать вредоносный код, способный вызвать непоправимые повреждения вашего компьютера, похитить все ваши сбережения и лишить вас способности к деторождению», он щелкнет OK, даже не прочитав его. Через тридцать секунд он и не вспомнит, было ли подобное предупреждение.

## Автоматизм действий пользователя

Когда в главе 6 я излагал основы криптографии, я говорил о том, как Алиса и Боб шифруют, расшифровывают, подписывают и верифицируют сообщения и документы. Например, я упоминал, как Алиса может применять шифрование с помощью открытого ключа: если она найдет ключ Боба в телефонной книге, она может использовать его для шифрования своего сообщения Бобу. На самом деле все совсем не так. Алиса никогда не шифрует и не подписывает свои сообщения Бобу. Она никогда не расшифровывает входящие сообщения. Она вообще не занимается криптографией. Все, что она делает, так это щелкает на соответствующей кнопке, а компьютер шифрует или подписывает сообщения и вообще делает то, чего хочет Алиса. В этом состоит принципиальное отличие.

Представьте себе, что в будущем мы будем постоянно подписывать цифровые документы. Как это будет выглядеть? Алиса составит документ с помощью некоторого приложения — текстового редактора, почтовой программы и т. п. — и щелкнет на определенном значке, чтобы сообщить о своей готовности подписать его. Приложение вызовет некоторую программу, которая создаст подпись. Алиса введет свой пароль (слово или целую фразу) или приложит палец к устройству идентификации, или еще каким-либо способом подтвердит, что она именно та, за которую себя выдает. Программа составит цифровую подпись и передаст ее вызвавшему приложению для присоединения к документу. Вуаля — подпись готова. Возможно, Алиса даже проверит подпись (снова с помощью компьютера), чтобы убедиться, что она правильная.

Это то, что я называю *автоматизмом действий пользователя* (human-computer transference). Алиса знает, чего она хочет, — подписать документ. Она должна иметь определенные гарантии того, что компьютер сделает именно то, что ей нужно. Однако обеспечить безопасность не так просто.

Предположим, мы хотим заставить Алису подписать нечто, что ей подписывать не хочется. Это легко сделать, если она поверит в то, что подписывает именно тот документ, который видит на экране. Нам нужно заставить компьютер обмануть Алису.

Мы напишем троянского коня, который будет размещаться в программе, создающей цифровую подпись. Этот троянский конь будет содержать документ, который мы хотим подсунуть Алисе, — несомненно, или что-нибудь постыдное, или сулящее выгоду, — и код для его подписания. Единственное, чего не хватает троянскому коню, — это ключа Алисы. Когда она вводит свой пароль, чтобы подписать какое-нибудь сообщение для нас, троянский конь подсовывает вместо него

компрометирующий документ. Программа, создающая цифровую подпись, возвращает ее основному приложению, которое присоединяет подпись к документу, который Алиса действительно собиралась заверить. Если она попытается проверить подпись, троянский конь снова подсунет подделку, и программа, создавшая подпись, подтвердит, что она правильная. Таким образом, троянский конь может заставить компьютер солгать Алисе. Затем она отправляет нам свое сообщение с ошибочной подписью — составленной для совершенно другого документа. Мы присоединяем эту подпись к копии нашей подделки и звоним в *Washington Post*. Тем временем троянский конь самоуничтожается, и все возвращается в исходное состояние.

Это легко реализовать в среде Windows: можно создать макрос, который будет попросту наблюдать за «открытым файлом» диалога PGP, копировать свой собственный файл под именем того, который как раз собирается подписать Алиса, и впоследствии возвращать старый файл на место. Язык макрокоманд программы Word позволяет сделать это, поэтому совсем не трудно создать такой макровирус.

Это всего лишь один пример. Троянский конь может подсунуть для подписи оба документа и переслать подделку позже, в подходящий момент. Или просто похитить ключ Алисы.

В этом нет ничего сложного, программирование — простое дело. Во всяком случае, если мы достигнем успеха, мы завладеем опасным для Алисы документом с ее подписью. Мы можем размахивать им в суде или передать журналистам, обоснованно заявляя, что под документом стоит действительная подпись. Однако вероятнее всего, что нас постигнет неудача. Как только кто-нибудь напишет троянского коня, подделывающего подписи, он распространится повсеместно. И если документ будет предъявлен в суде, одна из сторон может представить свидетельство эксперта о существовании этого троянского коня и о том, с какой легкостью можно заставить кого угодно поставить свою подпись неведомо на чем. Примет ли суд эту подпись в качестве доказательства? Все зависит от обстоятельств, а не от математических методов.

Суть фундаментальной проблемы состоит в том, что у нас нет никакого представления, что же в действительности делает компьютер по нашей команде. Когда мы приказываем ему сохранить документ, или зашифровать файл, или сложить числа в столбце, у нас нет уверенности в том, что этот «черный ящик» выполнит задание правильно или даже вообще его выполнит. Мы вынуждены все принимать на веру. Насколько трудно поймать за руку вороватого работника, настолько же трудно выловить разрушительную программу. Дело обстоит даже хуже. Представьте себе, что этот сотрудник работает в одиночестве и за ним некому уследить, Все средства наблюдения, которые мы можем установить, — скрытые камеры и микрофоны — управляются самим же этим работником. Все, что мы можем сделать, — это сравнивать поступающие к нему («на входе») материалы с производимым им продуктом («на выходе»). Но и этого недостаточно для полной уверенности в его честности.

Если Алиса не может доверять компьютеру, на котором работает, то она не может быть уверена в том, что он точно выполняет ее команды. Хотя бы потому, что когда она велит ему подписать документ, это не означает, что он не способен поставить ее подпись под другим документом. Проблему можно исчерпать, когда Алиса станет подписывать документы лишь на совершенно надежном компьюте¬

ре, а это трудно осуществить. Если речь идет о компьютере общего назначения, я никогда не поверю в то, что он достаточно надежен.

Вот если бы у Алисы был маленький компьютер с единственным назначением — создавать цифровые подписи, — тогда было бы на что надеяться. Я в состоянии представить карманное устройство с миниатюрной клавиатурой и экраном, в которое можно перенести документ с компьютера общего назначения. Алиса сможет просмотреть документ на экране — так как нет никакой гарантии, что «большой» компьютер загрузит именно то, о чем его попросят, — и ввести пароль с клавиатуры. Устройство создаст подпись под документом и передаст ее назад компьютеру общего назначения. Нам остается молиться о том, чтобы такая система была безопасной. Можно сконструировать ее так, чтобы устройство использовало только заводское программное обеспечение, а независимая проверяющая компания будет выдавать свидетельство о том, что программа работает правильно.

Однако при работе на небезопасном компьютере — то есть практически всегда — нет никакой гарантии того, что мы видим на экране именно то, что получили, или что оно работает так, как мы могли бы ожидать.

## Внутренние враги

В главе 4 я уже рассказал о внутренних врагах. Стоит вспомнить о тех проблемах, которые с ними связаны. Основная проблема заключается в том, что они пользуются безоговорочным доверием. Для них не проблема стащить деньги из кассового аппарата, внести путаницу в отчетность и замести следы, скопировать и переправить китайцам военные секреты, похитить набор бланков кредитных карточек, набить карманы фишками в казино, найти другой путь, чтобы грабители могли безопасно вывезти на грузовике краденое добро, и многое другое. Часто никакие средства компьютерной безопасности не в состоянии предотвратить нападения такого рода (хотя тщательная проверка имеет шансы впоследствии выявить виновных).

Киберпространство особенно чувствительно в этом отношении. Тот, кто пишет программу защиты, может предусмотреть «черный ход» в нее. Тот, кто устанавливает брандмауэр, может оставить тайную брешь в защите. Тот, кто по роду своей деятельности обязан проверять работу системы безопасности, может сознательно пропустить какие-то вещи.

Вот только один пример. В Чикаго для оплаты проезда в метрополитене используются как жетоны, так и проездные билеты. Пассажиры должны либо отдать контролеру жетон, либо предъявить проездной, после чего их пропускают на платформу. Многие годы контролеры принимали от пассажиров жетоны, а отмечали их как владельцев проездных билетов. В конце концов, в 1991 году руководство узнало об этом, и контролеры были арестованы. По самым скромным оценкам, ущерб составил сотни тысяч долларов. Когда начали работать более честные контролеры, дневная выручка на некоторых станциях удвоилась. На решение этой проблемы ушло несколько лет.

Компании пытаются снизить риск ущерба от внутренних врагов различными способами. Наилучшее решение состоит в том, чтобы принимать на работу чест¬

ных людей, но это легче сказать, чем сделать. Некоторые компании зашли так далеко, что проводят предварительный отсев претендентов, подвергая их проверке на честность. Другие пытаются распределить ответственность между сотрудниками и ограничить возможности причинения ущерба одним работником. Для того чтобы иметь возможность установить, какой вред причинен сотрудником, и предъявить ему обвинение в суде, необходимо проводить проверки. Тем не менее любая организация оказывается во власти работающих в ней людей.

## Манипулирование людьми

В 1994 году французский хакер по имени Антоний Зборальски позвонил в вашингтонский офис ФБР и представился представителем этой организации, работающим в американском посольстве в Париже. Он убедил собеседника на другом конце провода, и тот объяснил ему, как подключиться к системе телеконференции ФБР. Его звонки за последующие семь месяцев стоили ФБР 250 000 долларов.

Вообще, это распространенный хакерский прием — позвонить ничего не подозревающему работнику и представиться системным администратором сети или руководителем службы безопасности. Если хакер достаточно осведомлен об особенностях корпоративной сети, чтобы выглядеть убедительно, он может выудить у работника пароли, учетные записи и другие важные сведения. В одно прекрасное мгновение хакер размещает на доске объявлений компании новый телефон «справочного стола»[1] — свой собственный. Сотрудники компании будут регулярно звонить ему, и он соберет богатый урожай паролей и данных по учетным записям в обмен на свою помощь.

Манипулирование людьми — это хакерский термин для обозначения такого рода мошенничества: убедить кого-нибудь сделать то, что вам нужно. Это весьма эффективно. Манипулирование людьми преодолевает и криптографию, и средства компьютерной или сетевой безопасности, и любые другие технические приемы. Оно направлено непосредственно на самое слабое звено любой системы безопасности: бедный человечек старается выполнить свою работу и готов выручить другого, если в силах чем-то помочь.

Печально, но это гораздо проще, чем можно подумать. Бывает достаточно появиться в компьютерной фирме с каким-нибудь «железом» в руках, нацепив соответствующую эмблему компании-поставщика. Если побродить вокруг и поинтересоваться, нельзя ли здесь где-нибудь примоститься и чуток поработать, скорее всего, можно оказаться за клавиатурой и получить выход в сеть: ясно ведь, что это — посетитель компании.

По большей части манипулирование людьми осуществляется по телефону, что затрудняет поимку злоумышленника. Один из них звонил разным людям и заявлял: «Это оператор. Вам звонит за ваш счет... (имярек) из такого-то города». Если жертва соглашалась принять звонок, «оператор» продолжал: «Функция соединения за ваш счет заблокирована. Сообщите мне, пожалуйста, номер вашей карты, и я соединю вас». Это происходило в действительности. Злоумышленник находил

[1] Системы поддержки пользователей в Сети (help-desk). — *Примеч. перев.*

людей в конференциях Usenet и придумывал звонки за их счет от корреспондентов по конференции, что выглядело весьма правдоподобно.

Когда в 2000 году Кевин Митник давал показания в Конгрессе, он говорил о манипулировании людьми: «Нападения этого рода были столь успешны, что мне редко приходилось обращаться к техническим средствам. Компании способны истратить миллионы долларов на технические средства защиты, но все это будет напрасно, если можно позвонить по телефону и убедить кого-нибудь сделать на компьютере нечто, что ослабляет защиту или открывает доступ к интересующей информации».

Другой вид нападений с помощью манипулирования людьми направлен против кредитных карт. Предположим, Алиса узнала номер кредитной карты Боба. Она могла бы сделать покупки за его счет, но она более коварна. Она дает рекламу различных товаров — видеокамер, компьютеров, чего угодно еще — и предлагает их по очень низкой цене. Карл попадается на удочку и покупает какой-то продукт у Алисы. Она в свою очередь заказывает этот продукт у настоящего продавца, используя номер кредитной карты Боба. Продавец доставляет покупку Карлу и бывает озадачен, когда впоследствии Боб обнаруживает траты. В этом случае виновным посчитают Карла, а не Алису.

Автоматизированное манипулирование людьми можно использовать сразу против большого количества людей, и кто-нибудь постоянно будет одурачен. В 1993 году подписчики New York ISP Phantom Access получили следующее зловещее сообщение: «Мне стало известно, что ваша учетная запись была взломана посторонним лицом. Внесенные изменения столь значительны, что это позволило обнаружить ошибку. Пожалуйста, временно измените пароль для доступа к DPH7, чтобы мы смогли оценить серьезность вторжения. Когда проблема будет решена, я извещу вас. Благодарю за содействие. — Системный администратор». А в 1999 году пользователи AOL постоянно получали сообщения вроде: «В результате произошедшей ошибки была удалена из базы данных информация о более чем 25 000 учетных записей, в том числе и вашей. Чтобы получить доступ к резервным данным, нам необходим ваш пароль. Если мы не получим его, мы НЕ будем иметь возможности позволить вам подписаться на America Online в течение 24 часов, следующих за открытием настоящего письма».

Правдоподобие и неожиданность сообщения, ужас, который оно вызывает, заставляют жертву сдаться. Распространяемые по электронной почте вирусы и черви используют автоматизированное манипулирование людьми, чтобы заставить получателей открыть их. Вирус ILOVEYOU прятался в посланиях от людей, известных получателю. Сообщения имели правдоподобную тему и содержание, побуждавшее открыть вложение. Вложение было представлено как безвредный текстовый файл, на самом же деле это был файл VBScript. Я уже говорил об этом в главе 10. Люди не могут противодействовать таким вирусам, использующим манипулирование людьми.

В некоторых случаях технические средства безопасности в состоянии помочь. Если бы всегда готовые прийти на подмогу сотрудники помимо пароля должны были использовать маркеры доступа (или средства биометрического контроля), то славный парень на другом конце провода не сумел бы получить все, чего хотел. Его не выручил бы пароль, если бы компьютер имел устройство для распознава¬

ния отпечатков пальцев. Если бы системы были достаточно разумны и распознавали, что кто-то входит в них с удаленного компьютера, тогда как правила предписывают делать это исключительно на рабочем месте, возможно, кого-нибудь это встревожило бы.

Иногда предотвратить манипулирование людьми позволяют довольно простые процедуры. В Военно-морских силах США используются сейфы с двумя замками (конечно, с различными комбинациями цифр); один человек не должен знать шифр обоих замков. Это значительно затрудняет управление людьми. Можно использовать многие компьютерные трюки, чтобы ограничить возможности обманутого законного пользователя оказать помощь мошеннику. Технические средства способны усложнить задачу мошенника, в некоторых случаях весьма значительно.

Но скорее всего, манипулирование людьми будет действенно всегда. Взгляните на проблему глазами жертвы — Боба. Боб — славный малый. В компании он принадлежит к числу сотрудников низшего или среднего звена. Он не работает в службе безопасности, но, несомненно, прошел некоторую подготовку по этой части и знает, что нужно быть настороже, так как настырные хакеры не дремлют. Но вообще-то Боб довольно невежествен и не понимает проблем безопасности системы. Ему неведомы тонкости нападений. Он хочет лишь выполнять свою работу и готов прийти на помощь.

Мошенница Алиса приходит к Бобу со своей проблемой. Она, как и Боб, всего лишь винтик в механизме компании. Ей тоже нужно сделать свою работу. Все, чего она хочет от Боба, — узнать его пользовательское имя и пароль или какой-либо номер телефона, подключить здесь какое-нибудь «железо» или сделать еще что-нибудь вполне осмысленное. Конечно, формально это запрещено, но у Алисы какие-то затруднения, и ей нужно сделать всего лишь это. Каждый хоть однажды обходил средства безопасности, когда торопился закончить работу вовремя. Неужели Боб откажет в помощи? Или он не член команды? Разве он не знает, как часто глупые правила компании мешают работе? Конечно, он поможет. Он добрый.

Именно поэтому приемы манипулирования людьми работают. Люди в основном доброжелательны и приходят на помощь. И их часто обманывают. Взывая к естественным душевным качествам Боба, Алиса всегда добьется своей цели. Она убедит Боба, что она — такая же, как и он сам. Она позвонит в тот момент, когда Боб менее всего ожидает этого. Она знает, что средства безопасности мешают его работе, и может сыграть на этом. Если же она допустит ошибку, и Боб не попадется на удочку, она может обратиться к десяткам и сотням других Бобов в той же организации, которые сделают то, о чем их попросят.

# Часть III

# Стратегии

До сих пор мы изучали проблему по частям. Мы рассмотрели основные существующие угрозы. Мы познакомились с разными видами атак и типами злоумышленников. Мы многое узнали о различных технологиях и о том, каким образом они противодействуют (или почему не могут противодействовать) нападениям. Пришло время собрать это все вместе и попытаться решить некоторые проблемы безопасности.

Руководителей можно разделить на три категории в зависимости от их подхода к вопросам безопасности. Первая категория считает: «В этой области дела обстоят слишком плохо». Эта точка зрения, согласно которой системы безопасности настолько ненадежны, что просто невозможно вести учет товаров с помощью карманных компьютеров (PDA), осуществлять банковские платежи через Интернет или участвовать в лотерее с помощью сотового телефона. Вторая категория придерживается позиции «Я куплю себе безопасность». Согласно этой точке зрения, безопасность — всего лишь отметка об оплате в товарном чеке, и если вы приобрели брандмауэр, то защита вам обеспечена. Обе категории придерживаются крайних взглядов, и любая из этих позиций отражает упрощенческий взгляд на вещи. Третья категория рассуждает еще более странно: «Мы слишком незначительны, чтобы нас атаковали». Эта позиция более чем упрощенческая.

Мы хотим предложить нечто лучшее. Дела можно вести безопасно в цифровом мире так же, как можно их вести в мире реальном. На первый взгляд лучший способ обеспечить безопасность состоит в использовании как можно большего количества средств защиты: повесить больше замков на двери, везде использовать шифрование, брандмауэры, системы обнаружения вторжения, инфраструктуру открытого ключа в компьютерных сетях. К сожалению, все далеко не так просто. Во-первых, финансирование системы безопасности ограничено. Во-вторых, нагромождение различных средств защиты — не лучший путь к достижению цели.

Задача состоит в том, чтобы понять, как работает целостная система и как в нее вписываются средства защиты. Бесполезно рассматривать только отдельные технологии.

Безопасность — это цепь; ее прочность определяется прочностью самого слабого ее звена. Если вы решили осуществить зашифрованное телефонное соединение, вы должны побеспокоиться об алгоритме шифрования голосовой связи, о механизме обмена ключами, который позволит участникам диалога понимать друг друга, о процессе создания ключа, о безопасности программного обеспечения на телефонной станции, о физической безопасности аппаратов и т. д. Изъян в любом из этих звеньев сведет на нет все усилия.

То же самое относится к компьютерным системам. Если у вас есть сеть с брандмауэром, вы должны подумать о безопасности самого брандмауэра. Если ваша сеть снабжена брандмауэром и VPN, необходимо, чтобы были защищены оба этих устройства. Уязвимое место в одном из них может привести к тому, что вся сеть окажется в нерабочем состоянии.

Безопасность — это процесс, а не продукт. В этой части мы будем обсуждать процессы, связанные с безопасностью: атаки, способы защиты и взаимоотношения между ними. Мы поговорим о том, как осуществляются реальные нападения, и о том, как сконструировать систему, способную противостоять таким нападениям. Мы также поговорим о текущем состоянии средств безопасности, об их будущем и о том, что необходимо для их эффективной работы.

# Глава 18. Уязвимости и их ландшафт

В первой части мы теоретически рассматривали атаки: какие существуют виды нападений и нападающих. Но, как я каждый раз не устаю повторять, теория отличается от практики. Каждый, кто читает детективные романы или криминальную хронику в газетах, знает, что для осуществления нападения необходимо намного больше, чем просто найти уязвимое место. Можно считать, что нападающий с успехом использовал слабую точку, если ему удалось определить цель, спланировать атаку, сделать свое дело и скрыться. Недостатки в замке сейфа, находящегося в безопасном месте, менее существенны, чем аналогичные недостатки в банковском уличном ящике для депозитов.

То же самое и в цифровом мире. Потенциальному преступнику недостаточно найти изъян в алгоритме шифрования, использующемся в системе кредитных карт. Он должен получить доступ к соединению, он должен обладать достаточными знаниями о протоколах, чтобы создать поддельное сообщение, которое позволит ему на самом деле украсть деньги, и в конце концов он должен успеть убраться вовремя. Обнаружение изъяна в шифровании в отрыве от всех остальных шагов представляет только теоретическую ценность.

Подобно этому, существует великое множество мер противодействия, которые могут помочь «заткнуть дыру». В сейфе можно повесить более надежный замок или установить сигнализацию на окнах и дверях помещения, в котором сейф находится, и поставить у дверей охрану. Недостатки в шифровании могут быть исправлены, если использовать лучший алгоритм шифрования. Они не будут представлять опасности, если держать протоколы в секрете, использовать частную сеть для сообщений или просто менять ключи каждые пять минут.

## Методология атаки

В общем случае успешную атаку можно разбить на пять последовательных шагов:

1. Опознать цель, которая должна подвергнуться нападению, и собрать о ней информацию.
2. Проанализировать информацию и найти уязвимую точку в цели, которая позволит добраться до объектов, на которые направлена атака.
3. Получить необходимый уровень доступа к цели.

4. Осуществить нападение на цель.
5. Завершить атаку, что может включать в себя уничтожение всяких следов атаки, и скрыться от возмездия.

То есть, другими словами, необходимо определить, что атаковать и как атаковать, проникнуть внутрь, провести атаку и убраться вовремя. Первые два шага — это исследование. Вы в силах выполнить их в полной безопасности в собственной лаборатории; вы даже можете использовать для этого муляжи настоящей цели. Если вы ученый, скорее всего, вы остановитесь после второго шага и опубликуете полученные материалы. Последующие три шага связаны с риском, который неизбежен при взломе и проникновении в систему как виртуальную, так и реальную. Тут уж как повезет: либо успел скрыться вовремя, либо поймали.

Помните *Звездные войны?* Для того чтобы взорвать Звезду Смерти, повстанцы сначала должны были получить информацию, которую принцесса Лейа поместила в R2-D2. Это была единственная причина, по которой Люк должен был в первую очередь вызволить дроидов с Татуина. Спасение принцессы было в Мак-Гаффине. Это был первый шаг.

Шаг второй остался за кадром. Инженер восставших изучил информацию, полученную от дроидов и нашел слабое место в обороне их станции — проектировщики системы жизнеобеспечения никогда не предполагали, что их детище будут внимательно изучать профессионалы в области безопасности, и вот результат: Звезда Смерти, создание которой обошлось в десятки миллиардов «кредитов», может быть уничтожена через вентиляционную шахту.

Шаг третий продемонстрировал спецэффекты ближнего боя в ограниченном пространстве между крестокрылыми бойцами повстанцев (их экипировка приводит зрителей в восхищение) и защитниками станции. Задача крестокрылых состояла в прикрытии нескольких истребителей, которые получали возможность свободно перемещаться по всему пространству и простреливать насквозь вентиляционную шахту. Доступ к цели был достигнут совместными усилиями.

Молодой мастер Люк смог завершить четвертый шаг после того, как Хан Соло оторвался от Дарта Вейдера, севшего ему «на хвост», и вкрадчивый голос Алека Гинеса внушил Люку мысль отключить компьютер, который являлся целью (возможно, работающий с бета-версией), и использовать Силу.

Взрыв Звезды Смерти (шаг 5) исключил всякую возможность возмездия, по крайней мере на некоторое время. После этого было нетрудно убраться восвояси. Наши герои получили медали от альянса повстанцев, чье материальное положение улучшилось настолько, что они смогли позволить себе заказать новую форму, и Вселенная была спасена на несколько следующих серий. Список можно продолжить.

Только что описанный пример мало отличается от атаки через Интернет, нацеленной на компьютеры какой-либо компании. На первом шаге происходит выбор цели и сбор информации. Осуществить это на удивление легко. На веб-сайте обычно можно найти любую информацию: от сведений о том, что содержат различные базы данных Интернета, до способов работы с ними в режиме сетевого подключения. Специальный номеронабиратель поможет установить коммутируемое соединение. Существует множество техник, которыми нападающий может воспользоваться для того, чтобы узнать, как работает сеть, в которой находится его цель:

изучение результатов работы специальных программ, используемых для проверки доступности адресата, служебных сообщений, состояния портов и т. д. Сетевой модуль проверки текущего состояния способен выдать еще больше информации. Это чаще всего похоже на вышибание двери, хотя компьютер сам готов выдать совершенно посторонним людям массу информации о том, какое аппаратное обеспечение в нем используется, какие работают программы и какие службы они поддерживают. Все эти сведения могут с успехом использоваться нападающим.

Второй шаг — это нахождение уязвимого места. На этом этапе атакующий внимательно изучает всю собранную информацию, для того чтобы выбрать точку атаки. Вероятно, на одном из компьютеров установлена определенная версия почтовой программы, операционная система или Solaris, или Windows NT, которые содержат известные ошибки. Возможно, удастся использовать FTP или регистрационное имя, или что-нибудь еще. Часть оборудования, обеспечивающего поддержку порта, через который лежит путь к цели, может оказаться незащищенной. Не исключено, что нападающий в силах использовать телефонную сеть объекта, на который направлена атака. Чем больше уязвимых мест в различных частях системы известно атакующему, тем лучше он сможет спланировать нападение.

Шаг третий — получение некоторого вида доступа к компьютеру. В Интернете это тривиально, так как любой компьютер подключен к Сети и таким образом доступен. (Конечно, некоторые компьютеры находятся за брандмауэром и недоступны, но тогда брандмауэр, по всей видимости, доступен.)

Четвертый шаг — проведение атаки. Это может оказаться сложным делом, а может, наоборот, пустяковым. Если нападающий хорошо подготовлен, все проходит удивительно легко.

Заметим, что некоторые атаки включают в себя множество итераций. Нападающий может выполнить шаги с первого по четвертый неоднократно: проникновение на веб-сервер, получение необходимого уровня доступа, использование этого доступа для проникновения на другой сервер, расположенный внутри общей территории, защищенной брандмауэром, получение к нему доступа и т. д. На каждом шаге происходит сбор информации, определение цели и методов ее достижения, получение доступа и выполнение операции.

Пятый шаг — завершение атаки. Если нападающий искал какой-то определенный файл, он получает к нему доступ, делает что ему было нужно и исчезает. В его власти стереть записи в контрольном журнале и таким образом уничтожить все следы. Он также способен модифицировать системные файлы, чтобы было легче получить доступ в следующий раз. И стоит ему почувствовать опасность, быстро завершает начатое и скрывается. И исчезает мгновенно. Хождение вокруг да около характерно для любителей.

В руководстве по взлому «FAQ and Guide to Cracking» Микстера описаны те же самые шаги. Вот что он говорит о том, что нужно сделать в первую очередь после того, как вы получили корневой доступ (получение прав корневого пользователя на копьютере-цели выполняется на шаге 4):

1. Поэтапно удалить следы получения несанкционированного корневого доступа.
2. Собрать основную информацию о системе.

3. Удостовериться, что вы сможете выбраться оттуда.
4. Разрушить или обновить уязвимый домен.

Он особенно подчеркивает необходимость отключения контроля и уничтожения записей в контрольном журнале и получения информации о том, как часто система подвергается проверкам и как часто проводится анализ информации, хранящейся в контрольном журнале.

Хакерские инструменты позволяют автоматизировать множество процессов. Они действуют далеко не так эффективно, как виртуозный хакер, но они способны превратить несмышленого подростка в опасного противника.

Другой пример: атака против платежной системы, использующей смарт-карты. Первый шаг состоит в том, чтобы собрать всю информацию о платежной системе, которая может пригодиться: особенности ее построения, общедоступную документацию, сведения об используемых алгоритмах и протоколах и т. д. Возможно, вам удастся раздобыть массу информации, если вы знаете, где ее искать.

Шаг второй — изучить документацию, отыскивая слабое звено. Во второй части этой книги говорилось обо всех видах уязвимых мест, которые могут существовать в подобных системах. Уязвимыми могут быть алгоритмы и протоколы шифрования. Возможно, уязвимая точка скрывается в самой смарт-карте, из-за того, что система сопротивления вторжению не действует как предполагалось. Может быть, существует изъян в способах их использования — если его удастся обнаружить, это поможет достичь цели. Вам нужно найти все уязвимые точки, для того чтобы успешно атаковать систему.

На третьем шаге необходимо получить уровень доступа, необходимый для проведения атаки. Вы должны стать зарегистрированным пользователем этой платежной системы (возможно, под вымышленным именем). Вероятно, вам придется украсть чью-либо карту. Может быть, вам необходимо будет вступить в сговор с продавцом, который принимает смарт-карты в качестве платежного средства. Получение доступа — не всегда легкое мероприятие.

Четвертый шаг — выполнение атаки: размножить смарт-карту и использовать ее дубликаты, изменять содержимое памяти смарт-карты и пользоваться этим при совершении покупок, изменять баланс и требовать оплаты наличными — все, что вам удастся осуществить. Для выполнения последнего пункта вам недостаточно взломать систему смарт-карт, вы должны перевести все добытое с помощью взлома в наличные деньги.

Шаг пятый — заметание следов. Возможно, вы захотите ликвидировать все физические доказательства вашего нападения. Если оборудование, которым вы пользовались для осуществления атаки, находилось у вас дома, вы его уберете оттуда. Если доказательства вашего нападения остались в компьютерных файлах, вы их удалите. Может быть, вам удастся взломать компьютеры платежной системы и уничтожить записи, представляющие для вас опасность. Все что угодно, лишь бы замести следы.

В некоторых атаках присутствуют не все описанные этапы. Нападения ради огласки часто не включают в себя шаги 2, 3 или 5. Приведем в качестве примера атаку против алгоритма шифрования, использующегося в цифровых сотовых телефонах. Шаг 1 — получение информации об алгоритмах шифрования, применяющихся в сотовых телефонах. Шаги 2 и 3 пропускаются. (Цель известна и весь

доступ, который вам нужен, — это описания алгоритмов.) Шаг 4 — выполнение криптографического анализа и сообщение об этом средствам массовой информации. Шаг 5 не выполняется — вы ничего не делали нелегально. Такая атака потенциально успешна против любого алгоритма шифрования данных для цифровой сотовой связи.

В этой книге я на многих примерах постарался показать, что безопасность — это цепь, и надежность системы определяется прочностью самого слабого ее звена. Уязвимые точки как раз и представляют собой такие слабые звенья. Нахождение слабых мест в системе — это только первый шаг к их использованию. Не менее важно получить доступ к обнаруженной уязвимости, суметь реально использовать ее для совершения определенных действий и затем благополучно скрыться — без этого не бывает успешных нападений.

## Меры противодействия

Меры противодействия существуют для того, чтобы защитить уязвимые точки. Они могут быть простыми, наподобие постройки стены вокруг города, чтобы вражеская армия не смогла в него проникнуть, или сложными, такими как создание надежной системы проверки для обнаружения попыток мошенничества среди продавцов кредитных карт и установления личности преступников.

Обычно меры противодействия применимы для разрушения атаки на любом из пяти этапов ее проведения.

Большая часть обсуждавшихся во второй части технических мер противодействия применяется в компьютерах и компьютерных сетях. Я пытался обрисовать ситуацию в целом: как работают различные средства и методы или почему они не работают, каким образом они соотносятся друг с другом и т. д. Ни одна технология в области безопасности не должна рассматриваться как панацея: результат достигается, когда каждая из них используется эффективно.

Надежность системы всегда определяется ее самым слабым звеном, и это, вообще говоря, заставляет нас обратиться к рассмотрению отдельных технологий. В умело построенной системе эти технологии не лежат на поверхности, в конечном счете безопасность системы определяется их взаимодействием. Криптографические методы могут быть разрушены с помощью лобовой атаки или криптоанализа алгоритма. Можно также воспользоваться невнимательностью сотрудника и раздобыть пароль. Но замок на двери помещения, в котором находится компьютер, или хорошо сконфигурированный брандмауэр обеспечит защиту на другом уровне.

Помните начало *В поисках утраченной радуги (Raiders of the Lost Arc)?* Индиана Джонс должен был пройти через пауков, ловушки с шипами, ямы, отравленные стрелы, внезапно вылетающие, если наступишь не на тот камень, и саморазрушающееся устройство, срабатывающее при движении статуи. Это многоуровневая защита. Он преодолел ловушку с шипами, не затронув пусковой механизм, но он еще должен был увернуться от надвигающейся стены, остановить механизм и сделать массу других вещей. Самый легкий способ избежать ловушки определяет ее эффективность.

Так же как нападение на систему представляет собой нечто более сложное, чем просто нахождение уязвимых мест, защита системы более сложна, чем выбор мер

противодействия. Эффективная система таких мер покоится на трех составляющих:

- защита;
- обнаружение;
- реагирование.

В офисе военной организации служебные документы хранятся в сейфе. Сейф обеспечивает защиту от возможного проникновения, но той же цели служит сигнализация и охрана. Предположим, что нападающий — посторонний человек: он не работает в офисе. Если он попытается украсть документы из сейфа, он должен не только взломать сейф, ему нужно еще отключить сигнализацию и суметь пройти мимо охраны. Сейф с замком — это меры защиты, сигнализация — способ обнаружения вторжения, охрана обеспечивает реагирование.

Если охрана обходит офис через каждые пятнадцать минут, то сейф должен противостоять атакующему в течение пятнадцати минут. Если сейф находится в офисе, персонал которого присутствует только в рабочие часы, он обязан обладать способностью выдержать атаку в течение шестнадцати часов: от пяти часов пополудни до девяти утра следующего дня (и намного дольше, если офис закрыт в выходные дни). Если сейф снабжен сигнализацией, и как только кто-нибудь дотронется до него, сразу прибудет охрана, тогда он должен выдерживать атаку только в течение того времени, которое потребуется ей, чтобы добраться до места происшествия и принять меры.

Все сказанное означает, что надежность сейфа основывается на механизмах обнаружения и реагирования на месте. И сейфы классифицируются по этому признаку. Одному сейфу может быть присвоена классификация TL-15: это означает, что он способен противостоять профессиональному взломщику с инструментами в течение 15 минут. Другой сейф может быть отнесен к разряду TRTL-60, что будет означать, что ему по плечу сопротивляться такому же взломщику, да еще вооруженному паяльной лампой с подачей кислорода, 60 минут. Это оценки чистого времени атаки: время идет, только когда сейф подвергается нападению, — время, затраченное на планирование и подготовку, не учитывается. И тесты проводятся профессионалами, имеющими доступ к чертежам сейфа: нельзя рассчитывать на недостаток информации у нападающего. (Много общего с криптографическими атаками, не правда ли?)

Меры защиты, обнаружения и реагирования работают совместно. Сильный механизм защиты подразумевает, что вы не нуждаетесь в столь же действенных механизмах обнаружения и реагирования.

Классификация сейфов демонстрирует это ясно. Какой сейф вы купите: рассчитанный на 15 минут, на 30 минут, на 24 часа? Это будет зависеть от того, в течение какого времени сработает сигнализация (обнаружение) и прибудет охрана, чтобы арестовать взломщиков (реагирование). В отсутствие систем обнаружения и реагирования в действительности все равно, какой сейф вы выберете.

Большинство мер компьютерной безопасности носят профилактический характер: криптография, брандмауэры, пароли. Некоторые можно отнести к механизму обнаружения, как системы обнаружения вторжения. Реже встречаются механизмы реагирования: например, система ввода регистрационного имени и пароля, ко¬

торая блокируется после трех неудачных попыток — хотя механизмы обнаружения бесполезны без механизмов реагирования. Представьте себе систему обнаружения вторжения, которая только регистрирует атаку. Она подаст сигнал системному администратору, может быть, пошлет сообщение по электронной почте на его пейджер. Если администратор не отвечает в течение нескольких часов — допустим, он обедает, — тогда не имеет значения, что нападение было обнаружено. Не было предпринято никаких мер, чтобы решить проблему.

Обычная охранная сигнализация также является средством обнаружения. Когда она срабатывает, дальнейшее развитие событий зависит от того, есть ли кому реагировать на нее. Если взломщик знает, что на сигнал тревоги никто не обратит внимания, это то же самое, как если бы сигнализации не было вовсе.

Иногда невозможно использовать механизмы обнаружения и реагирования. Представьте себе обычное прослушивание: Алиса и Боб общаются с помощью незащищенной связи, а Ева их подслушивает. Ни Алиса, ни Боб не могут обнаружить прослушивание и, соответственно, у них нет возможности как-то отреагировать на него. Средство защиты — шифрование — должно обеспечить достаточно безопасную связь, чтобы прослушивание дало результаты, представляющие интерес для Евы.

Сравним только что приведенный пример с шифрованием кодов доступа для системы кредитных карт. Предположим, что заполучить эти коды можно только взломав систему. Если на всех автоматах установлена сигнализация (обнаружение) и коды доступа могут быть изменены в течение 15 секунд (реагирование), то алгоритм шифрования не обязательно должен быть очень надежным. Возможно, существует множество путей для того, чтобы получить эти коды, не вызвав при этом сигнал тревоги. Если коды меняются раз в неделю и это изменение никоим образом не связано с механизмом обнаружения (то есть автоматическое реагирование не предусмотрено), то алгоритм шифрования должен обеспечить защиту кодов в течение недели.

Неразумно надеяться только на защитные механизмы, и прежде всего потому, что часто одна атака может следовать за другой. Использование исключительно защитных механизмов обеспечит безопасность только в том случае, если технологии, лежащие в их основе, совершенны. Если бы наличествовала идеальная система защиты смарт-карт от вторжения, не было бы необходимости в обнаружении и реагировании. Система защиты смарт-карт, существующая в реальном мире, время от времени дает сбои; поэтому хорошо сконструированная система защиты обязана включать в себя механизмы обнаружения и реагирования на случай ее провала. Одна из основных идей этой книги состоит в том, что не существует совершенной технологии. Обнаружение и реагирование, таким образом, весьма существенны.

Представим себе компьютерную сеть. Если брандмауэры, операционные системы, пакеты серверного программного обеспечения абсолютно защищены, тогда нет необходимости в системах сигнализации. Никто не в силах проникнуть внутрь, так что незачем поднимать тревогу. В реальном мире не существует продуктов, у которых нет уязвимых точек. Всегда можно найти способ взломать брандмауэр, разрушить операционную систему, атаковать программное обеспечение сервера. Единственное, что спасет положение в отсутствие совершенных защитных барье¬

ров, это применение контрмер обнаружения и реагирования, чтобы получить вовремя сигнал, если система будет взломана, и иметь возможность противодействовать.

## Ландшафт уязвимых точек

В реальных системах множество уязвимых точек, и существует много различных способов провести атаку. Если террорист хочет взорвать самолет, он может протащить на борт бомбу, сбить его с помощью ракеты или захватить самолет и направить его на ближайшую гору. Хакер, желающий проникнуть в корпоративную сеть, может атаковать брандмауэр, веб-сервер, использовать модемное подключение и т. д.

Системы, действующие в реальном мире, обладают множеством различных возможностей для противодействия атаке. Оборудование авиалиний включает в себя детекторы металла, химические анализаторы и рентгеновские аппараты, позволяющие обнаружить бомбу, и системы, отыскивающие «ничейные» вещи, так что можно быть уверенным, что бесхозный пакет не взлетит вместе с самолетом, в то время как его хозяин остался на земле. (Эта система противодействия предполагает, что немногие террористы захотят взрывать себя вместе с самолетом, а большинство предпочтут спокойно разгуливать по земле, когда самолет будет взорван.) Военные самолеты имеют также системы противоракетной обороны. Корпоративные сети содержат брандмауэры, системы обнаружения вторжения, используют процедуры периодического обновления паролей или шифрования файлов на сервере.

И все это удивительно легко может стать бесполезным.

Я использую термин *ландшафт уязвимых точек,* чтобы изобразить обманчивый и сложный мир атак и мер противодействия. Использованная метафора допускает расширение списка описываемых атак — выстрелы из ружья в банковского кассира, шантаж программиста с целью заставить его включить троянского коня в кусок программного кода, проникновение через стену банка, обработка невнимательного сотрудника для того, чтобы получить пароль, — и контрмеры: пуленепробиваемое стекло, защищающее кассира; проверка всего персонала; камеры наблюдения снаружи здания; биометрический контроль. Различные куски ландшафта связаны с различными типами атак. Компьютерные атаки представляют собой, несомненно, только малую часть ландшафта.

Каждая система имеет свой собственный ландшафт уязвимых мест, хотя многие черты для различных систем схожи. (Каждая компьютеризированная система подвержена угрозе отключения питания. И почти каждая система использует угрозу ареста в качестве контрмеры.) Ландшафт уязвимых точек очень неровен, его составляют пики и долины различной высоты и глубины. Чем выше пик, тем эффективнее соответствующая мера отпора: вершина «используйте пароли» невысоко поднята над землей, а «выключите компьютер и утопите его в вонючем болоте», конечно, намного выше. Долины, с другой стороны, представляют собой узкие места: это возможности потенциальных противников атаковать вашу систему. Чем ниже долина, тем серьезнее изъян.

Уязвимая точка еще не означает выигрыш. Выигрыш — это то, о чем мы говорили в главе 3: выигрыш вора, которому удалось украсть деньги, выигрыш нечестного торговца, присвоившего чужую собственность, выигрыш озабоченного студента, заработавшего дурную славу. Уязвимые места могут быть использованы атакующими, чтобы добиться цели. Цель — украсть деньги; незащищенный кассовый аппарат — уязвимая точка. Испортить чью-либо репутацию — это цель; незашифрованные файлы на его жестком диске — уязвимая точка. Некоторые уязвимые точки бесполезны для достижения именно данной цели. В анонимных группах новостей целью атакующего может быть, например, установление личности корреспондентов. Нехватка идентифицирующей информации не сослужит ему хорошую службу. Если группа новостей построена по принципу платной подписки, у нападающего может быть иная цель — пользоваться ею бесплатно. В таком случае уязвимость системы идентификации будет ему на руку.

Ландшафт уязвимости можно представить различными способами. Я выделяю в нем четыре обширные области: физический мир, мир виртуальный, доверенности и жизненный цикл системы. Они связаны друг с другом. Противник способен действовать на уровне физического мира: взломать дверь и ворваться, забросать бомбами, отнять жизнь и т. д. С помощью Интернета тот же противник может напасть в виртуальном мире: отключить компьютеры и телефонную связь, взломать полицейские компьютеры и поместить ложное объявление о розыске всех членов совета директоров и т. п. Нападения на физические инфраструктуры из виртуального мира могут проводиться на расстоянии, мгновенно и без предупреждения. Они часто оказываются гораздо опаснее нападений на физическом уровне.

## Физическая безопасность

Проблему физической безопасности человечество пыталось решать во все времена: как только возникло понятие собственности. Стены, замки и вооруженная охрана — вот средства физической безопасности. Уязвимыми местами являются, например, не снабженная сигнализацией стеклянная крыша, дремлющая по ночам охрана и замки, которые можно открыть фомкой. Долгое время учреждения имели дело со всеми этими проблемами, и большинство из них научились использовать меры физической безопасности, соответствующие реальной угрозе. Они более или менее представляют своих противников и то, какие контрмеры являются достаточными для защиты их имущества.

Разработчики систем безопасности в цифровом мире часто забывают о физической безопасности. Постоянно похищаются портативные компьютеры, в которых хранятся секреты. За один особенно неудачный месяц 2000 года MI5 и MI6 (британские разведывательные службы) лишились портативных компьютеров с секретными сведениями. Возможно, воры не интересовались этой информацией или она была зашифрована, однако никто этого не знает точно. (Британские вооруженные силы, судя по всему, имеют множество проблем с сохранностью портативных компьютеров. В 1991 году компьютер, содержавший секретные указания в связи с Войной в Заливе, был похищен из автомобиля, принадлежавшего Королевским Военно-Воздушным силам. После того как были организованы широко

освещавшиеся полицейские мероприятия по розыску преступника, компьютер был возвращен с посланием: «Я — вор, а не изменник».) Удивительно много компьютеров крадут в аэропортах организованные шайки воров при прохождении пассажирами детекторов металла.

Меры физического противодействия часто используются совместно, так чтобы они усиливали друг друга, и обычно это бывает более эффективно, чем использование любой из них по отдельности. Охрана обходит прилегающую к запертому зданию территорию, огороженную забором. Банки используют охрану, сигнализацию, видеонаблюдение и сейфы, снабженные замками, срабатывающими в назначенное время.

Когда эти меры предпринимаются в совокупности, ни одна из них не должна обязательно обеспечивать полную защиту от нападения. Требования, предъявляемые к каждому из средств защиты, зависят от того, какие другие меры безопасности задействованы. Дверного замка стоимостью в 5 долларов может быть вполне достаточно при наличии забора и охраны внутри. А замок стоимостью в 50 долларов окажется бесполезен, если поблизости оставлено открытым окно. Отравленные стрелы будут излишни, если на пути злоумышленника установлены вращающиеся стальные лезвия.

## Виртуальная безопасность

Против угроз в виртуальном мире также были разработаны контрмеры. Установка брандмауэра аналогична возведению стен и запиранию дверей. Системы идентификации выполняют функции охраны и напоминают проверку пропусков. Шифрование позволяет создать в киберпространстве «секретную комнату» для конфиденциальных переговоров или электронный сейф для хранения информации.

Хорошая система защиты также использует несколько различных мер безопасности: брандмауэр защищает систему от проникновения посторонних, строгая идентификация дает уверенность в том, что лишь правомочное лицо может войти в нее, а дополнительные гарантии дает шифрование данных при сквозной передаче[1].

## Доверенности

Доверенность определяет, кому и как собственник доверяет распоряжаться своим имуществом или его частью. Например, поступающий на работу должен представить достоверную анкету, представленные им рекомендации должны быть проверены, а его прошлое не должно быть омрачено криминальными эпизодами. Если его приняли, то ему выдается служебное удостоверение с фотографией и свидетельство о праве на парковку. Различным группам людей предоставляется право входить в определенные помещения, открывать доступные им файлы или посещать некоторые собрания. Только определенные особы могут подписывать чеки,

[1] Сквозная передача (end-to-end) означает, что только отправитель и получатель сообщения могут читать передаваемую информацию. Все промежуточные устройства, включая брандмауэр, просто пересылают зашифрованные данные дальше. Это возможно, если протокол (сетевого уровня) поддерживает сквозную передачу, например IPsec. — *Примеч. ред.*

заключать контракты или проводить финансовые операции. При чрезвычайных обстоятельствах дополнительная гарантия безопасности обеспечивается отстранением от обязанностей: например, лицо, обладающее правом подписывать чеки, оказывается лишенным доступа к компьютеру, создающему подписи. Доверенность предполагает сложную структуру отношений. Некто может обладать правом изменять записи базы данных, но не инженерные спецификации. Другой человек будет, наоборот, иметь право изменять эти спецификации, но не иметь доступа к персональным данным.

В реальном мире не составляет труда определить, кто облечен доверием, а кто нет. Вы знаете, как это выглядит. Если иностранец заходит в офис и достает мелкие деньги, это уже вызывает подозрение. До тех пор пока организация настолько мала, что все лично знают друг друга, атака, связанная с физическим проникновением, практически нереальна. Любая другая организация должна обезопасить себя от шпионов: служащие должны отслеживать появление незнакомцев и при этом не думать ни о чем постороннем (в противном случае люди будут уязвимы для угроз, взяточничества, подкупа, шантажа, обмана и других отвратительных проявлений).

В виртуальном мире проблема гораздо сложнее — нужно обеспечить тот же уровень доверия между людьми без физического присутствия заинтересованного лица, имеющего возможность всегда изменить ситуацию. Например, в физическом мире злоумышленник, который хочет притвориться доверенным членом сообщества, рискует, что его могут найти и арестовать. В виртуальном мире шпион, который проник внутрь системы, представляясь доверенным лицом, гораздо меньше рискует быть обнаруженным и пойманным.

## Жизненный цикл системы

В целях промышленного шпионажа можно, например, подключиться к телефонной линии своего конкурента. Затем необходимо рассчитать, как и когда следует провести эту атаку. Оборудование офиса уязвимо в течение всего жизненного цикла: при его проектировании, при сборке, при установке и после того, как оно размещено там, где должно находиться. В зависимости от необходимого уровня доступа нападающий может изменить его свойства на любом из этих этапов. В некоторой точке жизненного цикла советские шпионы поставили «жучки» на пишущие машинки в посольстве Соединенных Штатов. Когда они их поставили? На фабрике в США, во время транспортировки в посольство или во время установки? Мы не знаем точно, но каждая возможность могла быть реализована. И в зависимости от того, насколько серьезной проверке подвергались машинки, «жучки» могли или не могли быть обнаружены.

Точно так же, преступник, который хочет украсть деньги из игрового автомата, имеет выбор: он может, воспользовавшись случаем, внести нужные изменения в схему при установке или взломать автомат, когда тот уже находится в казино. Каждый из этих видов атак имеет свои характерные особенности — сложность, вероятность достижения успеха, рентабельность — но все они допустимы.

Оборудование, использующееся в виртуальном мире — программное обеспечение, работающее на компьютерах, включенных в сеть. Нападающий может его атаковать в любой точке на протяжении всего жизненного цикла. Злонамеренные разработчики программного обеспечения в состоянии сознательно оставить «черный ход» в послед¬

ней версии операционной системы. Злоумышленник способен поместить троянского коня в наиболее популярный браузер и распространять эту программу бесплатно через Интернет. Он может написать вирус, который будет атаковать программное обеспечение при открытии исполняемого файла во вложении сообщения электронной почты. В его силах проанализировать программное обеспечение, использовать все существующие уязвимые точки. Возможности здесь не ограничены.

## Разумное применение мер противодействия

Ландшафт уязвимых точек предоставляет широчайшее поле деятельности для различных видов возможных атак, и поэтому имеет смысл при разработке мер противодействия исходить именно из ландшафта. Идея состоит в том, чтобы защититься от наиболее вероятных угроз, вместо того чтобы защищаться от угроз наиболее явных, игнорируя все остальные.

Не менее важно разумное вложение средств в применяемые меры противодействия. Не имеет смысла тратить кучу денег на самый лучший замок на входной двери, если злоумышленник способен проникнуть через окно. Нерационально тратить 100 долларов на пуленепробиваемое стекло, чтобы защитить имущество, оцениваемое в 10 долларов. Можно сказать, что, например, применять сложные методы шифрования для кабельного телевидения — это то же самое, что «повесить замок на сумку из бумаги».

Ценность чего-либо всегда определяется в зависимости от окружающих условий. До того как стали применяться жесткие диски, подростки иногда воровали в офисах дискеты... поскольку они могли представлять ценность. Некоторые компании потеряли таким образом довольно важные данные. А с другой стороны, маловероятно, что украдут кредитную карточку общей стоимостью около 100 долларов, у которой 0,25 доллара на депозите. Тщательный анализ стоимости очень важен при разработке рациональных мер противодействия телефонному мошенничеству и созданию пиратских копий программного обеспечения.

Следует помнить, что возможный противник далеко не всегда преследует цель обогащения. Иначе как тогда объяснить поведение хакера, который тратит сотни часов на то, чтобы взломать бесполезную компьютерную систему? Многие нападающие стремятся к огласке, или хотят отомстить, или имеют иные, нематериальные цели; помните об этом, когда анализируете ценности.

Также полезно иметь в виду, что блокирования любого из четырех первых шагов атаки достаточно, чтобы ее пресечь. Простые меры противодействия, такие как обучение персонала, грамотная политика безопасности, процедуры, связанные с контролем доступа, являются разумными и рентабельными средствами, позволяющими снизить риск, создаваемый ландшафтом уязвимых точек. Эти простые мероприятия могут значительно повысить сложность и рискованность действий, необходимых для осуществления успешной атаки.

Следующие несколько глав будут посвящены моделированию угроз, оценкам риска и определению необходимых мер противодействия.

# Глава 19. Моделирование угроз и оценки риска

Моделирование угроз — это первый шаг в решении проблемы безопасности. Это попытка осмыслить информацию, которую можно извлечь из ландшафта уязвимых мест. Что может оказаться реальной угрозой для системы? Если вы не знаете этого, как вы можете определить, какие меры противодействия нужно использовать?

Моделирование угроз — трудное дело, и успех в нем приходит только с опытом. Для его осуществления необходимо использовать системный подход и хорошо представлять себе все особенности ландшафта. Как лучше провести нападение на систему? Я нахожу, что истинные хакеры весьма искусны в решении этого вопроса, и может быть, компьютеры их привлекают в первую очередь именно возможностями интеллектуальной игры. Хакеры получают удовольствие, размышляя о недостатках систем: как можно их одолеть, почему это возможно и что при этом будет происходить? Они испытывают наслаждение, заставляя систему делать то, для чего она не была предназначена. Те же чувства испытывает умелец, способный переделать двигатель своего автомобиля, чтобы он работал так, как ему хочется, а не так, как предполагал его производитель. Такое же удовольствие получает хакер, взламывающий брандмауэр через Интернет, чтобы убедиться в том, что он способен «овладеть» чужим компьютером.

Я пришел к выводу, что наилучшими экспертами в области безопасности являются люди, исследующие несовершенства защитных мер. Они идут на избирательный участок, размышляя о том, как можно было бы в обход установленного контроля проголосовать дважды. Когда они пользуются телефонной карточкой, они думают о средствах защиты от мошенников и о том, как можно их обойти. Эти размышления вовсе не обязательно подталкивают их к конкретным действиям, и если они обнаруживают, например, появившееся «слепое пятно» в системе видеонаблюдения в магазине, это не означает, что они тут же предпримут попытку кражи.

Моделирование угроз имеет с описанной ситуацией много общего, и единственный способ изучить проблему — это практика. Начнем с кражи блинов.

Наша цель — поесть бесплатно в местном ресторанчике. Для этого у нас есть много возможностей. Можно поесть и убежать. Можно расплатиться подложной кредитной картой, фальшивыми чеком или наличными. Можно выманить посетителя из ресторана и съесть его блюдо. Можно прикинуться (а то и стать на самом деле) поваром, официантом, управляющим или хозяином (которого видели всего лишь несколько работников). Можно стащить тарелку с чужого столика или из устройства для подогрева, опередив официанта. Можно подождать у мусорного

бака, когда вынесут выбрасывать объедки. Можно включить пожарную сигнализацию и вволю попировать в полном одиночестве. Можно представиться управляющему некоей знаменитостью, могущей рассчитывать на бесплатный завтрак, или найти доверчивого клиента, которого можно уговорить заплатить за нас. Можно ограбить кого-нибудь поблизости от ресторана и расплатиться за еду. Можно подделать талон на бесплатное обслуживание. А кроме того, есть освященная веками традиция — ворваться с ружьем и прокричать: «Гоните сюда все ваши блины!».

Вероятно, существует множество других возможностей, но у нас уже есть общее представление. Взглянув на приведенный перечень, не так трудно понять, что большинству нападающих ничего не нужно делать в тот момент, когда деньги переходят из рук в руки. Это любопытно, поскольку означает, что безопасность системы платежей не защищает от кражи блинов.

Подобная ситуация складывается и в цифровом мире. Представим себе хранилище блинов в Веб — большинству нападающих не придется иметь дело с системой электронных платежей. Существует множество других уязвимых мест. (Вспомните изящное нападение через веб-страницу на корзину для покупок, описанное в главе 10, когда нападающий мог изменять цены товаров произвольным образом. Здесь кроется возможность для подобного нападения: можно изменить прейскурант таким образом, что блин будет стоить 0,00 долларов.) Наиболее успешные нападения редко проводятся на физическом уровне.

## Честные выборы

Перейдем к более значительным и интересным проблемам. Займемся проведением выборов. Это будут местные выборы — мэра города. Жульничество на выборах старо, как сами выборы. Насколько это трудно?

Предположим, имеется дюжина избирательных округов, в каждом из которых есть свой избирательный участок. На каждом участке присутствуют по три члена избирательной комиссии, которые следят за правильностью голосования. Избиратели получают у них бюллетени, которые потом опускают в урну. В конце дня все бюллетени пересчитываются специальной машиной. Избирательные комиссии всех двенадцати участков сообщают по телефону о результатах голосования в центральную комиссию, где подводятся окончательные их итоги, после чего один из претендентов объявляет о своей победе под дождем конфетти и под громкие звуки оркестра.

Эта система имеет множество уязвимых мест. Можно воздействовать на избирателей, на членов комиссии, можно подобраться к урнам для голосования и счетным машинам, можно дать ложное сообщение по телефону или воздействовать непосредственно на центральную комиссию. Давайте рассмотрим каждый вариант в отдельности.

Подкуп избирателей — освященный временем способ проведения выборов. Эта практика теряется во тьме веков и не имеет места лишь в странах третьего мира. Во время выборов 1996 года в городе Додж Кантри, Джорджия, насчитывающем 17 000 жителей, 21 человек был уличен в различных махинациях при голосовании, в том числе в подкупе избирателей. В большинстве штатов (включая Джорджию)

закон запрещает платить избирателям, поэтому политики вынуждены прибегать к другим ухищрениям: они обещают снижение налогов, организацию общественных работ, лоббирование законопроектов и благосклонность Белого Дома. Это эффективный способ, хотя и дорогостоящий.

И на него не стоит полагаться. Использование закрытых кабин для голосования делает невозможным прямой подкуп избирателей. Можно заплатить по 100 долларов каждому из них, чтобы они проголосовали за нужного кандидата, однако войдя в кабину для голосования, они могут отдать свой голос кому пожелают. (Снижение налогов в этом отношении эффективнее, особенно если речь идет о находящемся в должности претенденте: избиратели думают, что, голосуя за него, они добьются еще больших поблажек.) Есть старая история про чикагского политика, скупавшего голоса. Его подручные пачкали черной краской рычаги для голосования[1], соответствовавшие его имени, и получали возможность убедиться в том, что избиратели голосовали именно за него.

Эти виды мошенничества используются также и в том случае, когда применяется преимущественно голосование по почте. В Силиконовой долине от трети до половины всех бюллетеней направляется в избирательную комиссию по почте. В Аризоне даже проводился эксперимент по голосованию через Интернет на предварительных выборах Демократической партии 2000 года. В этом случае опасность состоит в том, что некто может отправиться в бедные кварталы города и скупить целый пакет чистых бюллетеней по 10 долларов за каждый (в Аризоне использовались идентификационные номера (PIN), которые также можно «скупить») — и обитатели этих кварталов будут довольны.

Правящая партия Сингапура нарушает тайну выборов своеобразным способом: территории, на которых проводится голосование, имеют крошечные размеры — вплоть до одного многоквартирного дома. Проконтролировать, как проголосовал отдельный избиратель, невозможно, однако власти откровенно лишают государственного финансирования те районы, жители которых голосуют за оппозицию. Это — вид массового подкупа избирателей.

Предположим, подкуп избирателей нам не по средствам, а кроме того, нас беспокоит, что кто-нибудь может разоблачить в газете наши махинации. А что, если взять избирателя на испуг? Фокус, проделанный мексиканской Институциональной революционной партией, состоял в том, что в отдаленных районах урны для голосования, непроницаемые для любопытных взглядов, устанавливали под деревом, в ветвях которого скрывался головорез, следивший за тем, чтобы избиратели голосовали «правильно».

Можно попытаться одурачить избирательную комиссию. Например, нанять группу артистов, которые будут изображать законных избирателей. Можно сделать так, чтобы некоторые избиратели проголосовали не один раз. Это все действенные методы, но существует защита от них. В Соединенных Штатах члены избирательной комиссии имеют списки законных избирателей, с которыми они сверяются при голосовании и делают соответствующие пометки. На первых общих (нерасовых) выборах в Южной Африке в 1994 году избирателям на руку на-

[1] В Соединенных Штатах процесс голосования давно был автоматизирован, и избиратель вместо заполнения бюллетеня должен нажать соответствующий рычаг в кабине для голосования. — *Примеч. перев.*

носили клеймо несмываемыми чернилами, чтобы лишить их возможности проголосовать дважды. На первых в постсоветский период выборах в Латвии (1990 год) проверяли удостоверения личности, в которых делали отметки о голосовании. Во время выборов 1999 года в Индонезии избирателей заставляли окунать пальцы в чернила. (Предполагалось, что чернила будут держаться все три дня, отведенные для голосования, однако некоторые обнаружили, что краситель смывается.)

Можно направить свои усилия непосредственно на членов избирательной комиссии. Если иметь их своими союзниками, нетрудно добиться потрясающих результатов. Можно внести в списки избирателей кого угодно — в начале двадцатого века многие умершие жители Чикаго регулярно участвовали в выборах — или просто подделать результаты голосования. Во время президентских выборов 1960 года чикагские демократы под руководством мэра Ричарда Дейли, жуткого типа, вовсю мошенничали, манипулируя голосами избирателей в пользу Кеннеди, и в результате Никсон эти выборы проиграл. (Когда республиканцы потребовали провести пересчет голосов в этом штате, демократы в свою очередь потребовали пересчета голосов в других штатах, и, в конце концов, обе стороны сдались.) Подобные вещи имеют место по сей день: на выборах 1996 года в Сенат в штате Иллинойс организация Демократической партии была обвинена в подкупе избирателей, а именно в том, что избиратели голосовали по несколько раз и даже имели место манипуляции с машинами для голосования.

Несмотря на широко распространенную коррупцию в избирательных комиссиях, мошенничество на выборах становится все более трудным делом. Можно попытаться подкупить членов избирательных комиссий, чтобы они закрывали глаза на некоторые вещи, но беда в том, что на каждом участке их трое. Или подкупить одного из них, но нет никакой гарантии в том, что двое других будут столь же сговорчивы. Расходы в этом случае даже превышают те, что связаны с подкупом избирателей, а вероятность того, что средства массовой информации будут поставлены на ноги, гораздо больше.

Что вы скажете насчет урн для голосования? Можно наполнить их фальшивыми бюллетенями, в этом и состоит основная идея мошенничества. Однако важно не перестараться так, чтобы проголосовали, скажем, 130 процентов избирателей. И нужно быть уверенным, что никто ничего не заметит: в некоторых странах третьего мира используют прозрачные избирательные урны, чтобы предотвратить подобные махинации.

Атаковать машину для подсчета голосов еще проще. Это компьютеризированное устройство, и если вредоносная программа сделает так, что не будут учтены голоса за одного из кандидатов, скорее всего этого никто не заметит. Можно попытаться внедрить троянского коня в программный код во время его написания (предполагается, что машина не просто механически подсчитывает голоса, а для этой цели имеется специальная программа). Или же улучить момент, когда никто не работает с машиной, и ввести вредоносный код. Можно одурачить избирательную комиссию, изобразив дело как установку новой версии программы. Существует масса возможностей для такого рода мошенничеств.

Можно намеренно внести опечатки в избирательные бюллетени или сдвинуть рамку на доли дюйма таким образом, что иногда машина не будет учитывать голоса, поданные за оппозицию, и этого никто не заметит. Еще способ: привести маши¬

ну в негодность; тогда избирательная комиссия будет вынуждена подсчитывать голоса вручную. И тогда подкупленный член комиссии может попытаться сфальсифицировать результат. На президентских выборах в Мексике в 1988 году компьютер «отказал» как раз в тот момент, когда лидировал один из претендентов на президентское кресло. Когда он снова заработал, оказалось, что победил действующий президент... и после этого избирательные бюллетени были быстренько сожжены. Я не хочу бросать тень на мексиканскую избирательную систему, но все это выглядит очень подозрительно.

Центральная избирательная комиссия — наименее подходящее для мошенничества место, так как она у всех на виду. Возможно, злоумышленнику удастся представить неверные сведения по районам, но один из членов избирательной комиссии может это обнаружить. Телефонные переговоры между окружными пунктами и центральным офисом — может быть, этот элемент системы ему удастся использовать в своих целях.

Итак, что же предпочесть? Кажется, наиболее верный путь к успеху — склонить большинство членов избирательной комиссии к тому, чтобы они действовали согласно нашим указаниям. В их власти прибавить и убавить голоса, подменить избирательные урны во время транспортировки; у них есть еще масса других возможностей. Тогда у нас будет доступ к машине для подсчета голосов, в наших силах привести для голосования мнимых избирателей или подбросить в урны фальшивые бюллетени. Вывод, однако, таков: все это осуществить достаточно сложно. Если люди, входящие в состав избирательной комиссии, не зависят полностью от одного из кандидатов — такая ситуация часто встречается в странах третьего мира, но редко в Соединенных Штатах, — сделать это практически невозможно.

Цель этого мысленного эксперимента — показать, что существует множество путей одолеть избирательную систему, и лишь в незначительной степени это имеет отношение к компьютерным системам. Можно взломать программу или вызвать отказ в обслуживании и тем самым вынудить членов избирательной комиссии вернуться к старой, гораздо менее надежной системе подсчета голосов. Но в конечном счете результаты выборов близки к истине. Если люди в избирательной комиссии заслуживают доверия, выборы, скорее всего, «чистые». Если они не заслуживают доверия, существует такое великое множество способов повлиять на исход голосования, что не стоит даже задумываться, какой из них наиболее предпочтителен.

Интернет вносит свои поправки в эту запутанную схему, и опасность значительно возрастает. Все старые способы мошенничества остаются в силе, но появляется масса новых возможностей: атаки против компьютеров в избирательных участках, атаки через сеть, атаки против компьютеров избирателей (которые в любом случае ненадежны). И нападения, приводящие к отказу в обслуживании, которые не могли бы быть проведены против централизованной системы. Что еще хуже, современная избирательная система не дает шансов исправить ситуацию в случае успешного нападения. В 2000 году в Аризоне на предварительных выборах было разрешено голосование через Интернет. Если бы возникли проблемы или подозрения, что проблемы существуют, что бы стали делать в Аризоне? Отменять выборы и переносить их на неделю? По этой причине ни один специалист по выборам не посоветует пользоваться услугами Интернета для голосования.

# Защита телефонов

Это должно быть просто. Организация — правительство, корпорация, группа защитников прав человека — нуждается в том, чтобы обезопасить от прослушивания свои телефонные переговоры. Решение, естественно, состоит в том, чтобы использовать шифрование. Но откуда следует ждать угрозы?

Противником может являться фирма-конкурент или правительство, некто, кто обладает необходимыми ресурсами и доступом, чтобы провести сложную атаку. Чтобы решить проблему, организация строит или покупает телефонную линию, в которой используется шифрование.

Как атаковать эту систему? Для этого необходимо взломать код, но предположим, что это нам не под силу.

Можно попытаться сделать так, чтобы телефоны работали неправильно. В системе существует много параметров: можно ослабить алгоритм шифрования, можно внести путаницу в систему генерации ключа, можно настроить телефон так, что он будет делать незашифрованные звонки, или можно использовать скрытый канал для того, чтобы выудить информацию о ключе, анализируя цепь передачи звуковых сигналов (когда эта операция была проведена открыто, она получила название «Клиппер»). Все эти нападения осуществимы во время создания оборудования, во время транспортировки или во время эксплуатации. Такие атаки можно провести, проникнув ночью на фабрику, подкупив того, кто там работает, или просто поменяв некоторые скрытые установки в момент запуска оборудования.

Это может показаться нереальным, но если вы обладаете ресурсами государственной разведывательной организации, все это — совершенно разумные методы проведения атаки. Швейцарская компания Crypto AG поставляла шифровальное оборудование правительствам многих стран третьего мира. В 1994 году один из ее руководителей был арестован в Иране за установку негодного криптографического оборудования. Когда спустя несколько лет он вышел из тюрьмы, он опубликовал материалы о том, что их компания годами вносила изменения в свою продукцию по заказу разведки США. В пятидесятые годы компания Xerox продавала в Россию копировальные устройства, в которых была спрятана небольшая фотокамера, тот, кто производил ремонт этих устройств, должен был всего-навсего периодически заменять в ней пленку.

«Советы» были не менее коварны: в Москве они установили жучки в американском посольстве во все оборудование, включая пишущие машинки IBM Selectric. Британские компании, занимающиеся шифрованием, славятся тем, что поставляют иностранным правительствам продукт, в котором предусмотрены специальные возможности для овладения чужими секретами. Даже если бы об этом не шла молва, наверняка аргентинское правительство подумало дважды, прежде чем использовать шифровальные устройства английского производства во время войны за Фолклендские острова.

Есть бездна возможностей для прослушивания, которые невозможно предотвратить за счет мер безопасности, применяемых в телефонной связи: можно установить жучок в самом защищенном телефоне (или просто в комнате, где стоят эти телефоны), подкупить людей, делающих или принимающих звонки, и т. д. И не приходится рассчитывать на то, что подобные проблемы удастся разрешить с помощью технических средств.

Одним из лучших методов нападения является просто выведение телефона из строя. Это легче сделать владельцу телефонной системы: например, когда телефоном пользуется правозащитная организация в проблемной стране третьего мира или когда международная корпорация пытается связаться со своим филиалом в развитой стране, славящейся промышленным шпионажем. Чтобы подслушать чужие разговоры, достаточно вызвать сбои в работе защищенного телефона. Скорее всего, собеседники продолжат общаться по обычному телефону и скажут все, что собирались сказать.

## Безопасность электронной почты

Защита электронной почты — несколько более интересная тема. В главе 12 я вкратце изложил принципы работы безопасных почтовых программ. С помощью криптографии можно достичь двух вещей: создать цифровую подпись для идентификации и обеспечить тайну переписки. (История такого средства защиты, как конверт, весьма любопытна. Вавилоняне запекали свои таблички в глиняные «конверты». Бумажные конверты впервые стали использовать китайцы — зачастую с восковыми печатями, дававшими дополнительные гарантии, — и такие конверты завоевали Европу, где они приобрели особенную популярность благодаря стараниям Людовика XIV.)

В любом случае существует уйма способов нападения на почтовую систему. Взять хотя бы криптографию: работают ли алгоритмы и протоколы так, как предполагали их разработчики? Или реализацию: нет ли в программном обеспечении таких ошибок, которые можно использовать? Здесь существуют все те «черные ходы», которые используются против безопасных телефонов: нельзя ли изменить программу на стадии разработки и усовершенствования или когда она уже находится в распоряжении пользователя? А что можно сказать о паролях, вводимых пользователями, чтобы прочесть зашифрованную корреспонденцию или подписаться под посланием? Почтовые программы проходят сертификацию, это делается для того, чтобы обосновать использование открытого ключа, но в главе 15 мы уже обсуждали возможные уязвимые места в модели безопасности системы сертификации. И нельзя забывать обо всех других уязвимых точках, которые не имеют отношения к системе электронной почты: можно подглядеть и прочитать чужое письмо перед его отправкой или по получении, сделать копию распечатки — все это не зависит от того, каким механизмом депонирования ключей государство (или компания) имеет глупость вынуждать пользоваться.

Использование шифрованной корреспонденции сопряжено с большим риском, чем в случае шифрования телефонных переговоров, когда опасность утечки информации существует только во время звонка. Поскольку корреспонденция может некоторое время храниться обеими сторонами, такая опасность остается всегда. Кроме того, нападающий способен взломать операционную систему используемого для переписки компьютера, а для телефонных переговоров применяется специальное оборудование, к которому намного труднее подобраться. Противник в силах с минимальным риском провести нападение на расстоянии и завладеть всей интересующей его информацией, а не одним только письмом. Наконец, нападение

может быть автоматизировано так, чтобы оно было направлено сразу на многие различные цели или просто дожидалось своего часа.

К обсуждению этого примера мы вернемся в главе 21.

## Смарт-карты «электронный бумажник»

Следующий, более сложный пример: электронная система платежей, основанная на использовании смарт-карты, на которой ведется баланс средств. (Их часто называют «электронным бумажником» — stored-value cards.) Некоторые такие карты уже опробованы на практике: система Mondex (а также MasterCard), VisaCash (испытана во время летних Олимпийских игр 1996 года в Атланте), Banksys's Proton. Мы рассмотрим некую абстрактную карту, не вникая в детали этих систем. Назовем нашу гипотетическую систему Plasticash.

Основная идея Plasticash состоит в том, чтобы использовать ее в денежных расчетах. Специальные терминалы станут неотъемлемой частью деловой жизни: они появятся в банках, в магазинах, будут присоединены к компьютерам, подключенным к Интернету. Когда покупатель захочет купить что-либо (или, в более общем смысле, просто перевести деньги кому-нибудь), они с продавцом вставят свои карты Plasticash в устройство чтения и записи и переведут деньги. (У продавцов, возможно, будут специальные карты, постоянно находящиеся в считывающем устройстве.) В банке или с помощью банкоматов можно будет как пополнить средства, находящиеся на карте, так и перевести их с карты на банковский счет. Обратите внимание: двум картам не обязательно находиться рядом друг с другом, достаточно соединить их по телефону или с помощью модема.

Такие карты обладают тем преимуществом, что они не обязательно должны работать в режиме онлайн, то есть находиться на связи с каким-либо центральным сервером где бы то ни было. (При использовании обычных платежных карточек торговый автомат обязан связаться с банковским компьютером в режиме реального времени.) Их недостаток состоит в том, что утрата или повреждение карты означают потерю денег.

Plasticash, как и всякая другая система электронных платежей, должна будет иметь полный набор средств защиты и использовать криптографию, меры компьютерной безопасности, средства защиты от подделки, возможности контроля и что угодно еще. Она будет обеспечивать необходимый уровень целостности данных, конфиденциальности, анонимности и т. д. Мы не будем вдаваться в подробности. Давайте рассмотрим варианты нападений на подобную систему в принципе.

В функционировании системы Plasticash участвуют три стороны: клиент, продавец и банк. Поэтому существуют три схемы взаимодействия между участниками расчетов:

- **Банк — клиент.** Клиент снимает деньги на свою карту.
- **Клиент — продавец.** Клиент переводит деньги со своей карты на карту продавца.
- **Банк — продавец.** Продавец вносит полученные деньги на свой банковский счет.

В первой части этой книги рассказывается о преступлениях, которые, возможно, будут совершаться и в сфере Plasticash: это кража денег, ложное обвинение, нарушение тайны частной жизни, вандализм и террор, а также разглашение сведений. Система Plasticash должна предусматривать меры противодействия использованию ее для подготовки других преступлений, таких как отмывание денег. Приготовления к преступлению трудно определить точно, представления об этом могут меняться при каждом пересечении границы государства и даже после каждых новых выборов. Также неясно, насколько законы и обычаи различных стран способны противоречить друг другу. Например, на международной арене принятые в США требования к финансовой отчетности могут восприниматься как нарушение швейцарских законов, охраняющих тайну банковской деятельности.

Когда мы говорим о мошенничестве со стороны банков, мы не обязательно имеем в виду, что речь идет о банковской «империи зла». Такие преступления могут совершаться отдельными мошенниками, работающими в заслуживающем уважения банке. Вообще, мы чаще имеем дело с мошенничеством клиентов и продавцов (отдельных мошенников, принятых на эту работу), поскольку теоретически банки в силах позволить себе лучшие механизмы и меры безопасности, а возможные потери в связи с ущербом репутации от нападений на банковские системы очень велики.

Итак, первый вид преступлений — это кража. Существует несколько способов похитить деньги с Plasticash. Ими могут воспользоваться как клиенты, так и продавцы:

- Взломать карту, чтобы прибавить себе денег. Это также можно осуществить несколькими способами, наиболее очевидный путь — добраться до устройства, регистрирующего находящуюся на карте сумму.
- Можно таким же образом увеличить или уменьшить размер платы за покупку.
- Можно научиться создавать или имитировать новые карты и изготовить фальшивые Plasticash, которые будут функционировать, как настоящие. Фальшивая карта не обязательно даже должна быть похожа на оригинальную: мошенник может пользоваться ею только при покупках в Интернете или переводить деньги с фальшивой карты на настоящую, с которой и будет производить платежи.
- Можно научиться клонировать карты. Мошеннику понадобится завладеть на время настоящей картой, чтобы сделать клон, после чего ее можно будет вернуть владельцу. (Успешные преступления такого рода уже совершались с канадскими банковскими карточками, и в 1999 году некоторые из мошенников были арестованы. Жулик-продавец ухитрялся в считанные секунды клонировать карту, использовавшуюся покупателем для расчетов.)

Вот способы мошенничества, используемые клиентами:

- Отрицать сделанные покупки. Это старый прием, состоящий в том, чтобы приобрести что-либо дорогостоящее, заявить о краже карточки и опротестовать произведенные платежи. Есть новый вариант этого метода — прикинуться дураком в свое оправдание, заявляя, например, следующее: «Visa виновата в том, что я потерял все свои деньги в азартных играх онлайн; это произошло не по моей вине». Такое мошенничество проводится на закон¬

ном административном уровне. С чем бы мы ни имели дело — с чеками, кредитными картами, дорожными чеками и чем угодно еще, — всегда найдутся люди, не желающие платить по счетам и утверждающие, что они не расходовали этих денег.

- Можно договориться с кем-нибудь о том, что он заявит о похищении своей карты, и получить ее дубликат. Такого рода жульничества также распространены во многих сферах и не ограничиваются системой Plasticash.

Виды мошенничества, доступные лишь продавцам:

- Принять платеж и отказаться передать покупку. Защита карты не будет препятствием для такого обмана, предотвратить его можно лишь с помощью административных мер и установленных законом процедур.
- Получить доступ к карте покупателя и произвести серию несомненно действительных переводов на свой банковский счет. Это очевидный способ мошенничества. Преступник попытается перевести деньги и затем быстренько снять их со счета.
- Повторить перевод денег. Продавец может каким-либо способом сделать так, что покупатель заплатит двойную цену.

Наконец, мошенничества, доступные банкам:

- Отказаться перевести в Plasticash сумму, которую внес клиент. Противодействовать этому можно только с помощью административных мер и регистрации банковских операций: клиент сумеет доказать незаинтересованной третьей стороне, что его обманули, если будет правильно организовано ведение записей.
- Прикарманить деньги, переведенные клиентом с Plasticash для занесения на его счет. Иначе как административными методами противодействовать этому нельзя.

Все эти мошенничества могут совершаться также в сговоре двух (или трех) сторон. Трудно представить какой-либо другой вид надувательства, которое могут совершить вместе покупатель с продавцом, но в зависимости от степени защищенности системы вероятны такие виды преступлений, которые будут успешны, если они действуют сообща, и обречены на провал при попытке осуществить их в одиночку. Кроме того, нетрудно представить себе преступления, совершаемые людьми, прикидывающимися обслуживающими терминал лицами или телефонистами.

Второй вид преступлений представляет собой ложное обвинение (клевету или шантаж). Их способны совершить как покупатель, так и продавец:

- Покупатель может заявить, что у продавца недействительная карта Plasticash (или неправильно функционирует терминал). То же самое может заявить и продавец относительно карты покупателя. Это мошенничество следует пресекать административными мерами.

Также вероятен шантаж со стороны банка:

- Можно подделать карту клиента (или продавца) и выдвинуть против него ложное обвинение. Весьма правдоподобно, что если банк выпускает карты, то для него не проблема и подделывать их. Приятно ли будет клиенту узнать, что в его карте числятся расходы на проституток?

Третий вид преступлений — это нарушение тайны частной жизни. Это происходит, когда кто-либо сообщает третьей стороне конфиденциальную информацию о некотором лице без его согласия. В зависимости от местного законодательства такие действия не везде считаются преступлением. Если разработчики Plasticash хотят, чтобы система распространилась по всему миру, имеет смысл составить перечень этих действий и не обращать на них внимания, если они считаются законными (и не причиняют никому вреда).

До тех пор пока система не станет обладать средствами для предотвращения нарушения тайны частной жизни, банки будут иметь неограниченные возможности для получения информации о расходах клиентов. («Я знаю, что Вы приобрели прошлым летом».) Этого можно избежать в некоторых случаях (но только в некоторых), если клиенты будут приобретать карты с фиксированной суммой денег на них, аналогично некоторым телефонным картам с предоплатой.

Продавец не сумеет непосредственно получить подобные сведения и узнать имя покупателя, однако с помощью других продавцов он может собрать информацию об использовании карты с известным ему идентификационным номером и, сопоставив данные, идентифицировать ее владельца.

Наконец, следует помнить и о возможности подслушивания: люди, вовсе не участвующие во взаиморасчетах, могут подслушивать и собирать информацию.

Четвертый вид преступлений, вызывающих беспокойство, включает вандализм и терроризм. Эти правонарушения в первую очередь направлены против системы в целом, хотя могут совершаться и против отдельных владельцев карт, продавцов и банков. Главная цель таких преступлений — помешать правильному функционированию системы. То, что называется атаками, направленными на отказ в обслуживании, в этом случае может оказаться весьма любопытно. Судите сами.

Действия, направленные против продавца:

- Можно создать помехи во взаимодействии с банком или покупателями.
- Можно объявить карту продавца похищенной или скомпрометированной.
- Можно физически повредить или уничтожить карту продавца.
- Нарушить электрическое питание терминала или разорвать телефонную связь с ним.

Действия, направленные против клиента (покупателя):

- Можно создать помехи во взаимодействии с банком или продавцами.
- Можно объявить карту продавца похищенной или скомпрометированной.
- Можно физически повредить или уничтожить карту клиента (покупателя).

Действия, направленные против банка:

- Можно создать помехи во взаимодействии с клиентами или продавцами.
- Можно физически повредить или уничтожить банковские технические средства, обеспечивающие безопасность.

Действия, направленные против системы в целом:

- Заставить систему самосовершенствоваться, прежде чем кто-либо поймет, что с этим делать. (Можно рассматривать это как аналог «проблемы 2000 года».)

- Вызвать отказ в обслуживании многих или вообще всех банков.
- Создать помехи во взаимодействии с клиентами или продавцами.
- Уничтожить открытый ключ высшего уровня сертификации в системах, основанных на PKI.

Преступные действия способны также дестабилизировать работу системы:

- Можно наладить массовый выпуск поддельных карт.
- Можно организовать массовые, широко распространенные мошенничества с картами и подорвать доверие к системе.

Наконец, поговорим об использовании системы для совершения преступлений, то есть о нарушениях закона с ее помощью. До сих пор мы рассматривали лишь возможность отмывания денег, но не менее заманчиво обсудить возможности других противозаконных действий. (Заметим, что большинство злодеяний связаны с передачей наличных денег из рук в руки. Наша система карт позволит легко избавиться от таких преступлений, как торговля наркотиками, незаконные азартные игры, проституция и т. д.)

Некоторые люди получают банковские карточки под вымышленными именами, но нетрудно склонить кого-либо к тому, чтобы он использовал свое настоящее имя. (Несомненно, в мире найдется много желающих открыть банковский счет, который, как они понимают, будет контролироваться другими людьми и использоваться для отмывания денег, если им предложат несколько тысяч долларов или, в отдельных случаях, возможность провести несколько дней или недель в пьяном или наркотическом угаре.) Если на такие карточки положить деньги, их можно использовать как компактное платежное средство, и не существует очевидного способа воспрепятствовать этому.

Обратите внимание на то, что решение вопросов морали и законности в этой сфере далеко не очевидно. Требования о предоставлении финансовой отчетности в государственные органы США и Великобритании могут причинять некоторые неприятности гражданам, но власти редко злоупотребляют этим. Во многих других странах, таких как Китай, Турция. Мексика или Сирия, дело принимает совсем другой оборот. Последнее обстоятельство чревато политическими и юридическими проблемами для тех компаний, которые обязаны предоставлять такие сведения, и способно привести к большему распространению мошенничества в этих странах.

## Оценка рисков

Недостаточно просто составить перечень угроз, необходимо знать, как реагировать на каждую из них. Здесь на помощь приходит оценка рисков. Основная идея состоит в том, чтобы оценить возможный ущерб от реализации угрозы и возможное число таких случаев в течение года, а затем вычислить ожидаемые потери за год.

Например, «планируемые» убытки от хакерского вторжения в сеть составляют 10 000 долларов в каждом случае (эта сумма включает в себя оплату труда тех, кто будет обнаруживать такие происшествия, приводить все в порядок и т. д.), а такие происшествия могут случаться трижды в день или тысячу раз в год. В таком слу¬

чае ожидаемые убытки за год не превзойдут 10 миллионов. (Можно понять, что из этого следует. Если ожидаемые убытки за год составляют 10 миллионов долларов, то приобретение, установка и поддержка брандмауэра за 25 000 в год — весьма выгодное дело. Приобретение нечто такого «супер-пупер-умопомрачительного» за 40 миллионов будет пустой тратой денег. При этом анализе предполагается, что брандмауэр и «супер-пупер-умопомрачительное» одинаково хороши для предотвращения угрозы. Позже мы еще вернемся к этой теме.)

Иногда вероятность реализации угрозы очень мала. Если речь идет о вторжении в систему конкурента с целью получения сведений о новых разработках, потери могут составить, например, десять миллионов долларов в каждом случае. Но предположим, что число таких вторжений составляет 0,001 или 0,1 % в год. Таким образом, ожидаемые убытки за год превращаются в 10 000 долларов, и меры противодействия, которые обходятся в 25 000 долларов, становятся совершенно невыгодными.

Страховые компании постоянно имеют дело с оценкой рисков, исходя из нее они и определяют размер страховых взносов. Они считают ожидаемые убытки за год для каждого случая, прибавляют расходы на свою деятельность плюс некоторую прибыль и таким образом получают сумму взносов при страховании от определенных рисков.

При этом, разумеется, им приходится делать множество предположений; конкретные риски, о которых мы говорим, еще слишком новы и трудны для понимания. Иногда требуется особенная проницательность для того, чтобы обнаружить вектор катастрофического развития событий, когда незначительная ошибка чревата многомиллионными потерями.

Для анализа рисков, связанных с компьютерным миром, производители предлагают множество алгоритмов и методов. Однако они в большей степени предназначены для оценки крупных рисков, таких как промышленный шпионаж, а не незначительных опасений вроде взлома шифровального ключа электронной почты.

Анализ рисков важен в том отношении, что позволяет сориентироваться в этой сфере. Огромные зияющие «дыры» в системе безопасности не страшны, пока вероятность реализации угрозы равна нулю. (Токио, например, до сих пор беззащитен перед огнедышащими драконами.) Маленькие «щели» обязательно нужно затыкать, если ежедневно через них может осуществляться десять миллионов нападений.

# Сущность моделирования угроз

При разработке системы безопасности жизненно необходимы как моделирование угроз, так и оценка рисков. Слишком многие разработчики систем представляют себе свою деятельность наподобие поваренной книги: смешаем в определенных пропорциях некоторые меры противодействия — хорошими примерами тому являются шифрование и брандмауэры, — и как по волшебству мы окажемся в безопасности.

Так никогда не бывает. Йоги Берра сказал: «Будьте осторожны, если вы не знаете, куда идете, может быть, лучше вам не попадать туда». Часто системы безопас¬

ности оказываются неспособны противостоять некоторым угрозам. Шифрование электронной почты может спрятать от посторонних глаз содержание корреспонденции, но никак не сумеет скрыть факт существования переписки. В некоторых случаях выявление корреспондентов оказывается более опасным для них, нежели знание содержания писем. В других ситуациях информация о том, что некто использует шифрование, оказывается чрезвычайно содержательной сама по себе.

Хорошая разработка получается в результате последовательного движения от технических требований к нахождению правильного решения, а не в результате применения сухой технологии для получения конечного продукта. В случае разработки систем безопасности это означает, что сперва необходимо заняться моделированием угроз, выработать политику безопасности и только после этого выбирать подходящие технологии. Угрозы определяют политику безопасности, а она, в свою очередь, — процесс разработки. В частности:

- **Следует понять, что реально угрожает системе, и провести оценку рисков.** Это легче сделать, если использовать опыт «реального мира» и знания об имевших место нападениях на похожие системы.
- **Определить политику безопасности для противодействия этим угрозам.** Это должен быть ряд положений вроде: «только уполномоченные банки вправе изменять баланс на картах Plasticash» или «все движения денежных средств в системе Plasticash должны быть доступны контролю».
- **Разработать меры противодействия, которые воплотят в жизнь политику безопасности.** Эти контрмеры должны представлять собой объединение механизмов защиты, обнаружения и реагирования.

Конечно, такая прямолинейная модель создания решения — это идеал, а реалии жизни не часто помогают в ее реализации. Более правдоподобно, что путь разработки будет напоминать спираль, и придется не один раз повторить эти три шага, с каждым разом все более и более приближаясь к достижению истинной безопасности. В наивысшей степени сказанное относится к новым системам и новым технологиям, когда действительные угрозы остаются неизвестны до тех пор, пока на практике не удастся определить, кто и на что будет нападать. Поэтому все хорошие системы предусматривают план действий в непредвиденных обстоятельствах и способы восстановления после катастрофических событий.

# Ошибки в определении угроз

Рассмотрение целей и методов нападающих кажется очевидным делом, однако многие организации, ведущие себя **вполне** разумно в других случаях, оказались неспособны сделать это. Военная контрразведка США потратила многие годы на то, чтобы построить защиту от одной хорошо финансируемой организации, имевшей единственную цель — прослушивание американских линий связи военного значения. Она преуспела в этом, однако совершенно упустила из виду опасность, исходящую от хакеров. Хакеров не интересует прослушивание. Их никто не финансирует. Они не организованы. Им не нужны военные секреты, им хочется поковыряться в системе ради развлечения и посмотреть, как она обрушится. Им хо¬

чется похвастаться перед приятелями и, может быть, увидеть свое имя в газетах. Некий сотрудник AT&T Bell Labs обнаружил дефект в реализации «Клиппер-чипа» (Clipper chip[1]) военной контрразведки и создал ей дурную славу. Зачем? Ради удовольствия поймать контрразведку на ошибке.

Если вы занимаетесь моделированием угроз, сперва обратите внимание на те случаи, когда люди глубоко заблуждались насчет реальности угрозы.

- В индустрии сотовых телефонов было потрачено много денег на разработку средств обнаружения мошенничества, но никто не понимал реальной угрозы. Предполагалось, что преступники будут пытаться пользоваться услугами телефонной связи бесплатно. В действительности все, что требовалось настоящим преступникам, — это анонимность, им не хотелось, чтобы телефонные звонки вели к ним. Номера сотовых телефонов перехватывались в эфире, использовались несколько раз и затем забывались. Система, имевшая целью предотвращение мошенничества, не была рассчитана на обнаружение таких действий.
- В той же индустрии сотовых телефонов, еще в давние времена аналоговой передачи сигналов, никто не беспокоился о безопасности связи, поскольку, как считалось, «сканеры дороги и редко встречаются». Спустя годы сканеры стали дешевы и широко распространены. Тогда, следуя замечательной традиции ничего не предпринимать, никто не побеспокоился о безопасности цифровых телефонов, поскольку «цифровые сканеры дороги и редко встречаются». Что же дальше? Они так же подешевели и стали более распространены.
- Хакеры часто предлагают на продажу средства взлома на веб-страницах и досках объявлений. Некоторые из этих хакерских средств сами заражены Back Orifice, и тот, кто их написал, получает доступ к компьютеру использующего их хакера. Аристотель называл подобное стечение обстоятельств «поэтической справедливостью».
- Когда обнаруживается уязвимость протокола безопасности Интернета, поставщик обычно пересматривает протокол с целью ее уменьшения. Однако ввиду важности обратной совместимости новый протокол часто делается совместимым со старым, уязвимым протоколом. Умный нападающий просто взламывает старый протокол и использует его уязвимость. Это называется *атакой на откат версии (version-rollback)*.
- Несколько лет тому назад монетоприемники японских автоматов для игры в пачинко были заменены устройствами для чтения магнитных карт. В системе использовались разные меры защиты от жульничества, но разработчики ошибались, считая владельцев игорных заведений хорошими парнями. В действительности многие из них были вовлечены в организованную преступность. И модель безопасности была построена плохо: владельцы помещений, где находились автоматы, не оставались в накладе и получали доходы независимо от того, настоящими или фальшивыми были карты, поэтому у них не было стимула обращаться в полицию по поводу

[1] См. главу 16. — *Примеч. перев.*

мошенничества. (Разработчики также полагали, что, установив для каждой карты предел в 100 долларов, можно ограничить потери.) Махинация была проведена тонко — она включала восстановление настоящих карт, «исчезновение» многих автоматов после землетрясения в Кобе и охватывала множество игорных заведений, — и общая сумма ущерба от нее составила около 600 миллионов долларов. По слухам, деньги утекли в Северную Корею.

- Производители автоматов, принимающих монеты, давно предвидели возможность манипуляций мошенников с механическими устройствами. Например, просверливание отверстий в автомате для воздействия на механизм барабана или использование устройств, воздействующих на счетчик монет, подлежащих выплате игроку. Несколько лет назад один из производителей автоматов для игры в покер был удивлен совершенно неожиданному способу воздействия с помощью статического электричества. Некоторые игроки обнаружили, вероятно, случайно, что если накопить достаточно большой заряд, походив по плюшевым коврам в казино, и разрядиться на автомат, из него посыплются все имеющиеся монеты.
- В конце 1999 года было взломано шифрование цифровых видеодисков (DVD). Если даже диск зашифрован, ключ для дешифрования должен быть в проигрывателе. Иначе и быть не может. Все было в порядке, пока проигрыватели оставались защищенными от взлома физическими устройствами, но с момента создания программного проигрывателя ключи присутствуют в программах. Кто-то просто произвел анализ программы и нашел ключ, таким образом, содержание видеодисков теперь может копироваться и распространяться через Интернет.
- В 1980 году при проведении Пенсильванской лотереи ее ведущий и несколько рабочих сцены, с которыми он был в сговоре, подтасовали пинг¬понговые шарики, использовавшиеся в розыгрыше, и выиграли 1,2 миллиона долларов. Никто не предполагал возможности такого сложного сговора. В настоящее время проведение лотереи контролируется более тщательно. (Подобное происшествие, но уже вследствие случайной ошибки, имело место при проведении Аризонской лотереи. В 1998 году кто-то обратил внимание на то, что в выигрышных номерах нет ни одной девятки. Оказалось, что алгоритм генерации псевдослучайных чисел содержит элементарную программную ошибку. Кажется, пингпонговые шарики надежнее компьютеров.)
- Правила дорожного движения большинства европейских стран предписывают использование на грузовых автомобилях устройства, называемого тахограф, который присоединяется к спидометру и фиксирует скорость, пройденное расстояние и другие сведения. Тахограф записывает показания спидометра на бумажную ленту, на которой водитель ставит свою подпись и дату; лента затем хранится какое-то время. Здесь трудно было что-либо подделать, и попытки обмана совершались чаще с использованием слабостей процедуры, а не технологии. Недавно Европейский союз начал финансирование проекта «Тахосмарт» для создания полностью цифрового устрой¬

ства взамен старого тахографа. Любая подобная система открыта для всех видов нападений, описанных в этой книге (хуже всего, что новое устройство, похоже, будет основано на использовании смарт-карт и будет еще менее надежным).

Приведенные примеры интересны тем, что нападающие использовали не недостатки мер противодействия, а просчеты модели безопасности. Во всех случаях меры противодействия существовали, но они не решали истинную проблему. Хотя они могли преодолеть некоторые смежные проблемы. И в некоторых случаях решения создавали еще большие проблемы, нежели те, с которыми удавалось справиться.

# Глава 20. Политика безопасности и меры противодействия

Если достаточно долго заниматься моделированием угроз, станет ясно, что понятие «система безопасности» имеет различные значения в зависимости от ситуации. Вот несколько примеров:

- Компьютеры, используемые в деловой сфере, должны быть защищены от хакеров, преступников и промышленных конкурентов. Военные компьютеры должны быть надежно защищены от тех же угроз, а также от проникновения вражеских военных сил. Некоторые коммерческие компьютеры, обслуживающие телефонные сети, также должны быть защищены от вторжений военных противников.
- Многие городские транспортные системы используют проездные карточки вместо наличных денег. Подобные им телефонные карты применяются повсюду и в Европе и в Азии. Такие системы обязаны быть застрахованы от всевозможных подделок. Конечно, это не проблема, если подделки обходятся дороже, чем настоящие карты.
- Программы безопасности электронной почты должны обеспечить защиту корреспонденции от любых попыток перлюстрации и внесения в нее изменений. Конечно, во многих случаях программными средствами невозможно обезопасить себя от некоторых манипуляций: это троянский конь в компьютере, атаки TEMPEST, видеокамера, которая может считывать с экрана, и т. д. Телефоны, использующие кодирование, имеют тот же недостаток: они сумеют обеспечить тайну переговоров в дороге, но бессильны против «жучков» в помещении.

Хитрость заключается в том, чтобы разрабатывать систему в расчете на реальные угрозы, а не использовать технологии безопасности все подряд в надежде, что из этого что-нибудь получится. Для чего необходимо выработать политику безопасности (иногда называемую моделью безопасности), основанную на анализе угроз, и уже затем разрабатывать механизмы защиты, которые реализуют эту политику и противодействуют угрозам.

# Политика безопасности

Политика безопасности системы подобна внешней политике правительства: она определяет цели и задачи. Когда правительство обвиняют в непоследовательности во внешней политике, это происходит потому, что в его действиях отсутствует логика и нет общей стратегии. Точно так же без политики безопасности меры противодействия цифровой системы будут неупорядочены. Политика — это способ обеспечить всеобщую взаимосвязь.

Хорошая политика формируется как ответ на угрозу. Если угрозы отсутствуют, то нет и политики: каждый может делать все что угодно. Соединенные Штаты нуждаются во внешней политике ввиду угроз со стороны других государств. Штат Пенсильвания не нуждается во внешней политике, потому что остальные штаты не представляют для него опасности. То же самое с политикой безопасности — она необходима, потому что моделирование угроз не заканчивается пустой страницей. Политика безопасности определяет рамки, в которых осуществляются выбор и реализация мер противодействия.

Большая часть этой книги посвящена тактике, а политика имеет дело со стратегией. Вы не можете решить, какие виды защиты от мошенничества нужно использовать в сотовом телефоне, пока у вас нет стратегии реализации этих контрмер. Вы не можете ожидать, что дюжина инженеров, каждый из которых отвечает за безопасность одной маленькой части системы, будут вести себя согласованно, если нет общей политики, направляющей их работу. О политике безопасности помнят всегда, когда определяют и реализуют меры противодействия.

Не нужно доказывать, что каждая организация нуждается в политике безопасности для своей компьютерной сети. Политика должна очерчивать границы ответственности (кто отвечает за ее реализацию, проведение в жизнь, проверку, пересмотр), определять, что является основой политики безопасности сети и почему именно это. Последнее замечание очень важно, так как произвольная политика, «спущенная сверху» без объяснений, скорее всего, будет проигнорирована. Более правдоподобно, что сотрудники станут следовать ясной, краткой, логичной и последовательной политике.

Политика безопасности — это то, как вы определяете, какие меры противодействия использовать. Нужен ли вам брандмауэр? Как его сконфигурировать? Нужны ли маркеры доступа или достаточно использования паролей? Можно ли пользователям разрешить доступ к видео с их браузеров? Если нет никакой политики, то нет и возможности логически обосновать ответы на эти вопросы.

К сожалению, большинство организаций не имеют сетевой политики безопасности. А если и имеют, то никто ее не придерживается. Я знаю историю проверки одной сети, в которой использовался брандмауэр, защищавший границы между двумя половинами внутренней сети. «Какая сторона находится внутри брандмауэра, а какая — вне его?» — спросил проверяющий. Этого никто не знал. Это — пример организации с плохой политикой безопасности.

В любом случае политика безопасности должна в первую очередь давать ответы на вопрос «почему», а не «как». «Как» — это тактика, контрмеры. Трудно выбрать правильную политику, но еще труднее определить комплекс мер противодействия, которые позволят ее реализовать.

# Доверяемое клиенту программное обеспечение

Мы коснулись различных аспектов этой проблемы в главах, посвященных защите программного обеспечения от копирования, присвоению интеллектуальной собственности и цифровым водяным знакам. Некоторые компании продают программные продукты с правами исключительно индивидуального пользования: аудио- и видеофайлы, которые нельзя скопировать или перепродать; данные, которые можно прочитать, но нельзя распечатать; программное обеспечение, которое не может быть скопировано. Другие компании «продвигают» по электронной почте свои секретные решения в письмах, которые нельзя прочитать по прошествии времени и которые автоматически «удаляются» после определенной даты. Третьи используют технологии электронной коммерции, в которых реализованы другие виды прав.

Общая идея всех этих решений состоит в том, что Алиса может послать Бобу файл, а затем проверить, что впоследствии происходит с полученным файлом. В случае программ, распространяемых по почте, Алиса хочет контролировать удаление файла с компьютера Боба. Если речь идет о продуктах с правами исключительно индивидуального пользования, Алиса посылает Бобу файл, но ограничивает время его просмотра, возможности копирования, изменения и пересылки третьим лицам.

Но это не работает. Контроль над тем, что Боб делает с некоторыми данными, предполагает, что доверенное (Алисой) программное обеспечение установлено на компьютере Боба. Такого не бывает, поэтому эти средства неэффективны.

В качестве примера рассмотрим игры онлайн. Многие игры в Интернете позволяют участвовать в них множеству игроков одновременно, а в некоторых даже проводятся турниры с денежными призами. Хакеры придумали компьютерных противников — ботов (bot), которые помогают в игре, особенно в таких, как Quake и NetTrek. Идея состоит в том, что бот реагирует намного быстрее, чем человек. Таким образом, игрок, их использующий, получает большие преимущества[1]. За этим последовала «гонка вооружений», когда создатели игры пытались выводить из строя этих союзников игроков и заставлять играть по справедливости, а хакеры, в свою очередь, делали более умных и менее уязвимых ботов.

Эти игры создаются в расчете на доверяемое клиенту программное обеспечение, а хакеры умело разрушают любую хитрость, противопоставленную разработчиками игр. Я постоянно восхищаюсь усилиями хакеров, которые они прилагают, чтобы преодолеть системы безопасности. Из этого можно извлечь двойной урок: не только неразумно считать, что программы будут использоваться согласно оказанному доверию, но также нет никакого способа когда-либо достичь нужного уровня защиты.

Противниками всех этих систем — пропадающих почтовых сообщений, ограничения права пользования музыкой и видео, справедливого ведения игры — яв¬

[1] Термин произошел от слова «робот». Первые боты представляли собой виртуальных противников, на которых игроки совершенствовали свою технику. В настоящее время под это понятие подпадают самые разные программы — от «вредных» до «полезных» (например, специальные боты помогают бороться со спамом в телеконференциях, удаляя письма спаммеров). — *Примеч. ред.*

ляются два типа нападающих: средний и квалифицированный пользователи. От среднего пользователя можно защититься любыми средствами. Дядюшка Стив хочет только получить бесплатно копию Norton Utilities, *Короля Льва* или самого последнего компакт-диска Робина Хичкока. Для этой ситуации не существует аналога в физическом мире; дядюшка Стив не сможет сделать отдельную копию сумочки от Шанель, даже если он этого захочет. С одной стороны, он более неуловим; с другой — причиняет меньше материального ущерба. Дядюшка Стив — не организованный преступник, он не собирается создавать преступную сеть. Он даже не станет покупать программное обеспечение, видео- или компакт-диск в том случае, если ему не удастся раздобыть бесплатную пиратскую копию. Остановить дядюшку Стива способны почти все меры противодействия, и нет необходимости в сложных программах для обеспечения безопасности.

Против квалифицированного пользователя бессильны любые контрмеры. В главе 16 я описывал героический путь, который преодолевают некоторые хакеры, чтобы взломать схемы защиты от копирования. Ранее в этой главе я рассказывал о специально разработанных программах-ботах, которые разрушают интерфейс пользователя в компьютерных играх. Поскольку преодоление контрмер имеет такое большое значение для хакеров, бесполезно пытаться построить систему, которая была бы неуязвима. И что еще хуже, большинство систем нуждаются в защите от наиболее ловкого взломщика. Если один человек взламывает Quake (или «Интердоверие», или «Пропажа Инк.»), он может придумать программное средство «выбрать и щелкнуть», которое затем использует любой желающий. Систему безопасности, которая была неприступна почти для всех, сумеет после этого вскрыть каждый.

Единственно возможное решение состоит в том, чтобы поместить механизм дешифрования в защищенное аппаратное устройство в надежде на то, что это замедлит работу профессионалов на несколько лет. Но как только кому-нибудь понадобится «программный проигрыватель», защита будет взломана в течение считанных недель. Индустрия DVD столкнулась с этим еще в 1999 году. Компания Glassbook приобрела такой опыт в 2000 году, когда незащищенные копии *Верхом на ядре* Стивена Кинга появились через два дня после того, как была выпущена электронная версия книги (возможно, имевшая защиту от копирования).

Любая разумная политика безопасности исходит из того, что от профессиональных пиратов невозможно защититься с помощью технологий. Профессиональные пираты в цифровом мире не отличаются от людей, которые подделывают сумочки от Шанель, и общество знает способы поимки таких людей (механизмы обнаружения и реагирования в физическом мире). Они могут быть эффективными или нет, но они совершенно бесполезны в случае борьбы с подделками в цифровом мире. Такая политика безопасности признала бы, что Дядюшка Стив — любитель, и что почти любая мера противодействия ему является достаточной до тех пор, пока она не будет взломана окончательно или не окажется заведомо уязвимой.

Обратите внимание, что это исследование показывает, что разумные поставщики продуктов должны найти альтернативные способы получения прибыли. Продажа копий книги в цифровом мире не приносит такого же дохода, как в мире вещественном. Намного выгоднее продавать обновления в реальном времени, подписку, а кроме того, есть дополнительные причины, чтобы люди покупали бумаж¬

ную копию. Мне нравится покупать компакт-диски, а не копировать их, потому что я получаю вкладыш с пояснениями. Я также покупаю настоящую книгу вместо того, чтобы распечатывать ее электронную копию, потому что книгу в переплете удобно носить с собой. Я готов платить за биржевые сводки, поскольку эта информация ценна тогда, когда поступает своевременно.

Реализацию альтернативных моделей можно видеть в методах общественного финансирования добрых дел: общественное телевидение, общественное искусство и уличные представления. Представления проводятся бесплатно, но индивидуальные пожертвования позволяют им состояться. Вместо того чтобы установить продажную цену в 29,99 доллара на эту книгу, возможно, я должен был поместить на веб-странице предложение о внесении взносов. Я написал бы книгу и поместил бы ее в общедоступном домене, но только после того, как получил бы взносы на 30 000 долларов. (Эта идея использовалась для финансирования некоторых анти¬бушевских кампаний в 2000 году. Люди обязуются сделать взносы со своих кредитных карточек, но деньги будут востребованы только в том случае, если соберется необходимая сумма. Заметьте, компания, выпускающая кредитные карточки, действовала как доверенное третье лицо в этой сделке.)

В других отраслях по-разному решают эти вопросы. Наиболее разумные компании, проводящие игры, выходят из положения, в частности, допуская использование ботов в некоторых турнирах, но проводя заключительные раунды в узком кругу, на проверенном компанией компьютере. Самые недоверчивые фирмы, поддерживающие электронную почту, подчеркивают, что при установке системы безопасности скорее происходит умаление ответственности, нежели достигается абсолютная безопасность программного обеспечения. Угрозу представляют не злонамеренные пользователи, копирующие и распространяющие электронную почту, а честные служащие, забывающие удалить почту, и злостные адвокаты, спустя годы использующие эту корреспонденцию в суде. Но попытка ограничить возможности пользователя на персональном компьютере обречена на неудачу. Честные люди остаются честными, но при этом возникает обманчивое чувство безопасности. Впрочем, иногда и этого достаточно.

## Банковские автоматы

Банкоматы — интересный пример, поскольку модели доверия и безопасности в этом случае переплетены более, чем это кажется на первый взгляд. Банкомат — это сейф, который выдает деньги по команде некоего внешнего устройства. Машина считывает данные пользователя (информацию с магнитной полосы и идентификационный номер), посылает их на центральный сервер и получает ответное сообщение (выдать деньги или отказать, не возвращать карту и т. д.). Банкомат должен быть защищен от мошенников на линиях связи, от взломщиков и от возможности того, что сейф просто увезут. Также необходимы контрольные записи на случай возникновения споров (такой вид учета еще далеко не идеален).

Многие люди должны иметь доступ к банкоматам. Инкассаторы объезжают их на бронированных машинах, чтобы наполнить автоматы наличными деньгами. Обслуживающий персонал должен иметь к ним доступ как в установленное по

графику время, так и незамедлительно в случае чрезвычайной ситуации. И если вспомнить, что состав персонала и охраны может меняться, банк должен иметь возможность прекратить доступ для одних работников и ввести доступ для других.

Также примите в расчет простые финансовые калькуляции. Потери от утраты банкомата определяются только стоимостью его замены и суммой наличных денег внутри. Не имеет смысла тратить 10 миллионов долларов на меры безопасности.

Криптография здесь довольно проста. Канал связи не нуждается в шифровании, требуется только аутентификация. Это можно сделать с помощью или MAC-адресов, или цифровой подписи. Контрольные записи, защищенные хэш-функцией, должны сохраняться и в банкомате, и на сервере.

Обеспечение компьютерной безопасности в этом случае очевидно. Машина должна все проверить. В подозрительных случаях она скорее завершит работу, чем необдуманно выдаст деньги. Программное обеспечение должно быть таким, чтобы его было трудно изменить и обслуживающий персонал не смог бы внедрить в систему троянских коней.

Обеспечение физической безопасности — тоже простая задача. Деньги обязаны храниться в сейфе. Здесь должны находиться и контрольные записи всех, кто открывает сейф (возможно, каждый человек будет иметь свою собственную комбинацию или уникальный маркер доступа). Любые долговременные шифровальные ключи должны быть стерты при первых признаках вмешательства.

Интересно заметить, что владельцы банкоматов только недавно получили в свое распоряжение адекватные физические меры противодействия. До недавних пор банкоматы устанавливали в стене банка или другом безопасном месте. В этой книге я упоминал о грабителях, которые увозили автоматы целиком, что причиняло массу беспокойства. Впоследствии кто-то пришел к заключению, что эти нападения единичны и что можно заработать намного больше денег, если установить банкоматы на каждой автостанции, в каждом баре, торговом центре и на каждой бензоколонке. Это были маленькие автономные банкоматы, не столь защищенные, но последнее не имело значения. Если их устанавливать в общественных местах, то появится больше возможностей для обнаружения и реагирования. В них хранится меньше наличности, поэтому и риск меньше. А доходы высоки, так что они достаточно выгодны. Даже если и исчезнет случайно какой-либо банкомат, идея стоит того.

Совсем недавно появилось другое новшество в политике безопасности. Кто-то наконец осознал, что банкомат состоит из двух частей: хранилища денег и сетевого компьютера, который указывает хранилищу, когда и сколько выдать денег. И нет никакой необходимости соединять эти две части в одном месте. В магазинах уже используется защищенное хранилище денег — кассовый аппарат. Теперь некоторые банкоматы представляют собой только компьютер, и они не содержат наличных денег внутри. Компьютер проводит аутентификацию и затем печатает бланк. Клиент относит бланк к кассовому аппарату и получает деньги. Все это хорошо работает только в случае небольших денежных сумм, но система действует. Это — изящный пример правильного подхода к безопасности... до тех пор, пока кто-нибудь не подделает бланк.

## Компьютеризированные лотерейные терминалы

Компьютеризированные лотерейные терминалы используются в большинстве лотерей типа «Кено». Организаторы лотереи приобретают защищенный компьютер и принтер, который печатает и заверяет лотерейный билет с выбранными номерами. Один или два раза в неделю проводится розыгрыш. Выигрыши бывают как маленькие, так и очень большие.

Угрозы в этом случае очевидны. Опасность исходит непосредственно от самих организаторов лотереи, которые могут вступать в сговор с людьми, обслуживающими лотерейную систему. Они способны мошенничать одним из двух способов: «покупкой» билетов или изменением информации об уже купленных билетах после того, как результаты станут известны. Более изощренный способ — использование ложного терминала, который собирает деньги, но вовсе не выплачивает выигрыши (на самом деле было бы разумнее выплачивать маленькие выигрыши и исчезнуть, когда кто-нибудь сорвет куш).

Эти опасности возникают в случае открытой политики безопасности. Терминалы лотереи должны работать в режиме онлайн и регистрировать все выбранные комбинации чисел на центральном сервере. На этом сервере должны храниться контрольные записи с маркерами времени, и он должен направлять терминалам информацию, которая будет печататься на билете. Сервер в первую очередь должен иметь средства обеспечения безопасности при проведении лотереи. И должен быть предусмотрен способ выявления ложных организаторов лотереи: очевидно, следует позволить получать небольшие выигрыши у любого распространителя билетов, а не только у того, который их продал. Помогают также регулярные проверки.

Есть еще много деталей, которые требуют проработки, но общую идею вы уже представляете.

## Смарт-карты против магнитных карт

В заключение давайте рассмотрим два различных механизма защиты: смарт-карты и запоминающие карты с магнитной полосой. В главе 14 я рассказывал о защите от вмешательства, посягательствах на смарт-карты и о безопасной территории. В главе 19 описывалась модель угроз гипотетической электронной валюте, основанной на смарт-картах. Обладая этими знаниями, давайте зададимся следующим вопросом: будет ли более безопасно пользоваться смарт-картой (картой с микропроцессором), чем картой с памятью (или только с микросхемой памяти, или с магнитной полосой) в конкретном случае?

Тому, кто способен взломать смарт-карту, это безразлично. Он умеет восстанавливать данные с карт обоих типов, но эти карты могут использовать шифрование в качестве защитной меры. Для того, кто не в силах взломать карту, это имеет существенное значение. Он, возможно, умеет считывать данные карты с магнитной полосой, но не может проникнуть в память смарт-карты. С другой стороны, если информация зашифрована каким-либо способом, также не имеет значения,

может ли он считывать магнитную полосу. Возможно, в этом случае имеется меньше различий, чем следовало предположить.

Давайте рассмотрим, как используются карты двух различных типов.

**Карты с магнитной полосой.** Пользователь помещает карту в считывающее устройство и вводит PIN (личный идентификационный номер), пароль или код. Устройство считывает данные с магнитной полосы и использует PIN для расшифровки данных. Затем эти данные обрабатываются устройством для выполнения системой разнообразных действий, для которых она предназначена: вхождение в систему, подписывание электронного чека, плата за стоянку и т. п.

**Смарт-карты.** Пользователь помещает карту в различные считывающие устройства и вводит тот же личный идентификационный номер. Устройство посылает PIN в смарт-карту, которая расшифровывает данные. Затем они используются картой (а не устройством) для выполнения системой нужных действий, а само устройство выполняет в системе функцию ввода-вывода данных.

В чем же различия? В обоих случаях примененное в преступных целях считывающее устройство в состоянии разрушить систему, так как это устройство является единственной связью карты с внешним миром. Как только станут известны секретные данные карты с магнитной полосой, устройство может делать все, что пожелает. Как только смарт-карта получит правильный PIN, считывающее устройство может заставить всех поверить всему, что оно захочет.

Основное различие между этими картами состоит в том, что смарт-карта умеет осуществлять некоторый контроль, так как имеет внутреннюю защиту. Например, если кто-нибудь украдет карту с магнитной полосой, он сможет грубыми приемами завладеть данными этой карты. Он может сделать это автономно, на компьютере, так что ее владелец даже не узнает о случившемся. (Хитрый вор возьмет карту, считает данные с магнитной полосы и затем положит ее обратно в бумажник жертвы.) Смарт-карту нельзя взломать подобным образом, поскольку ее можно запрограммировать так, что она будет выключаться после десяти (или около того) неправильных вводов пароля подряд. Так, если кто-нибудь похитит смарт-карту, узнать пароль с легкостью у него не получится. Он получит возможность сделать только десять попыток. (Предполагается, что вор не умеет взламывать карту. Если он на это способен, то он может сделать это так же, как и в случае завладения картой с магнитной полосой.)

Другое существенное различие состоит в том, что смарт-карта не выдает свои секреты. Например, при использовании карт для подписи документов смарт-карта будет более безопасна, чем карта с магнитной полосой. Карта с магнитной полосой передает считывающему устройству функцию подписания документа, тем самым сообщая ему все секретные данные. В этом случае остается только надеяться на лучшее. Преступник с помощью устройства чтения может украсть шифр подписи. Смарт-карта же самостоятельно ставит подпись. Сканирующее устройство может загружать в карту для подписи подложные документы, но оно не получит шифр подписи.

Есть и другие, более тонкие различия. Смарт-карта позволяет опереться на некоторые основные правила выполнения действий. В принципе это можно использовать и в системе, которая обращается к базам данных, и для карт с магнитной полосой, но смарт-карты позволяют добиться лучшей реализации.

Известно, что смарт-карты распространены как платежное средство по всей Европе, но не в Соединенных Штатах. Почему? Все объясняется особенностями телефонной связи. Система проверки американских кредитных карточек работает в режиме онлайн. Когда вы покупаете что-нибудь, продавец использует модем, чтобы убедиться в том, что ваша карточка действительна и вы платежеспособны. Пятнадцать лет назад эта система не могла бы работать ни в одной европейской стране. Плата за телефон была высока, многие магазины их даже не имели, а в Италии, например, их установки приходилось дожидаться год или два. Связь была дорогой и ненадежной. Создание онлайновой системы в Европе было невыгодно, поэтому индустрия кредитных карт отдала предпочтение смарт-картам, позволявшим хоть както обезопасить сделки. Дело не в том, что смарт-карты защищены лучше, чем карты с магнитными полосами, просто американский способ борьбы с мошенничеством был менее практичным. Настойчивое лоббирование европейскими производителями смарт-карт (Bull SA, Gemplus, и Schiumberger) также нельзя не учитывать.

Если говорить кратко, то существует определенное различие между картами с магнитными полосами и смарт-картами, но насколько это важно — зависит от их применения. Сопротивление вторжению в смарт-карту при достаточных затратах времени и средств всегда может быть преодолено, поэтому не имеет смысла создавать систему, безопасность которой основана на средствах сопротивления вторжению. Большинство людей не способны взломать смарт-карту, так что она достаточно хорошо защищена от большей части преступлений. Но обе карты создавались в предположении, что считывающему устройству следует доверять, поэтому они могут пострадать от действия устройств, используемых злоумышленниками. И все же, смарт-карта лучше защищена от взлома. И до тех пор, пока сопротивление вторжению не преодолено, смарт-карта надежно хранит свои секреты.

## Рациональные контрмеры

Хороши такие меры противодействия, которые не только защищают от известных угроз, но и эффективно действуют в непредвиденных случаях. Хотя и очень трудно обеспечить настоящую безопасность, неразумно полагать, что не случится катастрофа, если в чем-либо допущены просчеты.

Слишком многие системы безопасности чересчур хрупки: они разваливаются от самой незначительной ошибки. Вот несколько примеров:

- Главный секрет систем защиты европейских платных каналов телевидения в течение прошлого десятилетия находился в устройстве настройки телевизора. Это означало, что как только кому-то удастся преодолеть защиту и завладеть ключом, вся система окажется под угрозой.
- Защиту карточки с магнитной полосой MetroCard, использовавшейся для оплаты проезда в метро и автобусах Нью-Йорка, можно было обойти, просто сложив карту в нужном месте (это было в 1998 году).
- DVD-безопасность.

Рассмотрим в связи с этим кредитные карты. Карту трудно подделать, у нее имеются еще такие средства защиты, как голограммы, микропечать и «ультрафио¬

летовые» водяные знаки. Можно завладеть номером кредитной карточки, но как только она объявлена похищенной, ее номер оказывается в «горячем» списке. Зарегистрирована карта как украденная или нет, компьютерные программы все равно просматривают базу данных по движениям денежных средств, ища аномальные случаи их расходования. Даже если злоумышленник умеет обходить все эти защитные меры, карта имеет ограниченный кредит. И, в конце концов, срок действия карты рано или поздно истекает.

Использование некоторых других систем безопасности приводит к непредвиденным последствиям. Дорогие автомобили теперь имеют систему зажигания, которую нельзя замкнуть, это — защитная мера против воров. Сокращая количество угонов, такая мера также изменяет саму модель угрозы: вместо похищения со стоянки увеличивается опасность ограбления автомашин. Это показывает, что эффективность контрпрофилактики не самая высокая, такие меры, как обнаружение и реагирование — намного действеннее.

Вот еще пример: версия Trend Micro's OfficeScan (возможно, уже исправленная) — продукт, осуществляющий проверку на наличие вирусов и уязвимых мест для нападений, приводящих к отказу в обслуживании, сама имеет болевые точки как для нападений этого типа, так и для других.

Управление национальной безопасности действительно хорошо разбирается в этих вещах. Там создают многоуровневые системы защиты и постоянно задаются вопросом: что будет, если они не сработают? Что, если криптография подведет в тот самый момент, когда откажет защита безопасной территории, и система сигнализации останется единственной мерой противодействия? Что, если охрана вместо того, чтобы реагировать на сигнал тревоги, будет занята другим делом? Или что, если машина, которая генерирует ключи шифрования, выйдет из строя, а следом за ней — также и резервная машина? Что, если человек, отвечающий за ремонт резервной машины, подкуплен? Наверное, вы и сами можете продолжить список подобных вопросов.

# Глава 21. Схемы нападений

Даная была дочерью Акрисия, которому предсказали, что он умрет от руки сына своей дочери. Поэтому Акрисий заточил Данаю в бронзовой комнате, вдали от всего мужского рода. Зевс, воспылавший страстью к Данае, проник в ее комнату через крышу в виде золотого дождя. Даная родила Персея, и вы можете догадаться, чем закончилась эта история[1].

Моделирование угроз по большей части производится ad hoc — для конкретного случая. Вы размышляете о возможных угрозах до тех пор, пока больше уже не можете о них думать, и прекращаете это занятие. Потом вы удивляетесь и приходите в ярость, когда кто-нибудь придумывает такой вид нападения, о котором вы и не подозревали. Мой любимый пример по этому поводу — это банда калифорнийских воров, которые владели искусством вламываться в чужие дома, проделывая бензопилой отверстие в стене. Взломщики совершенно не вписывались в модель безопасности, так как защитные меры, используемые хозяевами, представляли собой дверную и оконную сигнализацию. Они оказались бесполезны в этом случае.

Для облегчения задачи я построил схему действий нападающих, которую назвал *деревом атак*. Деревья атак представляют собой методологию описания угроз и мер противодействия для защиты системы. В расширенном виде схема позволяет представить наглядно безопасность системы. Она дает возможность просчитать защиту, сравнить методы защиты различных систем и проделать множество других хитростей.

Основная идея состоит в том, что мы изображаем возможные нападения на систему в виде древовидной схемы, в которой основная цель помещается в корне, а различные пути ее достижения изображаются в виде ветвей и листьев. Приписывая каждому узлу в кроне определенное значение, мы можем выполнить некоторые основные расчеты, позволяющие сделать определенные заключения относительно различных способов нападения. Такая схема называется «деревом И/ИЛИ».

Я начну с простейших деревьев атак для систем защиты, не связанных с компьютерным миром, и постепенно выстрою общую концепцию рассматриваемого предмета.

## Основные деревья атак

На рис. 21.1 показана несложная схема нападения на сейф. Каждая схема нападения имеет свою цель, представленную в корневом узле дерева. В нашем примере

[1] Рок все же настиг Акрисия в виде бронзового диска, брошенного рукой Персея во время состязаний. — *Примеч. ред.*

цель — открытие сейфа. В компьютерных науках деревья «растут» сверху вниз. Чтобы открыть сейф, атакующему нужно открыть замок отмычкой, узнать его шифр, прорезать отверстие в сейфе или сделать что-то при установке сейфа, что позволит потом легко его открыть. Чтобы узнать шифр, взломщик должен либо найти где-нибудь запись комбинации цифр, либо выудить ее у владельца сейфа. И так далее. Каждый узел становится промежуточной целью, а дочерний по отношению к нему узел — это путь ее достижения. Конечно, это только образец дерева, к тому же незавершенный.

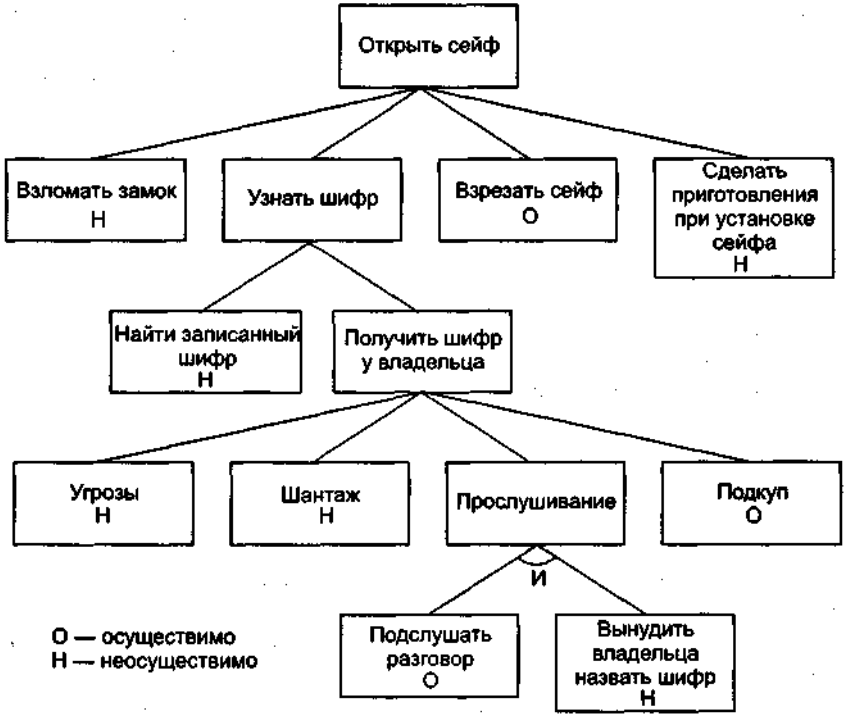



**Рис. 21.1.** Узлы атак

Обратите внимание: на рисунках все, что не обозначено явно как узел И, является узлом ИЛИ. ИЛИ — это узел альтернатив, в нашем примере имеется четыре способа открыть сейф. Узлы И представляют собой отдельные шаги для достижения одной цели. Для того чтобы узнать комбинацию сейфа, взломщик должен подслушать разговор «И» вынудить владельца сейфа назвать комбинацию. Достичь цели можно лишь в том случае, если будут пройдены обе промежуточные цели.

Это — основа дерева атак. Завершив построение, полезно присвоить определенные значения различным листьям дерева (на рис. 21.1 «О» означает «осуществимо», а «Н» — «неосуществимо»). Напоминаю, это только пример для иллюстрации. Не нужно думать, что эти оценки характеризуют реальную защиту моего сейфа в офисе. Сделав оценку узлов (скорее всего, на основании тщательного изучения самого сейфа), можно рассчитать безопасность цели. Узел ИЛИ осу-

ществим, если осуществим любой из его дочерних узлов. Узел И осуществим, если осуществимы все его дочерние узлы, и неосуществим в противном случае (рис. 21.2).

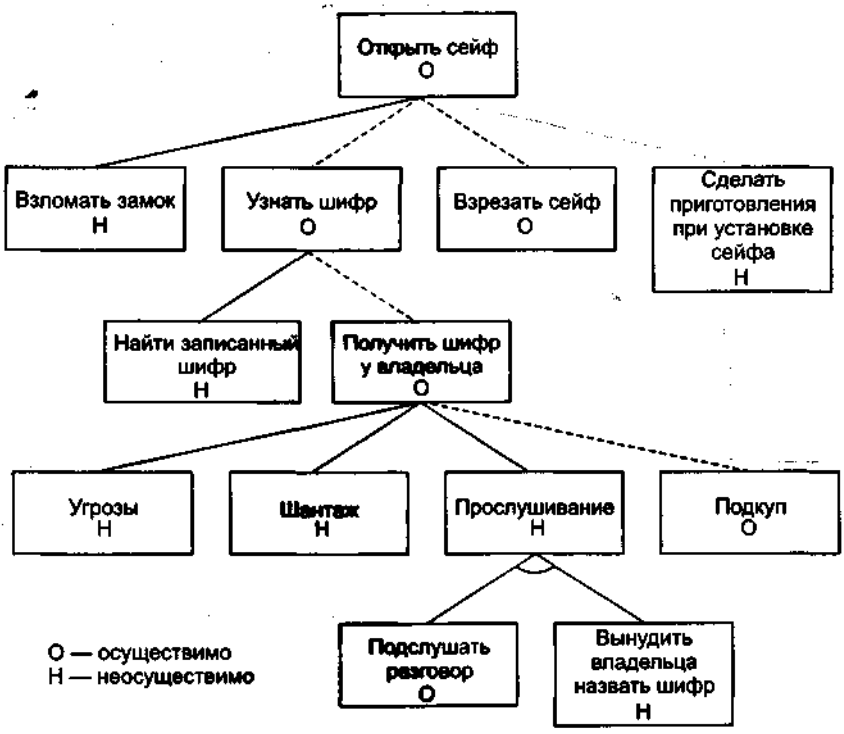



**Рис. 21.2.** Возможные способы взлома

Пунктирные линии на рис. 21.2 показывают все предполагаемые схемы взлома: последовательность осуществимых действий. В этой простейшей схеме существуют два реальных пути проникновения в сейф: прорезать в нем отверстие или узнать комбинацию, подкупив держателя сейфа. Эти сведения позволят правильно организовать защиту.

Присвоение значений (осуществимо или нет) узлам — это далеко не все. Можно затем установить определенные значения всем остальным узлам дерева, используя следующие критерии: легко или сложно, дорого или дешево, законно или незаконно, требуется взлом или нет, требуется специальное оборудование или нет. Рисунок 21.3 демонстрирует ту же самую схему с оценками узлов по критерию «требует специального оборудования» и «не требует специального оборудования».

Присвоить узлам характеристики «дорого» или «недорого» весьма полезно, но лучше показать, сколько именно это будет стоить. Можно задать узлам численные значения. На рис. 21.4 показана схема с указанием расходов на преодоление каждого узла. Так же как и оценка «да/нет», оценка затрат может легко быть распространена по всему дереву. Узлы ИЛИ принимают наименьшее из значений дочерних узлов, а значение узлов И равно сумме значений всех дочерних узлов.

На рис. 21.4 затраты указаны по всей схеме, а также выделен самый дешевый способ взлома.

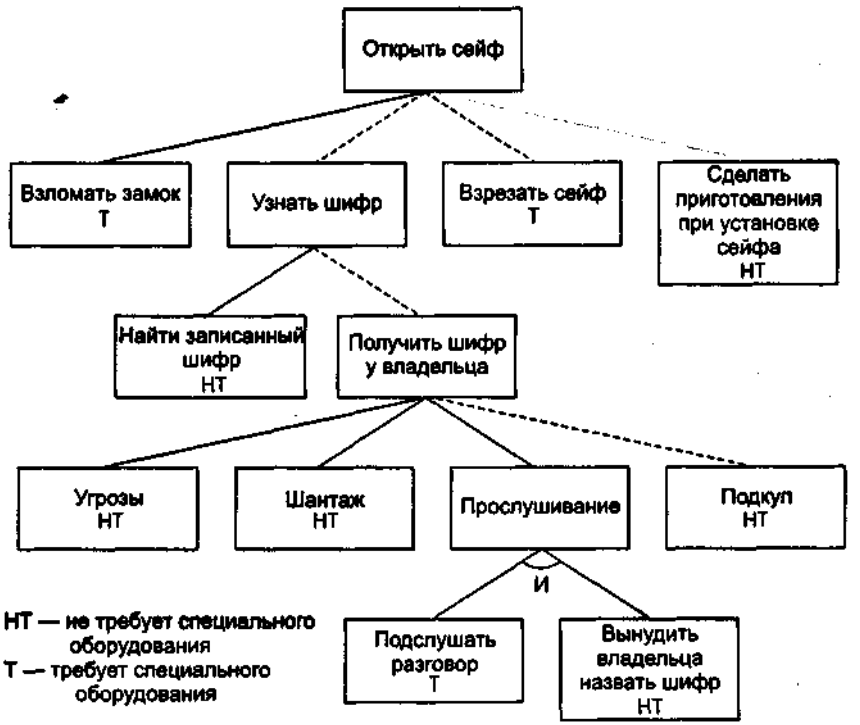



**Рис. 21.3.** Способы взлома: потребуется или нет специальное оборудование

Опять же, эта схема может использоваться для определения уязвимых мест системы. Рисунок 21.5 показывает все возможные способы взлома, которые потребуют расходов меньше 100 000 долларов. Если вы рассматриваете только недорогие виды нападений (может, стоимость содержимого сейфа составляет как раз 100 000 долларов), тогда они вас заинтересуют.

Существует много других всевозможных оценок узлов, включая вероятность успеха различных способов нападения или вероятность того, что нападающий использует определенный метод, и т. д.

В любой практической схеме узлы имеют множество значений, соответствующих различным переменным. Различные значения узлов можно комбинировать, чтобы узнать больше об уязвимости системы. Например, на рис. 21.6 видно, что самые дешевые виды взлома не требуют специального оборудования. Вы также можете видеть способы открытия сейфа, требующие наименьших затрат и сопряженные с небольшим риском, не требующие взлома, не требующие квалификации, самые дешевые с наибольшей вероятностью успеха, «законные» и т. д. Каждый раз, уточняя определенные характеристики нападений, вы больше узнаете о безопасности системы.

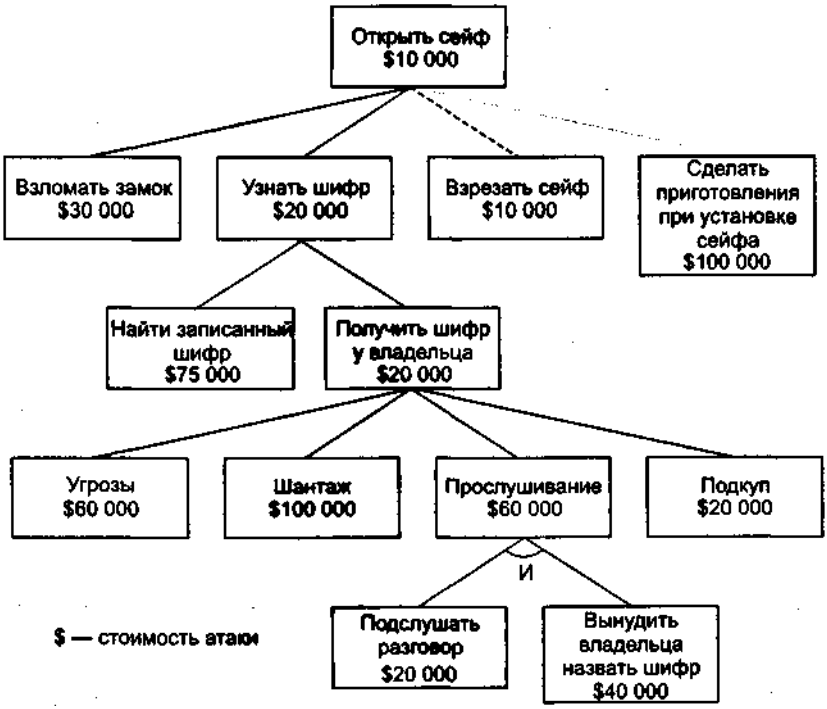



**Рис. 21.4.** Стоимость взлома

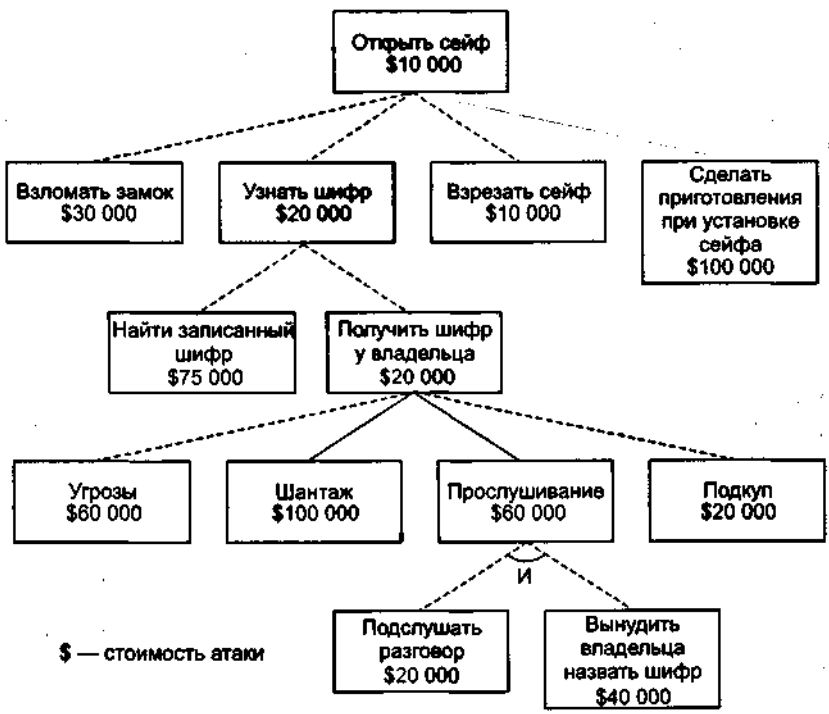



**Рис. 21.5.** Способы взлома, расходы на которые не превышают 100 000 долларов

Чтобы все это хорошо работало, необходимо объединить схему со сведениями о нападающих. Разные нападающие имеют различный уровень мастерства, успеха, неприятия риска, денег и т. д. Если вас беспокоит организованная преступность, то вы должны иметь в виду возможность дорогостоящих видов взлома и взломщиков, готовых попасть в тюрьму. Если вас тревожит угроза нападения террористов, вы не должны забывать о тех, кто готов умереть ради достижения своей цели. Если же речь идет об аспирантах, изучающих вашу систему безопасности, нет смысла беспокоиться о таких незаконных действиях, как взяточничество и шантаж. Характеристика нападающих определяет то, какой части дерева атак следует уделить особое внимание.

Деревья атак также позволяют сыграть в игру «что, если?», рассматривая различные варианты мер противодействия. Например, цель на рис. 21.6 оценивается в 20 000 долларов, поскольку самый дешевый вид взлома, не требующий специального оборудования, — это взятка лицу, знающему комбинацию. Что, если принять меры предосторожности — заплатить этому человеку сумму большую, чем возможная взятка, с тем чтобы он был менее сговорчив? Если предположить, что теперь расходы на подкуп составляют 80 000 долларов (напоминаю, это только пример; в реальности придется исследовать, как именно контрмеры влияют на значение узла), то цена увеличится на 60 000 долларов (возможно, на эту сумму придется нанять головорезов для достижения соглашения).

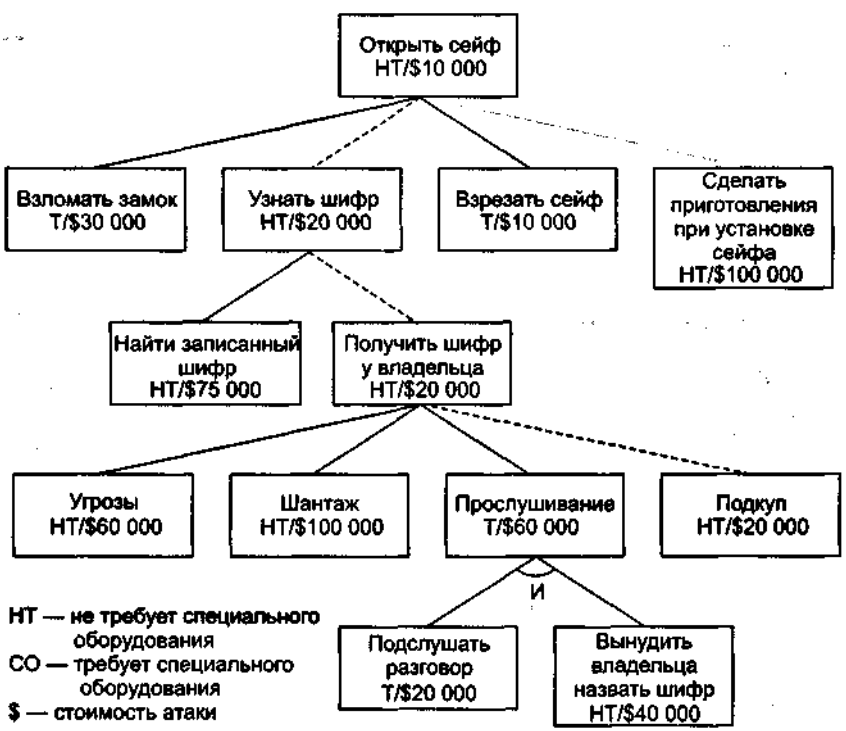



**Рис. 21.6.** Наиболее дешевые способы взлома, не требующие специального оборудования

# Деревья атак PGP

Ниже показана схема нападений на программы безопасности электронной почты PGP. Так как PGP является сложной программой, то и схема получается сложной, и будет проще изобразить ее в виде плана, а не графически. PGP имеет несколько средств безопасности, так что здесь показано только одно из нескольких деревьев атак на PGP. В нашей схеме цель состоит в том, чтобы прочитать сообщение, зашифрованное с помощью PGP. Другими мишенями могут быть: подделка подписи, изменение подписи в письме, незаметное изменение сообщения, подписанного или зашифрованного с помощью PGP.

Если компьютерная программа допускает изменение (с помощью троянского коня) или порчу (с помощью вируса), то ее можно использовать для того, чтобы PGP создавал незащищенную пару (открытый — закрытый) ключей (например, чтобы факторизация осуществлялась на основе множителей, известных нападающему).

# Дерево атак PGP

Цель: прочитать сообщение, зашифрованное PGP (ИЛИ)

1. Прочитать сообщение, зашифрованное PGP
   1.1. Расшифровать сообщение (ИЛИ)
      1.1.1. Взломать асимметричное шифрование (ИЛИ)
         1.1.1.1. Атаковать «в лоб» асимметричное шифрование (ИЛИ)
         Можно подбирать возможные ключи с помощью (известного) открытого ключа получателя до нахождения совпадения. Эффективность этого нападения значительно уменьшается с помощью произвольной избыточной информации, вводимой в процессе симметричного шифрования.
         1.1.1.2. Математически взломать асимметричное шифрование (ИЛИ)
            1.1.1.2.1. Взломать RSA (ИЛИ)
            На сегодня неизвестно, равнозначен ли взлом RSA разложению на множители.
            1.1.1.2.2. Разложение на множители RSA или вычисление дискретного логарифма для схемы Эль-Гамаля.
            Каждое из этих действии требует решения множества теоретических проблем, которые в настоящий момент кажутся очень сложными.
         1.1.1.3. Криптоанализ асимметричного шифрования
            1.1.1.3.1. Общий криптоанализ RSA и схемы Эль-Гамаля (ИЛИ)
            Способы общего криптоанализа RSA или Эль-Гамаль не известны. Криптоанализ одного зашифрованного текста даст общий подход для взламывания RSA и схемы Эль-Гамаля.
            1.1.1.3.2. Использование уязвимых мест RSA и Эль-Гамаля (ИЛИ)
            В RSA есть несколько уязвимых мест; тем не менее PGP устраняет большую часть угроз, с ними связанных.
            1.1.1.3.3. Тайминг-атаки (атаки, основанные на сравнительных измерениях времени) на RSA и Эль-Гамале.
            Тайминг-атаки на RSA уже известны, они вполне могут успешно применяться и против схемы Эль-Гамаля. Однако эти атаки тре¬

буют низкоуровневого контроля за компьютером получателя в то время, как он расшифровывает послание.

1.1.2. Взломать симметричный ключ

1.1.2.1. Взломать симметричный ключ с помощью атаки «в лоб» (ИЛИ)

Все алгоритмы шифрования с помощью симметричного ключа для PGP имеют ключи размером не менее 128 бит. Это делает нереальной лобовую атаку.

Атака «в лоб» в некоторой степени облегчается при включении избыточной информации в начало всех зашифрованных сообщений. См. OpenPGP RFC[1].

1.1.2.2. Криптоанализ шифрования с помощью симметричного ключа

Алгоритмы шифрования с помощью симметричного ключа (поддерживаемые PGP 5.x): IDEA, 3-DES, CAST-5, Blowfish и SAFER-SK 128. Эффективные методы для общего криптоанализа этих алгоритмов не известны.

1.2. Другими способами установить симметричный ключ, используемый для шифрования посланий

1.2.1. Вынудить (обманом) отправителя использовать открытый ключ получателя, чей закрытый ключ известен, для шифрования сообщения, (ИЛИ)

1.2.1.1. Заставить отправителя поверить, что некий подложный ключ (секретный ключ которого известен) — это ключ адресата

1.2.1.2. Убедить отправителя зашифровать послание не одним-единственным ключом: настоящим ключом получателя и другим, секретный ключ которого известен

1.2.1.3. Сделать так, чтобы сообщение было зашифровано некоторым другим открытым ключом, происхождение которого отправителю неизвестно

Этого можно добиться, запустив программу, которая заставит пользователя поверить, что используется правильный ключ, тогда как на самом деле шифрование производится другим ключом.

1.2.2. Заставить получателя подписать зашифрованный симметричный ключ (ИЛИ)

Если адресат слепо подписывает зашифрованный ключ, то он невольно открывает незашифрованный ключ. Ключ достаточно короток, поэтому хэширование не обязательно перед подписыванием. Или если хэш-функция сообщения соответствует зашифрованному ключу, то получателю можно предложить подписать сообщение (или его хэш-функцию).

1.2.3. Контроль памяти компьютера отправителя (ИЛИ)

1.2.4. Контроль памяти компьютера получателя (ИЛИ)

Незашифрованный симметричный ключ должен храниться где-нибудь в памяти во время шифрования и дешифрования. Если память доступна, это дает повод завладеть ключом и прочитать послание.

1.2.5. Определить ключ по генератору случайных чисел (ИЛИ)

1.2.5.1. Определить состояние генератора случайных чисел в момент шифрования послания (ИЛИ)

---

[1] RFC 2440. В настоящее время конкуренцию ему составляет Gnu PG (GPG) — открытая реализация стандарта OpenPGP. Проект был поддержан грантом от правительства Германии, впервые для открытого программного обеспечения. Русскоязычную документацию GPG можно просмотреть на сайте переводчика www.inar.ru/-zwon/gph.html. — *Примеч. ред.*

1.2.5.2. Внедрить программу (вирус), которая определенным образом изменит состояние генератора случайных чисел (ИЛИ)

1.2.5.3. Внедрить программу, которая непосредственно повлияет на выбор симметричных ключей

1.2.6. Внедрить вирус, который откроет симметричный ключ

1.3. Заставить получателя (помочь) расшифровать послание (ИЛИ)

1.3.1. Атаковать симметричный ключ с помощью зашифрованного текста (ИЛИ)

Шифрование в режиме обратной связи, используемое PGP, совершенно не защищено от таких атак. Пересылая адресату тот же ключ (или зашифрованный ключ) вместе с измененным текстом, можно заполучить полное содержание письма[1].

1.3.2. Атаковать открытый ключ с помощью избранного зашифрованного текста (ИЛИ)

Так как RSA и схема Эль-Гамаля достаточно гибки, можно внести определенные изменения в зашифрованный симметричный ключ. Этот измененный (зашифрованный) ключ можно переслать с подлинным сообщением. Это дает возможность атаковать симметричные алгоритмы. Или можно найти слабый зашифрованный текст, и его шифрование с помощью алгоритма симметричного ключа предоставит информацию об измененном ключе, что позволит получить сведения о подлинном ключе.

1.3.3. Отправить любое сообщение адресату (ИЛИ)

Если получатель автоматически расшифровывает сообщение и отвечает на него, то отправитель получит образец шифрования известного открытого текста.

1.3.4. Контроль исходящей почты получателя (ИЛИ)

Если получатель отвечает на сообщение без использования шифрования, то можно собрать информацию о полученном им сообщении

1.3.5. Сфальсифицировать поля «ответить» или «от кого» подлинного сообщения (ИЛИ)

В этом случае получатель может послать ответ по фальшивому адресу электронной почты, и даже если послание засекречено, оно будет зашифровано открытым ключом, секретный ключ которого известен.

1.3.6. Прочитать послание после того, как оно будет расшифровано получателем

1.3.6.1. Скопировать сообщение с жесткого диска или из виртуальной памяти компьютера (ИЛИ)

1.3.6.2. Копировать сообщение с резервной копии, хранящейся на магнитной ленте (ИЛИ)

1.3.6.3. Контроль сетевого трафика (ИЛИ)

---

[1] Исследование этой уязвимости приведено в работе Шнайера «Implementation of Chosen-Ciphertext Attacks against PGP and GnuPG». Способ эксплуатирует стандартную реакцию получателя сообщения, принявшего вместо связного текста околесицу. Скорее всего, получатель вложит в письмо текст исходного сообщения, нажмет вездесущую кнопку «Ответить» и попросит переслать сообщение повторно. Дальше нет проблем. Ключ известен — сообщение расшифровано. Единственное ограничение — текст должен быть не сжатым. — *Примеч. ред.*

1.3.6.4. Использовать средства приема электромагнитного излучения для считывания сообщения, выведенного на экран (ИЛИ)

1.3.6.5. Получение сообщения с устройств вывода

1.3.6.5.1. Получить текст с бумажной распечатки

1.3.6.5.2. Получить текст с фоточувствительного барабана принтера

1.3.6.5.3. Подслушать передачу информации с компьютера на принтер

1.3.6.5.4. Получить информацию из памяти принтера

1.4. Добыть секретный ключ получателя

1.4.1. Разложение на модули RSA или вычисление дискретного логарифма для схемы Эль-Гамаля (ИЛИ)

Оба эти способа требуют решения множества теоретических вопросов, которые в настоящее время представляются очень сложными.

1.4.2. Получить секретный ключ получателя из его связки ключей (ИЛИ)

1.4.2.1. Добыть зашифрованную связку ключей получателя (И)

1.4.2.1.1. Скопировать его с жесткого диска пользователя (ИЛИ)

1.4.2.1.2. Скопировать его резервную копию (ИЛИ)

1.4.2.1.3. Контроль сетевого трафика (ИЛИ)

1.4.2.1.4. Внедрить вирус или закладку для раскрытия копии зашифрованного секретного ключа

Недавно созданный вирус Melissa как раз подходит для такого случая. Существуют и другие возможности: сделать файл открытым для чтения или поместить его в Интернете.

1.4.2.2. Расшифровать секретный ключ

1.4.2.2.1. Взломать зашифрованное с помощью алгоритма IDEA сообщение (ИЛИ)

1.4.2.2.1.1. Взломать IDEA с помощью атаки «в лоб» (ИЛИ)

IDEA использует 128-битовые ключи. Поэтому успешная атака «в лоб» нереальна.

1.4.2.2.1.2. Криптоанализ IDEA

Эффективные методы криптоанализа IDEA не известны

1.4.2.2.2. Узнать пароль

1.4.2.2.2.1. Контроль клавиатуры в момент введения пользователем пароля (ИЛИ)

1.4.2.2.2.2. Убедить пользователя открыть пароль (ИЛИ)

1.4.2.2.2.3. Использовать программу, запоминающую нажатия на клавиши, когда пользователь вводит пароль (ИЛИ)

1.4.2.2.2.4. Угадать пароль

1.4.3. Контроль памяти получателя (ИЛИ)

Когда пользователь расшифровывает полученное письмо, секретный ключ должен помещаться где-нибудь в памяти.

1.4.4. Внедрить вирус для раскрытия секретного ключа

На самом деле более изощренным является способ, указанный в пункте 1.4.2.1.4, где вирус дожидается расшифровки секретного ключа.

1.4.5. Создать для получателя незащищенную пару ключей (открытый — закрытый)

При рассмотрении схемы сразу становится очевидным, что взлом RSA и IDEA алгоритмов шифровки — не самое выгодное нападение на PGP. Существует множество способов считывания посланий, зашифрованных PGP. Вы можете подсмотреть изображение на экране, когда получатель расшифровывает и читает послание (используя троянского коня вроде Back Orifice, приемник TEMPEST или скрытую камеру), завладеть секретным ключом, после того как пользователь введет пароль (с помощью Back Orifice или специального компьютерного вируса), получить пароль (с помощью программы, запоминающей ввод с клавиатуры, приемника TEMPEST или Back Orifice) или попытаться овладеть паролем «в лоб» (это будет в меньшей степени энтропией, чем генерировать 128-битовые ключи IDEA). Выбор алгоритма и длина ключа — наименее существенные вещи из тех, которые могут сокрушить всю систему безопасности PGP.

Ниже приведена более общая схема: цель состоит в прочтении определенного сообщения или во время пересылки, или на одном из двух компьютеров.

## Дерево атак для чтения сообщения электронной почты

Задача: прочитать определенное сообщение электронной почты, посланное с одного компьютера, использующего Window 98, на другой.

1. Убедить отправителя показать письмо (ИЛИ)
   1.1. Подкуп
   1.2. Шантаж
   1.3. Принуждение с помощью угроз
   1.4. Обман
2. Прочитать сообщение, когда оно вводится в компьютер (ИЛИ)
   2.1. Улавливать электромагнитное излучение экрана компьютера (Мера противодействия: использовать TEMPEST)
   2.2. Визуально контролировать экран компьютера
   2.3. Контролировать видеопамять
   2.4. Контролировать шнур для подключения дисплея
3. Прочитать сообщение, хранящееся на диске отправителя (контрмера: использовать SFS для шифрования данных на жестких дисках) (И)
   3.1. Получить доступ к жесткому диску (контрмера: установить замки на все двери и окна)
   3.2. Считать файл, защищенный SFS
4. Прочитать сообщение, когда оно пересылается от отправителя к получателю (контрмера: использовать PGP) (И)
   4.1. Перехватить сообщение во время пересылки (контрмера: использовать программу шифрования транспортного уровня)
   4.2. Прочитать сообщение, зашифрованное PGP
5. Убедить получателя показать сообщение (ИЛИ)
   5.1. Подкуп
   5.2. Шантаж
   5.3. Принуждение с помощью угроз
   5.4. Обман

6. Подсмотреть сообщение во время его прочтения (ИЛИ)
   6.1. Принимать электромагнитное излучение экрана монитора (контрмера: использовать TEMPEST)
   6.2. Следить за изображением на экране
7. Прочитать сообщение, хранящееся на диске получателя (ИЛИ)
   7.1. Получить сообщение с жесткого диска после его расшифровки (контрмера: использовать SFS для шифровки данных на диске) (И)
      7.1.1. Получить доступ к жесткому диску (контрмера: установить замки на двери и окна)
      7.1.2. Считать файл, зашифрованный SFS
   7.2. Получить резервную копию расшифрованного сообщения
8. Получить бумажную распечатку сообщения (контрмера: хранить бумажные копии в сейфе) (И)
   8.1. Получить доступ к сейфу
   8.2. Открыть сейф
9. Украсть компьютер отправителя и попытаться извлечь из него сообщение
10. Украсть компьютер получателя и попытаться извлечь из него сообщение

## Создание и использование деревьев атак

Как создавать дерево атак? Вначале определите возможные цели нападения. Каждая цель формирует отдельное дерево атак, хотя различные деревья могут иметь общие узлы и одно дерево может являться частью другого. Затем обдумайте все возможные виды нападений на каждую цель и включите их в схему. Повторяйте все действия, спускаясь вниз по дереву, пока не закончите. Покажите схему еще кому-нибудь, чтобы он тоже поработал над ней. Проделывайте это столько раз, сколько потребуется; возможно, на анализ уйдет несколько месяцев.

Этот процесс требует творческого подхода, но часто вместо мозгового штурма для решения конкретной задачи используется рутинная методика. Не забывайте высматривать новые формы атак в ландшафте уязвимых точек и делайте это при исследовании каждого шага возможного нападения. Конечно, всегда есть вероятность, что какой-то вид нападения вы упустите из виду, но со временем все у вас будет получаться хорошо. Как и всякий другой вид исследования проблем безопасности, создание дерева атак требует определенного подхода и практических навыков.

Создав однажды дерево атак и проанализировав значения всех его узлов (эти значения будут меняться со временем, поскольку вы будете уточнять сведения о возможных нападениях), вы сможете использовать эту схему для принятия решений по вопросам безопасности. Значения корневого узла позволяют оценить степень уязвимости цели. Вы сможете определить, уязвима ли система для отдельных видов атак, например распределенной атаки, приводящей к отказу в обслуживании. Вы можете использовать дерево атак для того, чтобы очертить круг допущений, исходя из которых решаются вопросы безопасности системы: например, средства безопасности PGP созданы в предположении, что никто не смо¬

жет подкупить программиста[1]. Вы можете оценить последствия изменений в системе или значение вновь обнаруженного слабого места; вычислить новые значения узлов на основе полученной информации и определить, как это влияет на узел, содержащий цель. И наконец, вы можете сравнивать и классифицировать виды нападений по затратам, вероятности успеха и т. д.

Такого рода исследования приводят к удивительному выводу о том, что кажущиеся уязвимыми места на самом деле таковыми не являются. Те, кто используют PGP, обычно беспокоятся о длине ключа: что следует предпочесть — 1024-битовый RSA или 2048-битовый? Схема нападения показывает, что длина ключа RSA не имеет значения. Существует множество других видов нападений, намного более простых, чем взлом открытого ключа: внедрение программы, запоминающей ввод с клавиатуры, изменение программы на жестком диске жертвы. Увеличение длины ключа с 1024 до 2048 бит нисколько не усложняет дерево атак; гораздо более опасны нападения, направленные на преодоление мер компьютерной безопасности. Деревья атак позволяют оценить перспективы системы в целом.

Дерево атак обладает еще одним ценным свойством: содержащиеся в нем сведения остаются актуальны в дальнейшем. Однажды построив дерево атак PGP, вы сумеете использовать его в любой другой ситуации, когда речь идет о PGP. Эта схема может быть включена в другую, более обширную. Например, на рис. 21.2 представлена схема, целью которой является прочтение определенного письма, посланного с одного компьютера, использующего Windows 98, на другой. Если вы посмотрите на терминальные узлы дерева, то заметите, что целые деревья атак на PGP и на сейф вставлены в этот план нападения.

Такая возможность расширения схемы означает, что вам не обязательно знать все на свете. Если вы используете PGP, то вам не нужно знать детали дерева атак на PGP; все, что вы должны знать, — это значения корневого узла. Если вы эксперт в области компьютерной безопасности, вам не обязательно быть в курсе, насколько трудно взломать сейф определенной модели, нужна оценка значений корневого узла. Создав однажды библиотеку деревьев атак на определенные виды компьютерных программ, дверные и оконные замки, на сетевые протоколы безопасности и т. п., вы можете затем использовать их где угодно.

---

[1] Создатель PGP Филип Зиммерман в руководстве пользователя к программе говорит, что «защита от подобных нападений попадает под категорию общих мер защиты от вирусных инфекций». PGP была разработана для однопользовательского персонального компьютера, а главный ключ к работе с PGP — доверие пользователя к самому себе. — *Примеч. ред.*

# Глава 22. Испытание и верификация программных продуктов

Мы уже неоднократно затрагивали тему испытания средств безопасности. В главе 7 обсуждался выбор криптографических примитивов. Там же была выдвинута идея, что наилучшим способом проверки надежности криптографии является открытый криптоанализ, проводимый в течение многих лет. В главе 8 мы рассматривали различные стандарты безопасности компьютера — Оранжевую Книгу и Общие Критерии, а также проверку их соответствия этим стандартам. В главе 13 обсуждались надежность программного обеспечения и то, как ошибки оборачиваются уязвимостью. Испытания позволяют проверить работоспособность системы безопасности: одно дело смоделировать угрозы, разработать политику безопасности и применить меры противодействия, но будут ли эти меры работать на самом деле? Несомненно, вы уже приобрели солидный брандмауэр или антивирусную программу, или VPN (виртуальную частную сеть), или систему защиты от мошенничества для платного телевидения, или систему биометрического контроля, или основанную на использовании смарт-карт систему расчетов, или программу шифрования электронной почты и т. п., но достаточно ли прочна их защита? Большинство программных продуктов ненадежны, и причина кроется в недостаточности тестирования.

Провести правильное испытание средств безопасности не удается по нескольким причинам. Во-первых, изъяны в системе защиты могут возникать когда угодно: при разработке модели безопасности, при создании системы, при реализации, они могут появиться в алгоритмах и протоколах, в исходном коде, при взаимодействии человека с компьютером, в используемой компьютерной системе (в аппаратной части, операционной системе или другом программном обеспечении). Во-вторых, одно-единственное упущение способно лишить продукт защиты. Вспомните, что безопасность — это цепь, надежность которой определяется самым слабым ее звеном. В-третьих, и это наиболее важно, эти недостатки не могут быть обнаружены во время бета-тестирования. Безопасность никак не связана с функционированием. Программы шифрования в состоянии работать нормально, будучи совершенно незащищенными. Недостатки остаются необнаруженными, пока кто-нибудь специально не примется искать их.

На протяжении всей книги я подчеркивал, насколько трудно обеспечить действительную безопасность. Одно дело — спроектировать систему обороны, другое — должным образом реализовать ее, третье — исключить влияние ятрогенных

эффектов...[1] но совсем другое — провести испытания и убедиться в том, что все сделано правильно.

Раньше я был президентом Counterpane Systems, консультационной компании по вопросам шифрования и безопасности. Большую часть времени я тратил на оценку продуктов компьютерной безопасности. Как правило, меня приглашали, когда продукт был почти готов, чтобы проверить, действительно ли он безопасен. Более разумные заказчики обращались ко мне на раннем этапе разработок, чтобы удостовериться в безопасности разработки. Иногда я оценивал готовый продукт, основанный на решениях, которые были проанализированы мною ранее. Эта глава — квинтэссенция того опыта.

## Неудачи испытаний

Перечитайте главу 13, посвященную надежности программного обеспечения, найдите словосочетание «компьютер Сатаны» и вспомните, как продукты безопасности должны работать при появлении противника. Теперь подумайте, как и зачем проводят функциональное испытание.

При функциональном испытании невозможно найти недостатки системы безопасности. В отличие от многих других условий проекта, безопасность не связана с функционированием. Если вы создаете код для текстового процессора и хотите проверить функцию печати, то должны подключить принтер и посмотреть, печатает ли он. Если вы находчивы, то испытаете несколько типов принтеров и напечатаете различные виды документов. Все просто: если программное обеспечение работает как следует, то вы в этом убедитесь.

Безопасность — нечто иное. Представьте, что вы встраиваете функцию шифрования в тот же текстовой процессор. Затем проверяете его таким же образом: шифруете ряд документов, затем расшифровываете их. Дешифрование восстанавливает открытый текст, а зашифрованный текст похож на бессмыслицу. Все это великолепно работает. К сожалению, эти испытания ничего не говорят о безопасности шифрования.

Функциональное тестирование хорошо для обнаружения случайных погрешностей, которые приводят к тому, что компьютерная программа ведет себя непредсказуемо, в основном перестает работать. Недостатки системы защиты не проявляются столь эффектно; обычно они невидимы, пока не станут известны злоумышленникам. Испытание средств безопасности — это не беспорядочное использование программного обеспечения и наблюдение за его работой. Это сознательное выявление проблем, создающих угрозу безопасности. Функциональное испытание никогда не выявило бы, что нападающий может создать веб-страницу, которая будет запускать некоторую программу на компьютере пользователя, просматривающего эту страницу с помощью Microsoft Internet Explorer 3.0 или 3.0.1. Как раз этого и не удастся обнаружить при бета-тестировании.

---

[1] Медицинский термин. «Ятрогенные заболевания (от греч. Iatros — врач и ...ген) — психогении, обусловленные неосторожными высказываниями или поведением медицинских работников, которые создают у человека представление о наличии у него какого-либо заболевания или об особой тяжести имеющейся у него болезни.» — *Примеч. ред.*

Представьте, что производитель поставляет программный продукт, не прошедший вообще никакого функционального испытания: ни внутри компании, ни с помощью бета-тестирования. Производитель лишь заверяет, что программа должным образом компилирована. Вероятность того, что программное обеспечение не имеет ошибок, в этом случае равна нулю. Даже если это простой продукт, все равно он содержит тысячи ошибок и будет все время ломаться самым неожиданным образом. Он не будет работать.

А теперь представьте, что производитель продает программный продукт без какого-либо испытания средств безопасности. Правда, теперь производитель проводит обычные функциональные испытания. Но вероятность того, что этот продукт не содержит ошибок в защите, также равна нулю.

К сожалению, слишком многие программные продукты, даже продукты безопасности, имеют те же проблемы.

Даже достаточно полный анализ безопасности не сильно поможет. Я обнаруживал от 5 до 15 ошибок на тысячу строк кода, и это — в конечном продукте, после всех испытаний. Мы все знаем, какое огромное количество ошибок можно найти в операционной системе Microsoft, даже после сотен человеко-часов испытаний. Точно так же дни, недели и даже месяцы исследования безопасности не приведут ни к чему.

Другая сторона проблемы состоит в том, что полноценное исследование безопасности может быть проведено только опытными специалистами. Вспомним, что о продукте безопасности в лучшем случае можно будет сказать: «Я не могу взломать его, и другие умельцы также не смогут сделать этого». Только опытные специалисты в области безопасности в состоянии действительно обнаружить недостатки системы, поэтому качество любого испытания зависит от профессионализма исследователей.

Иногда недостатки защиты обнаруживаются случайно. Хороший пример — изъян в защите пароля Microsoft Bob: она позволяет повторно вводить пароль и после трех неправильных попыток. Хотя это — исключение. Вероятность случайного попадания на какую-либо ошибку в системе безопасности очень низка, иногда она стремится к нулю. Более эффективен целенаправленный поиск.

К сожалению, еще не создана такая полезная вещь, как всеобъемлющий справочник по вопросам безопасности. Те из нас, кто работают в этой области, зачастую создавали свои собственные справочники, содержащие перечни возможных нападений и уязвимых мест, которые встречаются в коммерческих продуктах, описаны в научной литературе или придуманы нами самими. Подобные перечни огромны — пару лет назад я составил такой список из 759 нападений, но и он не был исчерпывающим.

Нетрудно провести испытания на предмет некоторой заданной уязвимости. Иные слабые места легче найти, чем другие. Поиск каждого узкого места требует много времени, но приближает к цели. Всестороннее испытание на предмет всех известных слабых мест все еще остается трудным делом, так как для этого нужно постоянно обновлять и пополнять их перечень. Это отнимает время, но все же осуществимо. Проблема в другом: испытание на предмет всех возможных слабых мест невозможно.

Обратите внимание, я не говорю «очень трудно» или «невероятно трудно». Я сказал «невозможно».

Поиск всех возможных слабых мест предполагает исследование даже тех из них, **о** которых ни вы, и ни кто-либо еще не могли и подумать. Если вы строите мост, вы должны быть готовы гарантировать, что мост не обрушится в результате действия природных сил. Вероятно, вы сможете составить список воздействий, к которым мост будет устойчив. Вы даже можете предусмотреть защиту моста от возможных террористических актов. Но вы никогда не станете утверждать, что мост устоит перед какой-либо неизвестной технологией, которая еще не создана.

Все сказанное касается не только программного обеспечения широкого потребления. Эти рассуждения в равной мере относятся и к аппаратным средствам безопасности, и к большим частным системам, и к военным аппаратным и программным средствам, и ко всему остальному. В том числе к технологиям безопасности, не имеющим отношения к компьютерам. Это общие проблемы.

Что делать разработчику системы? В идеале — он должен перестать полагаться на своих собственных проектировщиков и бета-тестирование. Ему следует нанять независимых экспертов в области безопасности, которые проведут испытания. На них придется истратить значительные средства; скорее всего, это потребует стольких же усилий, сколько и сама разработка и реализация.

Но никто, за исключением военных, не собирается поступать таким образом. И даже они, видимо, не всегда делают это, а только когда речь идет о системах управления ядерным вооружением.

Производители же намерены поступать, как всегда, то есть продавать ненадежные продукты, и лишь затем устранять изъяны в защите, которые будут обнаружены и преданы гласности. Они будут делать из ряда вон выходящие заявления и надеяться, что никто не призовет их к ответу. Они будут проводить конкурсы по взлому их систем и устраивать другие рекламные трюки. Они станут выпускать новые версии программ так быстро, что к тому времени, когда кто-нибудь потрудится закончить анализ безопасности, они смогут сказать: «Да, но это было в гораздо более ранней версии». Однако продукты все равно останутся ненадежны.

## Выявление недостатков защиты продуктов при использовании

Каждый день обнаруживаются новые изъяны в безопасности представленных на рынке программных продуктов. Их раскрывают сами клиенты, исследователи (ученые и хакеры) и преступники. Насколько часто — это зависит от известности продукта, упорства исследователей, сложности программы и качества проведенного производителем испытания средств безопасности. Если речь идет о популярной операционной системе, то это случается несколько раз в неделю. В случае малоизвестной программы шифрования прореха обнаруживается лишь однажды за все время ее существования.

Так или иначе, кто-нибудь да находит уязвимые места в системе безопасности. И что дальше?

Существует несколько вариантов его дальнейших действий. Он может сохранять все в тайне и никому не сообщать об этом или поделиться только со своими друзьями. Он может уведомить производителя. Или оповестить своих собствен¬

ных клиентов, постаравшись не раскрывать ошибку, чтобы только его продукты могли защищать пользователя (мне встречались компании, поступавшие таким образом). Или предать гласности свою находку. (Конечно, он всегда может использовать в преступных целях свои знания об уязвимых местах, но давайте предполагать, что он — честный человек.) Практика предания гласности, известная как *полное раскрытие* (full disclosure), стала популярна в последние годы. И остается предметом горячих споров.

Но сначала немного истории.

В 1988 году, после того как использование червя Морриса продемонстрировало, насколько легко провести нападение через Интернет, Агентство перспективных исследовательских программ (Defense Advanced Research Projects Agency, DARPA) начало финансирование группы, которая, как предполагалось, будет координировать ответные меры безопасности, повышать уровень осведомленности в вопросах безопасности и вообще делать много полезных вещей. Она называется Группой компьютерной «скорой помощи» (Computer Emergency Response Team, CERT), ее центральное подразделение находится в Университете Карнеги-Мел¬лона в Питсбурге.

Все эти годы CERT действует как центр обмена информацией по вопросам уязвимости систем безопасности. Как и предполагалось, люди сообщают в CERT обо всех обнаруженных ими уязвимых местах. CERT проверяет, действительно ли они существуют, уведомляет об этом производителя и после того, как последний исправит ошибки, публикует подробные сведения о них (и о сделанных исправлениях).

Это хорошо выглядит в теории, но плохо работает на практике. Были три главные претензии к этой системе. Во-первых, CERT медленно проверяла наличие уязвимых мест, поскольку получала множество сообщений и не успевала справляться с работой. Во-вторых, производители не торопились исправлять ошибки, о которых им сообщала CERT, поскольку она не публиковала никакие сведения, прежде чем ошибки были исправлены, и не было необходимости в спешке. И в-третьих, CERT не сразу публиковала сообщения даже после того, как все исправления были сделаны.

Практика полного раскрытия возникла из-за общего разочарования в описанной процедуре. Конференции в Интернете, такие как Bugtraq и NT Bugtraq (организованные в 1993 и в 1997 годах соответственно), превратились в форум для людей, считавших, что производителей извещать бесполезно, а единственный способ повысить безопасность — предавать гласности случаи, когда ее средства оказываются ненадежны. Это была реакция протеста на «башню из слоновой кости», воздвигнутую учеными, хранившими в тайне свои познания. Как писал один хакер: «Теперь обсуждение проблем безопасности не будет ограничено закрытыми списками рассылки так называемых специалистов по вопросам безопасности, и подробности можно будет найти не только в пространных, перегруженных деталями академических статьях. Напротив, информация станет общедоступной, и каждый сможет использовать ее по своему усмотрению».

Сегодня многие исследователи публикуют в конференциях сообщения об обнаруженных ими уязвимых местах, иногда делая также сообщения в печати. Средства массовой информации и компьютерная пресса перепевают в рассылках эти сообщения, обрастающие слухами и домыслами. (Вот почему за прошедшие годы

в прессе было так много подобных историй.) Производители стараются «залатать» прорехи в защите сразу, как только они становятся достоянием гласности, поэтому они также могут публиковать сообщения о том, как быстро и тщательно они исправляют ошибки. Системы безопасности совершенствуются намного быстрее благодаря практике полного раскрытия.

В то же время хакеры используют эти рассылки для сбора информации об уязвимых местах и для написания вредоносных программ. Некоторые виды нападений довольно сложны, но те, кто способен разобраться в них, могут составить программы с интерфейсом вида «выбрать и щелкнуть», которые сумеют использовать и все остальные. Это — обратная сторона полного раскрытия, которая может быть истолкована таким образом, что публикация подробностей об уязвимых местах приносит больше вреда, чем пользы, вооружая хакеров средствами для взлома системы. Те, кто придерживается такой точки зрения, считают, что средства безопасности лучше работают, если их уязвимые места не обнажаются перед публикой.

Сторонники полного раскрытия возражают на это, заявляя, что такие представления основаны на далеко не всегда верном предположении, будто сведения об уязвимых местах предает гласности обязательно тот, кто первым их обнаружил. Иногда уязвимые места становятся известны нападающим за месяц или даже год до того, как их обнаружит производитель (эти сведения тайно распространяются в хакерском подполье). Как говорится, пример поучительный. Чем скорее уязвимые места станут общеизвестны и будут исправлены, тем лучше будет всем.

Действительность показывает, что «латание» слабых мест не является решением проблемы; многие системные администраторы не используют «заплаты», сделанные производителями. Многие компании лукавят, заявляя: «Мы выпустили "заплату". Что еще мы можем сделать?» В физическом мире товары с браком часто возвращают продавцу. Но это никогда не случается в компьютерном мире. Даже после того как производитель устраняет ошибки и стихают волнения в прессе, системы так и остаются уязвимыми.

Следующий пример поясняет сказанное. В апреле 1999 года кто-то обнаружил брешь в Microsoft Data Access Components, которая позволяла контролировать удаленную систему Windows NT. Об этом сразу же было сообщено в открытой конференции. Хотя ее модератор утаил от публики подробности этой опасности больше чем на неделю, все же какой-то хакер провел анализ и выяснил детали, которые позволяли использовать уязвимость.

Примерно в то же самое время Microsoft выпустила «заплату», которая должна была препятствовать нападающим в использовании уязвимости пользовательских систем. Microsoft также издала бюллетень по этой теме безопасности и опубликовала несколько других сообщений по вопросам безопасности.

Но «заплата» Microsoft не смогла чудесным образом устранить причину опасности. В тот же год, во время празднования Дня Всех Святых хакеры, воспользовавшись уязвимостью системы, стерли более чем 25 веб-сайтов, основанных на NT, принадлежавших администраторам безопасности, которые не беспокоились (или даже не знали, что следовало побеспокоиться) об обновлении конфигурации в течение шести месяцев.

Вот и все в двух словах.

Microsoft никогда не исправила бы это уязвимое место, если бы не существовал сценарий его использования (exploit script). В других случаях Microsoft заходила так далеко, что или полностью игнорировала проблему, объявляя опасность «чисто теоретической» и поэтому не заслуживающей внимания, или утверждала, что исследователь лжет. Вопросы болевых точек безопасности беспокоят Microsoft лишь в связи с формированием общественного мнения. Когда возникает проблема, компания что-нибудь предпринимает, но редко заранее. Таким образом, огласка сведений об уязвимых местах приводит к их устранению.

Эти сведения также позволяют написать имитатор атаки[1], давая возможность группе преступных хакеров извлечь выгоду из уязвимых мест как (1) в промежутке времени между их обнаружением и опубликованием «заплаты», так и (2) после этого, поскольку многие системные администраторы не используют «заплаты» Microsoft.

Что лучше: публиковать сведения или сохранять их в тайне?

Иногда это зависит от производителя. Большинство компаний непредвзято реагируют на сообщения о нападениях на их системы. Они признают и устраняют проблему, помещают ее решение на свой веб-сайт, и все приходит в норму. Другие производители реагируют иначе; некоторые компании, предоставляющие услуги цифровой сотовой связи, отвечают ложью и нападками на публикации об изъянах алгоритмов шифрования, а не исправляют положение. Индустрия развлечений преследует в судебном порядке людей, показавших, насколько паршиво обеспечена безопасность DVD-плейеров, и людей, впоследствии говоривших об этом. Вообще говоря, выявленные уязвимые места, которые нелегко устранить (намного труднее внести изменения в 10 миллионов проданных сотовых телефонов, нежели разослать через Интернет решение проблемы программного обеспечения), гораздо более осложняют жизнь компании.

Иногда у исследователя нет выбора. Один служащий Управления национальной безопасности неофициально утверждал, что его коллеги зафиксировали несколько новых нападений в Интернете, но им было запрещено публиковать информацию. Позже некоторые из них были обнаружены другими исследователями, другие остались тайной. Бывает, что у исследователя есть выбор, но он предпочитает молчание. В течение нескольких лет Стив Белловин скрывал написанную им статью о нападениях на систему службы доменных имен (DNS). Белловин и Чес¬вик преднамеренно не рассказали в своей книге, посвященной брандмауэрам, о синхронных лавинных адресациях в сети.

Netscape обычно предлагала 1000 долларов (и тенниску в придачу) в награду нашедшему дефект в безопасности своего программного обеспечения. Было выписано всего несколько чеков, однако в 1997 году датский хакер нашел прореху в системе безопасности и потребовал большие деньги. Дело обернулось так, что он не получил своих денег: его описание эффектов, связанных с программными ошибками, позволило инженерам Netscape воспроизвести и устранить их и без его по-

---

[1] Exploit script — наиболее близкий перевод — «сценарий использования ошибок или дефектов в программах в своих интересах». Эквивалента в русском языке пока не имеется. Наиболее подходящее определение — «имитатор атак» — введено в употребление экспертом по сетевой безопасности А. Лу¬кацким (exploit применяется в процессе «зондирования» ). — *Примеч. ред.*

мощи. В 2000 году французский исследователь обнаружил, как взломать систему безопасности смарт-карт CB (Groupement des Cartes Bancaries). Затем, по сведениям из разных источников, он то ли предложил свои услуги, то ли занялся шантажом. В конечном счете он был арестован и осужден условно.

Безопасность рождается в соперничестве, даже в академических «башнях из слоновой кости». Кто-то предлагает новую схему: алгоритм, протокол, техника; другой взламывает ее; кто-то третий все восстанавливает и т. д. Все превращается в забаву. Но когда дело касается уже выпущенных и используемых систем, это может обернуться мошенничеством. Действительно ли выгода от огласки нападения перевешивает возрастание угрозы со стороны противника, получившего сведения о таких возможностях? (На языке Агентства национальной безопасности это называется «выпуском акций».) Почему компания должна наживаться на работе исследователя? Будет ли компания игнорировать проблему до тех пор, пока исследователь не обнародует данные? Заботит ли самого исследователя реакция публики? В любом случае, как вести себя исследователю?

Этот последний вопрос никогда не имел должного обсуждения. Публикация слабых мест безопасности — это своего рода «атака ради огласки»: исследователь хочет увидеть свое имя в газете. Иногда такие сведения разглашает или консультант по вопросам безопасности, или служащий компании, которая занимается оценкой уязвимости или предлагает средства защиты сети. Это особенно верно, если новость появилась в прессе; сообщение в PR Newswire или Business Wire дорого обходится, и никто не будет этого делать, не будучи уверенным, что затраты окупятся.

Вообще я одобряю практику полного раскрытия и думаю, что она скорее укрепляет безопасность, нежели наоборот. Эта книга, которую могут прочитать как хорошие, так и плохие парни, не представляет угрозы безопасности потому лишь, что в ней описаны методы нападений. Точно так же разглашение сведений о слабых местах не то же самое, что их появление. Производители не беспокоятся об устранении обнаруженных, но неопубликованных ошибок (этим грешит не только Microsoft, мы наблюдаем такое почти в каждой крупной компании), поэтому публикация — первый шаг к ликвидации ошибки. Наказывать того, кто разгласил сведения об ошибках, — все равно, что казнить гонца, принесшего дурные вести. Виноват во всем сам производитель, выпустивший ненадежное программное обеспечение.

Но бывают и исключения из правил.

Во-первых, я против такой огласки, которая, прежде всего, сеет панику. Сообщения о слабых местах, о которых нет достаточных свидетельств, очень вредны. (Пример тому — случай, когда кто-то обнаружил переменную, содержащую три буквы NSA, в шифровании API Microsoft[1] и объявил, что Агентство национальной безопасности (National Security Agency) установило лазейку в изделия Microsoft.) Так же плохи сообщения об уязвимых местах в ответственных системах, которые не могут быть легко устранены и знания о которых способны причинить серьезный вред (например, программное обеспечение управления воздушным движени¬

[1] Программный интерфейс шифрования Crypto API, обеспечивающий шифрование и передачу электронной подписи для приложений независимых производителей. — *Примеч. ред.*

ем). Я полагаю, что это остается на совести исследователей — определять баланс выгоды от раскрытия уязвимости и связанных с этим опасностей.

Во-вторых, я верю в эффективность предварительного уведомления производителей. CERT впадает в крайность, давая иногда компаниям годы для разрешения проблемы. В результате многие производители не принимают уведомление всерьез. Но если предупреждение о том, что сведения об уязвимых местах будут опубликованы через месяц, исходит от исследователя, то может оказаться, что такое сообщение появится одновременно с объявлением о сделанных исправлениях. Это выгодно каждому.

И в-третьих, я полагаю, что эта практика заходит слишком далеко. Написание научных статей по вопросам уязвимости помогает исследованиям и помогает в разработке систем безопасности. Написание демонстрационного кода — часто необходимая часть исследования. С другой стороны, массовое распространение средств нападения — плохая идея. Создание хакерских инструментов с интерфейсом «выбрать и щелкнуть», которые может использовать каждый начинающий хакер, не приводит ни к чему хорошему. Эта тенденция помогает преступникам и делает вычислительные сети менее безопасными. Она не решает проблему, а создает новые.

Иногда трудно определить, что хорошо и что плохо. Средства оценки уязвимости могут использоваться как для укрепления безопасности, так и для взлома системы. Средства удаленного доступа во многом похожи на Back Orifice. Если такая компания, как Microsoft, лживо отрицает в прессе реальность обнаруженных уязвимых мест, будет ли правильным опубликовать реальный сценарий нападения? Я считаю, что нужно содействовать решению проблем, а не создавать новые. Полное раскрытие — это часть решения. Устранение проблемы и усиление безопасности сети — тоже часть решения. Я не против тех средств, которые можно применять как в хороших, так и в дурных целях, но я не приемлю те средства, которые предназначены только для скверных дел.

В холле здания ЦРУ высечена в камне цитата из Библии: «И познаете истину, и истина сделает вас свободными» (Ин 8: 32). Знающие правду способны использовать эти знания, чтобы добиться победы над теми, кто не знает ее (или кто отказывается поверить в нее). Полное раскрытие приводит нас ближе к истине, чем что-либо другое.

## Открытые стандарты и открытые решения

В главе 7 я рассказывал о преимуществах использования открытого, общедоступного шифрования вместо частного, закрытого. Поскольку единственным свидетельством в пользу безопасности криптографических примитивов является длительное исследование многими специалистами, наиболее выгодно сделать их открытыми. Именно этот довод побуждает любого разумного разработчика системы безопасности использовать открытые решения для всего, что связанно с безопасностью, включая открытый исходный код программного обеспечения.

Обратите внимание: безопасность не имеет ничего общего с функциональностью. Поэтому никакое бета-тестирование не поможет выявить недостатки безопас¬

ности. Единственный способ обрести уверенность в устойчивости системы к нападениям — это длительное испытание ее специалистами. И только одним способом можно достичь этого — сделать подробности системы общеизвестными.

Детали хорошо спроектированной защиты не являются секретом. В тайне сохраняются лишь некоторые изменяемые параметры: ключи шифрования, пароли, маркеры доступа и т. д. Противоположность этому — *«безопасность через засекречивание»* (security by obscurity), где сохранение в тайне деталей системы становится условием безопасности. Если система разработана таким образом, то ее защита довольно хрупкая. Как смогли убедиться разработчики системы безопасности цифровой сотовой связи или схемы шифрования DVD, или интерфейса FireWire, рано или поздно потаенное становится явным. Плохо разработанная система защищена, пока детали остаются в секрете, но быстро ломается, как только о них кто-нибудь узнает. Хорошо разработанная система безопасна, даже если ее детали общеизвестны.

Итак, поскольку хорошая разработка системы безопасности не связана с засекречиванием и можно много выгадать, если опубликовать подробности системы безопасности, имеет смысл поступать именно так. Открытые системы, скорее всего, будут тщательно исследоваться, а значит, будут и более надежны, чем закрытые системы.

Эти рассуждения применимы непосредственно к программному обеспечению. Единственный способ найти недостатки безопасности в коде состоит в том, чтобы исследовать его. Это верно для всех кодов — и открытых, и закрытых. И этим не может заниматься кто попало, требуются специалисты в области безопасности программного обеспечения. На протяжении нескольких лет их помощь неоднократно потребуется для оценки безопасности системы с различных точек зрения. Можно нанять таких экспертов, но будет намного дешевле и эффективнее позволить всему обществу заниматься этим. И лучший способ поспособствовать этому — опубликовать исходный код.

Предвижу возражение, что публикация кода подарит нападающим информацию, необходимую для обнаружения и использования уязвимых мест системы. Сохранение кода в тайне, как считается, не позволяет нападающим получить нужную информацию.

Это наивное утверждение. Обнародование исходного кода увеличивает не количество слабых мест, а осведомленность широкой публики. Производители, держащие исходный код в тайне, скорее всего, небрежны. А производители, делающие свой код открытым, имеют больше шансов обнаружить уязвимые места и устранить их. Засекреченное программное обеспечение ненадежно. Публикация исходного кода обеспечивает большую безопасность, чем сохранение его в тайне.

Однако открытое программное обеспечение не гарантирует безопасность. Нужно помнить о двух вещах.

Во-первых, простая публикация кода не означает автоматически, что его станут исследовать на предмет безопасности, и уж конечно не означает, что этим займутся специалисты. Например, исследователи нашли ошибки переполнения буфера в коде, созданном в Массачусетском технологическом институте для Kerberos, через десять лет после выпуска этого кода. Другой открытый модуль — программа Mailman, предназначенная для работы со списками адресатов конференций, — бо¬

лее трех лет имела бросающиеся в глаза недостатки защиты, пока сам разработчик не пересмотрел код и не обнаружил их.

Исследователи безопасности — непостоянные и вечно занятые люди. Они не имеют ни времени, ни склонности исследовать каждую часть опубликованного исходного кода. И хотя полезно сделать исходный код открытым, это не гарантирует безопасность. Я мог бы назвать дюжину библиотек открытого кода программ защиты, о которых никто никогда не слышал. С другой стороны, открытый исходный код защиты различных программных средств UNIX изучался многими профессионалами в области безопасности.

Кроме того, обнародование кода не гарантирует, что проблемы безопасности решаются сразу, как только обнаруживаются. Нет оснований надеяться, что некая часть открытого исходного кода двухлетней давности имеет меньше недостатков, чем часть закрытого кода такого же стажа. Если открытый исходный код интенсивно исследовался, это может быть похоже на правду. Но только то обстоятельство, что часть исходного кода была доступной в течение нескольких лет, само по себе ничего не означает[1].

Я — за открытый исходный код и полагаю, что таким образом можно повысить уровень безопасности. Но программное обеспечение не становится автоматически надежным только потому, что делается открытым, и, наоборот, оно не делается небезопасным, если остается закрытым. Другие отмечали, что открытый код кажется более безопасным, и эта необоснованная вера заставляет людей доверять ему больше, чем следует. Это плохо.

Также обратите внимание, что в этом исследовании полностью игнорируется существенный вопрос о том, как сделать программное обеспечение безопасным в первую очередь на стадии разработки. Использование открытого кода — это, во-первых, стратегия бизнеса и, во-вторых, стратегия безопасности. К сожалению, похоже, что традиционные методы закрытого программного обеспечения более эффективны при создании высококачественного крупного продукта. Возможно, лучше всего в отношении безопасности создать закрытое программное обеспечение и затем сделать его открытым (как поступила Netscape со своим кодом браузера).

## Перепроектирование и закон

Стараясь отвертеться от практики полного раскрытия и использования открытого исходного кода, некоторые компании пытались защитить себя с помощью законодательных мер и прилагали усилия к тому, чтобы обратное проектирование (reverse engineering) было объявлено незаконным. Закон Соединенных Штатов об авторском праве в компьютерной сфере DMCA (Digital Millennium Copyright Act) объявляет обратное проектирование уголовным преступлением, такое же положение

[1] Есть мнение, что промышленным стандартом мог бы стать стек протоколов IPX/SPX (для сетевой ОС Novell NetWare), если бы Novell не упорствовала в сокрытии «внутренностей» протоколов, а не TCP/IP. Вследствие этой политики Microsoft имела сложности при согласовании IPX с NWLink. Многие авторитеты компьютерного мира склоняются в сторону *полной* открытости исходного кода. — *Примеч. ред.*

содержится в UCITA (Uniform Computer Information Transactions Act). В настоящее время это становится законом в нескольких штатах.

Мы уже знаем, к чему это приводит. Ассоциация контроля за копированием DVD (DVD Copy Control Association) начала судебное преследование тех, кто перепроектировал их схему защиты DVD, и тех, кто создал общедоступные средства, позволявшие использовать слабые места. Эти люди были арестованы. Mattel выиграла дела против хакеров, которые воспроизвели средства безопасности в CyberPatrol — программе, блокирующей доступ к определенным ресурсам.

Налицо опасный прецедент. Законодательными мерами не укрепить безопасность систем и не воспрепятствовать нападающим в поисках слабых мест. Все, на что они годятся, — это дать производителям возможность не беспокоиться по поводу паршивой безопасности своих продуктов и сваливать с больной головы на здоровую. Конечно, легче реализовать плохие средства безопасности и запретить кому бы то ни было обращать на это внимание, чем трудиться над созданием надежной защиты. Несмотря на то что подобные законы помогают сдерживать распространение взломанного программного обеспечения (пример тому — случаи с DVD и Mattel), в конечном счете они надолго останавливают развитие средств безопасности.

## Состязания по взломам и хакерству

Мы часто сталкиваемся с объявлениями: «Компания X предлагает 10 000 долларов каждому, кто прорвется через ее брандмауэр (взломает ее алгоритм, успешно использует ее протокол в мошеннической операции или сделает что-либо подобное)». Эти состязания хакеров проводятся для демонстрации того, насколько сильна и надежна защита объектов, подвергающихся нападениям. Логика здесь примерно такова: «Мы предложили приз за взлом цели, но никто не сделал этого. Значит, цель хорошо защищена».

Но это не так.

Состязания — негодный способ демонстрации безопасности. Очевидно, продукт (или система, протокол, алгоритм), выдержавший испытание, заслуживает не больше доверия, нежели тот, который ему не подвергался. Результаты состязаний вообще не содержат никакой полезной информации. Тому есть четыре основных причины.

Во-первых, состязания в основном нечестны. Криптоанализ в этом случае осуществляется в условиях, когда нападающий знаком со всем, кроме главного секрета. Он имеет доступ к алгоритмам, протоколам, исходному коду. Он знает зашифрованный и открытый тексты. Вероятно, он даже владеет частичной информацией о ключе. И результат криптоанализа может быть каким угодно. Это может быть полный взлом: когда средства безопасности удается преодолеть в разумные сроки. Это может быть доказательство того, что теоретически взлом возможен: результат, который не имеет практического применения, но все-таки показывает, что средства безопасности не так хороши, как было объявлено. Большинство состязаний по хакерству имеют произвольные правила, определяющие, с чем должен работать нападающий и что следует считать удачным взломом. По некоторым правилам алгоритмы не раскрываются.

Состязания по взлому компьютеров ничем не лучше. Они не раскрывают того, как используются программные продукты, поэтому ничего нельзя сказать относительно того, что является причиной взлома: изъяны самого продукта или ошибки, допущенные при его установке или конфигурировании. Они выявляют особенности различных частей системы: если в состязании проверяется надежность брандмауэра, то что можно сказать об уязвимости операционной системы, из-за недостатков которой этот брандмауэр может оказаться неэффективным?

Правила, по которым определяют победителя состязаний, также произвольны. В 1999 году Microsoft установила веб-сервер Windows 2000 и рискнула предложить хакерам взломать его. Внезапно сервер исчез из Интернета. Вскоре он появился вновь, и Microsoft успокоила, что причиной небытия было отключение питания. (Как это ни странно, похоже, что действительно просто забыли установить систему бесперебойного питания и это сказалось на испытании.)

Нечестные соревнования не новы. В середине 1980-х проводили состязание авторы алгоритма шифрования, названного FEAL. Они предложили файл с зашифрованным текстом и обещали награду первому, кто его прочитает. С тех пор алгоритм неоднократно взламывался. Все признают, что FEAL совершенно ненадежен, однако никто так и не был признан победителем.

Во-вторых, результаты состязания невозможно проанализировать. Они представляют собой случайные, бессистемные испытания. Вправе ли мы считать, что работа десяти человек, каждый из которых затратил по 100 часов, соответствует 1000 часов анализа? Может быть, все они пытались провести одни и те же нападения? Обладают ли они достаточной компетенцией для такого исследования, или они только случайные люди, которые услышали о состязании и захотели испытать удачу? Одно только то обстоятельство, что никто не побеждает в конкурсе, не означает, что цель надежно защищена. Из него следует всего лишь, что никто не признан победителем...

В 1999 году журнал PC Magazine объявил одновременно состязания по взлому Windows NT и Linux box. Первой была взломана вторая. Свидетельствует ли это о том, что Linux менее надежен? Конечно, нет; это означает только, что участники игры сначала проникли в Linux box.

В-третьих, объявленная награда редко бывает достаточным стимулом для участия профессионалов в состязаниях. Анализ безопасности требует большого труда. Люди, хорошо разбирающиеся в этих вопросах, берутся за такую работу по разным мотивам (деньги, престиж, борьба со скукой), но стремление выиграть тенниску едва ли побудит их к этому. Профессионалы в области безопасности зарабатывают гораздо больше, занимаясь своей обычной работой на заказчика или публикуя статьи с результатами своих исследований.

Взглянем на ситуацию с позиций материальной заинтересованности. В среднем час работы компетентного аналитика криптографии или ведущего специалиста по компьютерной безопасности стоит 200 долларов. Всего за неделю работы они получают 10 000 долларов. Этого времени недостаточно, чтобы разобраться с кодом. Вознаграждение в 100 000 долларов выглядит соблазнительно, но обратное проектирование — скучное занятие, а для исчерпывающего изучения все равно не хватит времени. Награда в миллион долларов уже вызывает интерес, но большинство компаний не могут позволить себе предложить такую сумму. Да и исследова¬

тель не имеет никакой гарантии в получении вознаграждения: он может ничего не найти или, смертельно устав от бесплодных попыток, проиграть кому-то другому, кроме того, компания способна изменить правила игры и ничего не заплатить. Неужели кто-то будет жертвовать своим временем (и рисковать добрым именем) ради рекламной акции какой-то компании?

И в-четвертых, состязания никогда не приводят к положительному результату в отношении безопасности. Если что-то было взломано, понятно, что это ненадежно. Но если цель устояла, это вовсе не означает, что она защищена.

Все эти четыре причины являются общим правилом. Бывают и исключения, но редко. Состязания по взлому RSA, как с помощью разложения на множители, так и путем лобовой атаки на симметричный алгоритм, — все это честные и хорошие соревнования. Эти соревнования успешны не потому, что исследователи борются за денежные призы, а вследствие того, что интерес к проблеме взлома этого алгоритма тем или иным способом присутствует всегда. Просто соревнования фокусируют внимание на том, что само по себе интересно. Конкурсы по взлому AES — больше соревнования, чем криптоаналитические расчеты, но они также были честными.

Состязания, если они правильно проводятся, могут принести пользу в отдельных областях исследований. Они помогают находить недостатки и исправлять их. Но они бесполезны для оценки безопасности. Хозяин может предложить 10 000 долларов тому, кто проникнет в его дом и украдет книгу с определенной полки. Но если никто не сделает этого в установленный срок, то это не значит, что дом надежно защищен. Возможно, никто из потенциальных взломщиков просто не слышал о конкурсе. Возможно, они были слишком заняты другими делами. Возможно, они не знали, как проникнуть в дом, но знали обходной путь: как подделать свидетельство о праве на недвижимость и перевести дом на свое имя. Может быть, они проникли в дом, но осмотрелись и поспешили убраться, прихватив нечто, стоящее больше 10 000 долларов. Поэтому состязания ничего не доказывают.

Состязания по криптоанализу — вообще не более чем реклама продукции. Даже честность устроителей не гарантирует того, что будет проведен анализ безопасности цели. Если цель устоит, это не будет означать, что нет недостатков в ее защите.

# Оценка и выбор продуктов безопасности

Обыкновенные люди (или обычная компания, или заурядное в этом отношении государство) вообще не способны создать свои собственные средства безопасности. Чаще всего они вынуждены выбирать между множеством готовых решений и надеяться на лучшее. Вывод, содержащийся в этой книге, о том, что практически невозможно разработать безопасные программные продукты и что большинство коммерческих продуктов являются небезопасными, звучит не ободряюще. Что может сделать измотанный системный администратор, занятый обеспечением безопасности электронной почты посольства или сети своей компании? Или простой гражданин, обеспокоенный безопасностью систем электронной торговли или сохранением в тайне медицинских сведений?

Первое, что приходит в голову, — а так ли уж это важно? Или, точнее, чья это проблема? Я беспокоюсь только о том, чтобы никто не вмешивался в мои частные

дела. Меня мало волнует возможность мошенничества с кредитными карточками Visa. Мои возможные потери ограничены 50 долларами. Я беспокоюсь о сохранении в тайне идентификационного номера моей банковской карточки, потому что если кто-то очистит мой счет, то это будет моей проблемой, а не банка.

Меня также заботят некоторые другие вещи, но я не могу на них повлиять. От меня не зависит, какие брандмауэры и средства защиты базы данных использует IRS (информационно-поисковая система) для защиты моей налоговой информации. Или что использует мой медицинский страховщик для защиты записей о состоянии моего здоровья. Возможно, я могу поменять страховщиков, но в общем я не свободен в своем выборе. (Полагаю, что если бы я был достаточно богат, то мог бы выбирать банки в лучше регулируемых условиях, например в Швейцарии, но большинству из нас это недоступно.) Даже если законы требуют соблюдения тайны (секретности, идентификации, анонимности и неприкосновенности и т. п.), все равно нет никакой гарантии, что люди, ответственные за обеспечение мер безопасности, выполняют свою работу хорошо. Я не могу проверить государственные средства безопасности только потому, что я хочу удостовериться в их эффективности. Печально, но факт — большинство аспектов обеспечения безопасности неподконтрольны простым людям.

Ради интереса давайте представим, что система безопасности находится под вашим контролем: Кроме того, вы несете финансовую ответственность за ее функциональность: вы потеряете деньги, если идентификационная схема будет взломана. Вам предъявят иск в случае нарушения защиты и утечки информации частного характера и т. д. Вы уже оценили риск и решили, что нужно приобрести средства безопасности определенного типа. Как выбрать правильный продукт? Как оценить его возможности?

Проблема состоит в том, что плохие средства безопасности выглядят точно так же, как и хорошие. Я могу предложить два продукта — пару VPN (виртуальных частных сетей), например. Они имеют одинаковые возможности и одинаковые особенности. В них встречаются одинаковые модные словечки: тройной DES (стандарт шифрования данных), IPsec и т. д. Они в равной степени удовлетворяют требованиям безопасности. Каждая сеть безопасна и каждая может быть взломана. Обычный пользователь не имеет никакой возможности увидеть разницу. Специалист в области безопасности сумеет это сделать, но уйдет полгода работы на то, чтобы он составил свое мнение. Это просто не оправдывает затрат.

Я постоянно поражаюсь газетным статьям, которые сравнивают средства безопасности и выставляют им оценки. Недавно я видел одну публикацию о брандмауэрах. Авторы пытались сравнить их надежность: в лабораторных условиях устанавливали различные брандмауэры и подвергали их воздействию 300 атак. Все это интересно, но имеет лишь весьма отдаленное отношение к тому, насколько надежен будет брандмауэр в реальной конфигурации и насколько эффективно он будет противостоять реальным противникам. Все, о чем говорилось в статье, так это о том, может ли брандмауэр, установленный в лаборатории, выдержать определенную атаку, а не о том, способен ли он укрепить безопасность сети. Легко сравнивать функциональные особенности, исследовать аспекты безопасности намного труднее. Я видел даже более ужасающие статьи, в которых средства безопасности

оценивались только на интерфейсе пользователя. По-видимому, авторы пытались *«что-нибудь* измерить», и интерфейс пользователя был единственной вещью, которую они могли наблюдать.

Но даже если они оценили уровень безопасности, будет ли эта оценка применима в том случае, когда вы используете продукт некоторым вполне определенным способом? Например, я не беспокоюсь, насколько надежной можно сделать отдельную операционную систему. Меня интересует, насколько она будет защищена 90 % времени в реальных ситуациях. Я тревожусь о том, насколько на ее работу могут повлиять ошибки внешнего окружения. Или насколько защищена будет система, после того как ее установит обыкновенный системный администратор. Вот в чем вопрос.

Легче опознать откровенно плохие продукты. Продукты, сопровождаемые заведомо невыполнимыми обещаниями вроде: «гарантирована защита от взлома» или «неподдающееся взлому шифрование», конечно, почти всегда оказываются ненадежными. Продукты, сопровождаемые невероятными псевдонаучными заявлениями о новых ошеломляющих технологических прорывах (обычно это касается технологий шифрования), почти всегда смертельно опасны. Другие предупреждающие сигналы — это ссылки на загадочных «специалистов по безопасности», использование ключей нелепой длины, отказ от общепризнанного опыта без серьезного на то основания (в отношении безопасности выгоднее следовать в общем потоке) и проведение фантастических состязаний по безопасности. В этой книге я обращаю внимание на то, что действительно хорошо на практике обеспечения безопасности: использование известной и опубликованной криптографии, использование открытых протоколов, выявление недостатков различных технологий. Компании, демонстрирующие незнание этих принципов, не заслуживают доверия. Конечно, может оказаться, что продукт, имеющий некоторые из этих предупреждающих знаков, надежен, только это маловероятно. Помните — гениев всегда намного меньше, чем дураков.

Пока что все было легко. Но после того как вы определили, какие продукты производятся компаниями, очевидно, не имеющими понятия о том, чем они занимаются, все становится намного сложнее. Все остальные продукты в равной мере изобилуют модными словечками, и все они сопровождаются правильными рассуждениями. Одни из них старше других, но значит ли это, что они более надежны? Об уязвимых местах одних было больше публикаций. Говорит ли это, что они менее укреплены, потому что в них обнаружено множество недостатков и, вероятно, будет найдено еще больше, или наоборот, — что они более надежны, потому что большее количество ошибок уже было найдено и исправлено? Нет никакой возможности узнать это. Вот почему так много компаний по безопасности используют рекламные приемы, свойственные юристам, ведущим дела пострадавших от транспорта, сея страхи, неуверенность и сомнения.

Опустить руки и отказаться от принятия решения — это не выход из положения. Существуют продукты безопасности, и потребители вынуждены выбирать из того, что имеется. Глупо отказываться от установки брандмауэра только потому, что вы не знаете, какой лучше. Кто-то сказал: «Лучше сейчас иметь посредственные средства безопасности, чем не иметь никаких, надеясь найти самые совершенные».

Правда в том, что испытание средств безопасности может только выявить существующие изъяны, но не доказать их отсутствие. Верно также и то, что в отсутствии недостатков убедиться невозможно, здесь не помогут ни попытки доказать безопасность системы, ни формальные модели безопасности, ни детальные деревья атак. Мы возвращаемся назад к исходной позиции, где мы были, когда выбирали алгоритм шифрования или протокол. Лишь продолжительные испытания, в которых участвуют многие люди, позволят нам начать доверять продукту безопасности.

Единственное, что можно сделать, — это осуществить такой процесс, который обеспечивает безопасность в любом случае, независимо от возможных изъянов продукта. Мы вернемся к этому в главе 24.

# Глава 23. Будущее программных продуктов

Напрашивается закономерный вопрос: «Какие технологии помогут созданию продуктов, обеспечивающих безопасность, в будущем?» Конечно, криптография постоянно улучшается. Несомненно, мы все лучше и лучше проектируем брандмауэры. Будет ли это помогать? Ответ: и да, и нет. «Да» — потому что отдельные технологии непрерывно совершенствуются. «Нет» — потому что фундаментальные проблемы остаются.

Технологии развиваются. Центральные процессоры стали намного быстрее работать, чем десять лет назад, что дает возможность применять шифрование почти повсеместно. Например, можно полностью зашифровать цифровую сотовую связь с помощью сильных алгоритмов без видимого замедления работы.

Технологии компьютерной и сетевой безопасности тоже совершенствуются. Сегодняшние брандмауэры намного эффективнее разработанных 10 лет назад. Системы обнаружения вторжения все еще находятся на ранней стадии развития, но и они постепенно улучшаются.

И это верно почти для каждой технологии, обсуждавшейся во части II книги. Технологии защиты от несанкционированного доступа становятся качественнее, то же можно сказать и о биометрических технологиях. Мы даже создали более эффективные механизмы защиты цифровых копий (несмотря на вскрытие DVD).

Но кое-что остается неизменным — основы технологий и люди, использующие их. Криптография всегда будет не больше, чем математика. Недостатки безопасности всегда будут присутствовать в программном обеспечении. Пользователи никогда не захотят запоминать длинные пароли. Люди будут всегда уязвимы для манипуляций.

Ситуация ухудшается. Системы становятся более сложными, и для компьютерных систем это означает нечто большее, чем аналогичные усовершенствования в любой другой области. Будущее цифровых систем — сложность, а сложность — главный враг безопасности. Безопасность не укрепляется, а становится более уязвимой.

## Сложность программного обеспечения и безопасность

Цифровая технология представляет собой нескончаемый ряд новшеств, непредсказуемых последствий и неожиданностей, и нет никакой причины полагать, что это прекратится в обозримом будущем. Единственная вещь, которая остается неизменной, — это то, что цифровые системы постоянно усложняются.

В течение последних нескольких лет мы наблюдаем этот процесс. Микропроцессоры стали более сложными. Операционные системы и программы тоже усложнились. Иногда без достаточно веской причины: существует целое моделирование виртуального полета, скрытое в каждой копии Microsoft Excel 97[1]. Компьютеры и сети стали более сложными. Появились сложные сетевые службы, загружаемые в память модули, программные агенты и распределенная обработка данных. Отдельные сети объединились, что привело к дальнейшему возрастанию сложности. Интернет, вероятно, — наиболее сложная система, когда-либо созданная человечеством. И она не станет более простой в ближайшее время.

Глобальная финансовая система стала более сложной. Цифровые системы в вашем автомобиле, посудомоечной машине и тостере стали сложными. Усложнились смарт-карты, а также сети, которые их обслуживают. Дверной замок вашего гостиничного номера, сигнализация, сотовые телефоны, система контроля состояния окружающей среды — все стало более сложным. Букин¬гемский Фонтан в Чикаго управляется удаленным компьютером, находящимся в Атланте.

С точки зрения потребителя, это замечательно. Появилось больше возможностей выбора. С точки зрения профессионала в области безопасности, это ужасающее. Сложность — главный враг безопасности. Эта истина известна с момента появления компьютера и, вероятно, будет верна и в дальнейшем. И поскольку киберпространство продолжает усложняться, безопасность будет становиться все более хрупкой. Тому имеется несколько причин.

Первая причина — количество изъянов систем защиты. В главе 13 я рассказывал о надежности программного обеспечения и о ее связи с безопасностью. По мере усложнения системы возрастает и количество ошибок при выполнении программ, и число уязвимых мест защиты. Это происходит всегда. При усложнении программного обеспечения возрастает и количество ошибок. И процент этих ошибок будет влиять на безопасность, хотя это не всегда очевидно.

Вторая причина — модульная структура сложных систем. В главе 10 говорилось о модульном коде и связанных с ним проблемах безопасности. Сложные системы обязательно модульные; нет никакого другого способа управиться со сложностью системы, кроме как разбить ее на части, поддающиеся управлению. Мы никогда не смогли бы сделать Интернет таким сложным и интересным, каким он является, не прибегая к модульности. Но разрастание модульной структуры грозит ослаблением безопасности, поскольку взаимодействие модулей создает дополнительные возможности для взлома защиты.

Третья причина — это взаимосвязь сложных систем. Распределенные и объединенные в сети системы опасны по своей природе. Усложнение систем может привести к тому, что незначительные проблемы становятся неразрешимыми, — повлечь

---

[1] Неудачный пример. «Пасхальные яйца» — неотъемлемая принадлежность большинства программных систем. Тот же симулятор полета в Excel 97 занимает ничтожную часть суммарного кода Office и скрыт от лишних глаз (переходим в ячейку X97:L97, нажав F5. Нажимаем Enter, Tab, Ctrl+Shift вместе со значком «Мастер диаграмм»). А до него — в Excel 95 — мы наблюдали встроенный Doom. В Excel 2000 мы можем поучаствовать в полноценных автогонках со стрельбой (предварительных операций нужно проделать еще больше). Списки разработчиков и даже фотографии есть почти в каждом программном продукте. Такие шалости появились одновременно с первыми программами и не могут служить аргументом в разговоре об усложнении программных продуктов. — *Примеч. ред.*

за собой эффект «бабочки»[1]. Мы уже наблюдали примеры того, как все становится доступным благодаря Интернету. В течение нескольких лет мы считали, что такие интернет-приложения, как почтовые программы, безопасны, но недавняя эпидемия макровирусов показала, что и Microsoft Word и Excel нуждаются в укреплении средств защиты. Апплеты Java должны не только быть безопасными в использовании, но также не допускать возможности применения их для нападения. При написании программного кода для веб-страниц используются тонкие взаимодействия между сценариями CGI, HTML, фреймами, программным обеспечением веб-сервера и cookies. В 2000 году баг Internet Explorer 5.0 блокировал правильную работу Windows 2000 при установке 128-битового шифрования. Фотокопировальные устройства, порты маршрутизаторов, RAID-массивы — ко всему этому можно получить доступ через Интернет, со всеми вытекающими отсюда угрозами для безопасности. Мошеннические драйверы принтера могут скомпрометировать Windows NT; файлы PostScript могут содержать вирусы. Вредоносный код, вложенный в сообщение электронной почты, может проникнуть через брандмауэр. Помните, я говорил о версии Windows NT, которая имела оценку безопасности C2, но только в том случае, если компьютер не подключен к сети и не имеет накопителя на гибких магнитных дисках? Помните WebTV-вирус?[2] Сколько пройдет времени, прежде чем кто-нибудь придумает вирус, поражающий сотовые телефоны?[3]

Четвертая причина в том, что чем сложнее система, тем она труднее для понимания. В главе 17 я рассказывал о манипуляциях людьми и о негативном влиянии человеческого фактора на безопасность. Эти проблемы обостряются из-за сложности систем. Люди, управляющие реальной системой, как правило, не обладают

---

[1] «Суть идеи прекрасно сформулирована в рассказе Рея Брэдбери "И грянул гром"... Начальные отклонения с течением времени нарастают, малые причины приводят к большим следствиям... взмах крыльев бабочки в неустойчивой системе может со временем вызвать бурю, изменить погоду в огромном регионе.» — С. П. Капица, С. П. Курдюмов, Г. Г. Малинецкий, «Синергетика и прогнозы будущего». — *Примеч. ред.*

[2] WebTV — телеприставка от Microsoft с выходом в Интернет, предоставляет дополнительные услуги — электронная почта, просмотр веб-страниц др., считается безопасной в отношении вирусов. В начале 2000 года в форумах и телеконференциях WebTV распространился Flood Virus, первый в мире для такого рода устройств, с механизмом размножения по принципу «Мелиссы» — путем изменения подписи под размещаемыми пользователями сообщениями. Засорение форумов и досок объявлений лавиной сгенерированных сообщений выводит их из строя. Microsoft, как всегда, все опровергала («Это не вирус. На WebTV не может быть вирусов»), кивая на злонамеренный программный код. Но в 2002 году был зарегистрирован новый представитель телевирусов — WebTV 911. Пользователи (ныне MSNTV), открывшие вложение в сообщение электронной почты, не предполагали, что тем самым позволяют вирусу перезагрузить модем для входа в командный режим, после чего вирус при помощи стандартной команды hayes-модемов (+++ATH0) набирал номер 911, а полиция не заставляла себя ждать. — *Примеч. ред.*

[3] В то время, когда печаталась эта книга, вирус SMS-Flooder уже поражал германские телекоммуникационные сети, круглосуточно затапливая их SMS-ссобщениями, пользуясь функцией саморассылки почтового клиента Microsoft; в Испании I-Worm.Timofonica рассылал SMS случайным пользователям сети MoviStar. Всего за несколько месяцев появился ряд вирусов, воздействующих на сети сотовой связи. В Норвегии некоторые такие SMS вызывали блокировку кнопок мобильных телефонов Nokia (вплоть до необходимости разрядки аккумулятора и замены SIM-карты). Это — самая малость того, что может сделать вирус, имеющий доступ к коммуникационным интерфейсам Windows (MAPI — почтовый и TAPI — телефонный), нещадно эксплуатируемым вирусами и закладками (запись переговоров, снятие денег со счета и т. д.). В интерфейс современных моделей телефонов («щелкни & звони») на радость хакерам уже заложена возможность самораспространения вирусов по принципу «Мелиссы». — *Примеч. ред.*

полным пониманием ее устройства, в том числе и проблем, связанных с безопасностью безопасности. А если кто-то плохо разбирается в вопросе, он, скорее всего, воспользуется помощью знающего человека. Сложность не только делает фактически невозможным создание безопасной системы, но и приводит к чрезвычайным трудностям в управлении ею.

Пятая причина — трудность анализа. В 18-21 главах я обрисовал процедуру проектирования и анализа систем безопасности: моделирование возможных угроз, определение механизмов защиты и проектирование системы безопасности. Чем сложнее система, тем тяжелее выполнить такой анализ. Все становится очень запутанным: спецификации, проект, создание и использование системы. Дерево атак для любой сложной системы становится гигантским. И, как мы уже неоднократно убедились, все это релевантно анализу безопасности.

Последняя, шестая причина — повышенные требования к испытаниям сложных систем. В главе 22 говорилось об испытаниях безопасности. Я доказывал, что единственно разумный способ исследования безопасности системы состоит в проведении тестирования непосредственно на ней. Однако чем сложнее система, тем труднее сделать такие оценки. Чем сложнее система, тем больше будет возникать ошибок, имеющих отношение к безопасности, и в спецификации, и в процессе разработки, и при вводе в действие. И, к сожалению, количество ошибок и трудности их распознавания растут не пропорционально возрастанию сложности, а намного быстрее.

Предельно упрощая, допустим, что система имеет 10 различных настроек, по 2 возможных варианта. Тогда 45 различных комбинаторных сочетаний могут взаимодействовать самым неожиданным образом, и в целом наберется 1024 различные конфигурации. Каждое взаимодействие способно привести к недостаткам безопасности и должно быть особо проверено. Теперь предположите, что система имеет 20 различных настроек. Это означает 190 различных сочетаний (по 2 из 20) и примерно 1 миллион конфигураций ($2^{20}$). 30 различных настроек определяют 435 различных пар и миллиард конфигураций. Даже небольшие увеличения в сложности системы означают стремительный рост количества различных конфигураций.

Увеличение числа возможных взаимодействий приводит к возрастанию объема работы во время оценки безопасности. Для системы с умеренным количеством параметров проверка всех двухпараметрических взаимодействий — тоже немалая работа. Проверка каждой возможной конфигурации — сложнейшая задача. Таким образом, трудность проведения оценки безопасности растет по мере увеличения сложности. Появление дополнительных потенциальных изъянов и усложнение анализа безопасности неизбежно приводит к уменьшению надежности систем.

В современных системах ситуация не столь сложна; часто параметры ортогональны, то есть независимы. Конечно, раз системы усложняются, то число связей увеличивается. Это происходит, например, если параметры находятся на разных уровнях в системе, и эти уровни разделены строго определенным интерфейсом. Такое разделение системы на относительно независимые модули с четко определенными интерфейсами — признак удачной разработки. Правильное разделение на модули может значительно уменьшить эффективную сложность системы, при этом все важные функции будут сохранены. Конечно, параметры в пределах одного модуля могут взаимодействовать, и эти взаимодействия должны быть проанализированы, поэтому количество параметров должно быть сведено к минимуму. Разбиение на модули работает хорошо при использовании должным образом, но многие

реальные системы все же имеют взаимозависимости, которые позволяют параметрам в различных модулях влиять друг на друга.

Более сложная система менее надежна с любой точки зрения. Прежде всего, она содержит большее количество уязвимых мест, а ее модульность усиливает эти недостатки. Ее тяжелее испытывать и анализировать.

Это ухудшает положение. Увеличение числа недостатков безопасности пагубно влияет на защиту: безопасность всей системы зависит от прочности ее самого слабого звена. Один-единственный недостаток может свести на нет защиту всей системы.

Реальные системы не показывают никаких признаков уменьшения сложности. Фактически они становятся более громоздкими все быстрее и быстрее. Microsoft Windows — пример такой тенденции. Windows 3.1, выпущенная в 1992 году, имеет 3 миллиона строк кода. В 1998 году Windows NT 5.0 насчитывала уже 20 миллионов строк кода, а в 1999 году она была переименована в Windows 2000 и содержала в среднем от 35 до 60 миллионов строк кода. Сравнительные данные приведены в табл. 23.1.

**Таблица 23.1**. Тенденция к сложности исходного кода

| Операционная система | Год | Количество строк кода |
|---|---|---|
| Window 3.1 | 1992 | 3 млн |
| Window NT | 1992 | 4 млн |
| Window 95 | 1995 | 15 млн |
| Window NT 4.0 | 1996 | 16,5 млн |
| Window 98 | 1998 | 18 млн |
| Window 2000 | 2000 | 35-60 млн (приблизительно) |

Windows 2000 ошеломляет своим размером, и будет иметь больше изъянов защиты, чем Windows NT 4.0 и Windows 98 вместе взятые. В свое оправдание Microsoft утверждала, что нужно потратить 500 человеко-лет, чтобы сделать Windows 2000 безопасной. Я привел эти цифры только для того, чтобы продемонстрировать, насколько неадекватна эта оценка[1].

Вы также можете видеть, что сложность увеличивается, по количеству системных вызовов. Версия UNIX 1971 года имела 33 вызова. В начале 1990-х их количество в операционных системах достигало уже 150. Windows NT 4.0 SP3 имеет 3433. Количество системных вызовов для различных операционных систем представлено в табл. 23.2.

Вначале брандмауэры имели дело только с FTP (протоколом передачи файлов), протоколами Telnet, SMTP, NNTP и службой DNS. И это все. Современные брандма¬

[1] Версия Windows XP в некоммерческом варианте включала в себя 50 миллионов строк кода. Критика компании возымела действие; «обнаружив неожиданные» проблемы с безопасностью, Билл Гейтс убедил разработчиков сконцентрироваться на задачах безопасности, и Microsoft была вынуждена перед выпуском официальной версии убрать лишние 5 миллионов строк и более 30 уязвимостей. Windows XP действительно продвинулась вперед в сторону безопасности, как признают эксперты, но все равно — 45 миллионов строк кода, в котором «все цепляется друг за друга» (определение Петера Ньюмэна из SRI International) — «это слишком много». По словам Вильяма Малика, аналитика безопасности IT consultancy Gartner, использовать встроенный в XP брандмауэр в качестве единственной меры защиты может оказаться ошибкой. Сокращение объема кода — явление временное, и общая тенденция именно в сторону роста. Открытое программное обеспечение гонится вослед за продукцией софтверного гиганта. Дистрибутив Linux Red Hat 7.1 насчитывает 30 миллионов строк кода. Из UNIX-систем рекордсменом по размеру является дистрибутив Debian GNU/Linux 2.2 — 55 миллионов строк. — *Примеч. ред.*

уэры должны взаимодействовать с сотнями протоколов и с запутанным набором правил доступа к сети. Некоторые новые протоколы разработаны подобно HTTP, чтобы они могли «работать с брандмауэром» (то есть обходить его). И пользователи, устанавливающие автоматическую связь, могут не беспокоиться о брандмауэре; теперь для этой цели существуют широкополосные аппаратно-программные средства с DSL и кабельными модемами. Хуже того, существует доступное программное обеспечение, позволяющее пользователям домашних компьютеров представлять себя в качестве веб-серверов. Больше особенностей, больше сложности, меньше надежности.

**Таблица 23.2.** Тенденция к сложности в операционных системах

| Операционная система | Год | Количество системных вызовов |
|---|---|---|
| UNIX Ied | 1971 | 33 |
| UNIX 2ed | 1979 | 47 |
| SunOS 4.1 | 1989 | 171 |
| 4.3 BSD Net 2 | 1991 | 136 |
| SunOS 4.5 | 1992 | 219 |
| HP UX 9.05 | 1994 | 163 |
| Line 1.2 | 1996 | 211 |
| SunOS 5.6 | 1997 | 190 |
| Linux 2.0 | 1998 | 229 |
| Window NT 4.0 SP3 | 1999 | 3 433 |

Сертифицированный открытый ключ в X.509 версии 1 был определен на 20 строках ASN.1. Сертификат X.509 версии 3 занимает приблизительно 600 строк. Сертификат SET — 3000 строк.

Полный стандарт SET занимает 254 страницы. И это только формальная спецификация протокола; есть еще руководство для программиста на 619 страницах и бизнес-описание на 72 страницах. По различным причинам SET никогда не будет широко использоваться, но в любом случае я полагаю, что никто не способен пробраться через все эти лабиринты, не наделав ошибок. Ошибки в работе программного обеспечения обнаруживаются главным образом во время бета-тестирования, но при этом недостатки защиты, скорее всего, не будут найдены. Тем не менее они там будут. Если их найдет порядочный человек, то он опубликует свои исследования в печати. Если их обнаружит мошенник, то он воспользуется ими для внедрения в систему кредитных карт: возможно, для получения крупной суммы денег или для создания поддельного счета на кредитной карте, возможно, затем, чтобы повлиять на обработку данных кредитных карточек и подчинить себе всю систему.

Сложность пробирается во все. В 2000 году вычислительные возможности «Мерседес 500» были больше, чем 747-200[1]. Мой старый термостат имел один наборный

[1] Дальнемагистральный самолет, модификация самого большого в мире пассажирского авиалайнера, легендарного «Боинга 747». Первый серийный «747-100» начал регулярные рейсы в 1971 году. Выпускался серийно в 1972-1988 годах. Всего было поставлено 393 самолета, последний — в 1991 году. Гордость Америки: на его основе ВВС США создали сеть воздушных командных пунктов управления и наведения; участвовал в операции «Буря в пустыне», перевезя 644 000 солдат и 220 000 тонн груза; использовался в качестве президентского самолета «Air Force One». — *Примеч. ред.*

диск, и было легко установить температуру. Мой новый термостат имеет цифровой интерфейс и руководство по программированию. Я гарантирую, что большинство людей понятия не имеют, как его заставить работать. Термостаты, основанные на системе Home Gateway от Sun Microsystems, могут быть подключены к Интернету, так что вы можете заключить договор с какой-нибудь компанией, занимающейся контролем окружающей среды, для управления вашим хитроумным устройством. Sun рисует в своем воображении связь с Интернетом для всех ваших приборов и дверных замков. Вы думаете, кто-нибудь проверит программное обеспечение рефрижератора на наличие изъянов защиты? Я уже рассказывал о современном злонамеренном коде и о взаимодействии сценариев Java, HTML, CGI и веб-браузеров. Есть кто-нибудь, кто беспокоится, что новые сотовые телефоны, работающие с Wireless Access Protocol, будут способны пересылать апплеты Java? И только вопрос времени, когда появится вирус, поражающий сотовые телефоны.

Компьютерные игры имели обыкновение быть простыми. Теперь можно играть, используя сеть. Любой в состоянии зайти на веб-сайт и принять участие в игре с несколькими игроками. Сейчас кто-нибудь еще может войти в систему компьютера игрока через Интернет. Просто, теперь он «сервер»! Мама и папа могут держать в компьютере некую конфиденциальную информацию (рабочие секреты, финансовые данные) а их отрок предоставляет всем возможность проникнуть внутрь. Кто-нибудь проверяет безопасность этих игр? Уязвимость в функции автоматического обновления игры Quake3 Arena позволяет нападающему модернизировать любой файл на компьютере пользователя. Napster также делает доступным ваш компьютер, после чего вы, скорее всего, найдете ошибки переполнения в программном обеспечении.

Положение ухудшается. Нынешнее поколение игровых приставок (Sega Dreamcast, Sony PlayStation 2) выпускается с такими свойствами, как 56-килобайтовые модемы, IP-стеки и веб-браузеры. Миллионы из них уже проданы. Возможно, браузеры и операционные системы будут безопасны; если — да, то такое случится впервые. Хорошая шутка: вы играете с кем-нибудь в Sonic через модем, и вдруг этот игрок проникает в ваш компьютер и побеждает! Если это только игровой пульт, то круто! Это не повод для волнения. Но не забывайте, что игровые компании требуют, чтобы вы делали все электронные покупки с помощью игрового пульта. Там будут и номера кредитной карточки, и электронный бумажник, и много чего еще. Добро пожаловать в мир, в котором ошибка переполнения буфера в Tekken 3 поставит под угрозу вашу финансовую безопасность.

Эта проблема становится всеобщей: сегодняшние игрушки будут завтра применены для опасных целей. Программное обеспечение для массового потребителя становится лучше при добавлении разных свойств и функциональных возможностей, но при этом проигрывает в надежности. Несмотря на сказанное и на факт, что программы и сети не были разработаны для опасных применений, но так или иначе, они используются для этого. Мы стали зависимы от систем, неизвестных в своей надежности. Скоропалительные решения вида «отправить, наконец, эту проклятую штуку» стали частью нашей критической инфраструктуры. Интернет и операционные системы — одни из самых наглядных примеров этого.

Конечно, ошибки безопасности обнаруживаются и исправляются, но это сизифов труд. Изделие «программное обеспечение» уже выпущено. Через какое-то время ошибки безопасности найдены и устранены, и безопасность улучшается. Затем

выходит версия 2.0 с новым кодом, с добавленными особенностями и увеличенной сложностью — и мы опять там же, откуда и начинали. Бывает и хуже.

У военных это называется «среда, насыщенная целями»[1].

В будущем системы с сетевой структурой будут сложными, а значит, менее стойкими. Индустрия технологий гонится за спросом на особенности, на возможности выбора, на скорость. Нет никаких стандартов по качеству или безопасности и нет никакой ответственности за опасное программное обеспечение. Следовательно, нет экономического стимула поддерживать высокое качество. Вместо этого есть экономическая подоплека для создания продукции самого низкого уровня, какой только выдержит рынок. И пока клиенты не потребуют «лучшее качество и безопасность», ситуация не изменится.

Я вижу две альтернативы. Первый вариант — это «притормозить», упростить и попытаться добавить безопасности. Клиенты не потребуют этого (такие проблемы слишком сложны для их понимания), поэтому это должна потребовать группа защиты прав потребителей. Я могу легко представить организацию для Интернета, подобную Управлению по контролю за продуктами и лекарствами (FDA), но в окружающей нас среде, где на утверждение рецепта нового лекарства требуется десятилетие, это решение экономически не выгодно.

Другой путь — признать, что цифровой мир будет «номером один» со сверхрасширяющимися особенностями и возможностями, со сверхбыстрым выпуском изделий, со сверхувеличивающейся сложностью и сверхскоротечно падающей безопасностью. Если мы признаем эту реальность, то сможем попытаться заняться делом вместо того, чтобы прятать голову в песок и отрицать проблему.

Я повторюсь: сложность — самый ярый враг безопасности. Безопасные системы должны быть «обрезаны до кости» и сделаны настолько простыми, насколько это возможно. И нет никакой замены этой простоте. К сожалению, простота противоречит основным тенденциям развития цифровых технологий.

# Новые технологии

Уже появились новые технологии, которые могут полностью изменить средства безопасности и в лучшую, и в худшую сторону. Так как эта книга не по предсказанию будущего, упомяну только несколько наиболее интересных достижений.

**Крупные достижения в криптографии.** Криптография лишь в некоторой степени основана на математических доказательствах. Лучшее, что мы можем сказать — то, что мы не можем ее взломать, так же как и другие ловкие люди, пытавшиеся сделать это. Всегда есть шанс, что когда-нибудь мы изучим новые методы, которые позволят нам взломать то, что недоступно сегодня. Как говорят в Агентстве национальной безопасности: «Атаки всегда совершенствуются, они никогда не ухудшаются». Мы наблюдали это в прошлом, когда, казалось бы, надежные алгоритмы были разрушены с помощью новых методов, и мы, вероятно, увидим это и в будущем. Некоторые люди даже предполагают, что в АНБ уже многое знают из

[1] Target-rich environment — обилие воздушных, наземных и морских целей, подлежащих уничтожению. — *Примеч. ред.*

этой новой математики и спокойно, с выгодой для себя взламывают даже самые сильные алгоритмы шифрования. Я так не думаю: может быть, они и владеют некоторыми секретными техниками, но не многими.

**Крупные достижения факторизации.** Вызывает опасение, что все алгоритмы открытых ключей базируются в основном на двух математических задачах: проблеме разложения на множители больших чисел (факторизации) или проблеме дискретного логарифмирования. Разложение на множители становится осуществлять все легче, и это происходит намного быстрее, чем кто-либо думал. Не доказано математически, что эти проблемы невозможно решить, и, хотя математики так не думают, может быть, кто-нибудь придумает способ эффективного решения этих проблем еще в наши дни. Если это случится, то мы окажемся в мире, где криптография открытых ключей не будет работать, и тогда главы этой книги будут представлять только исторический интерес. Но все не так страшно; инфраструктура аутентификации, основанная на симметричном шифровании, может служить для той же цели. Но в любом случае я не думаю, что это произойдет[1].

**Квантовые компьютеры.** Когда-нибудь квантовая механика фундаментально изменит способ работы компьютера. В настоящее время можно смело утверждать, что квантовые компьютеры способны сложить два 1-битовых числа, но кто знает, что будет дальше?[2] Возможно, если иметь в виду новые квантовые методы расчета, большинство алгоритмов открытых ключей покажутся устаревшими (см. предыдущий пункт), но на самом деле они всего лишь заставят нас удвоить длины ключа для симметричных шифров, хэш-функций и MACs (кодов аутентификации сообщения).

---

[1] В 17-й главе говорилось, что «Криптография с привлечением открытого ключа использует простые числа, и лишь в одном случае из миллиарда число может оказаться в действительности не простым». В криптографии доказательство того, что число — простое, до сих пор подразумевало некоторую вероятность ошибки. В 2002 году индийские математики (Технологический институт в Канпуре) закрыли проблему, решения которой не могли найти веками. По их утверждению, получаемые результаты абсолютно точны. О практическом применении пока нет разговора ввиду низкой скорости работы алгоритма, но все мировые авторитеты заинтересовались открытием Малиндры Агравала и внимательно изучают опубликованные выкладки. А если представить, что такой стимул, как приз в 1 миллион долларов, объявленный Clay Mathematics Institute (Кембридж, Массачусетс), подтолкнет решение гипотезы Римана (1859 году), предполагающей существование закономерности в распределении простых чисел? Это уже революция в передаче и хранении зашифрованных данных. Вера в стойкость дискретного логарифмирования и разложения на множители основывается на гипотетическом допущении сложности решения задачи и не имеет строгого доказательства. Сила материального стимулирования известна. В начале XVIII в. английское адмиралтейство объявило, что оно выплатит 20 тысяч фунтов стерлингов тому, кто найдет способ определять долготу местонахождения корабля в открытом море. Долго ждать не пришлось. Обещанные деньги получил в 1725 году английский часовой мастер Джон Гаррисон, который создал серию хронометров, погрешность которых составляла несколько сотых долей секунды в сутки. Стратегически важное изобретение способствовало тому, что Англия стала владычицей морей. — *Примеч. ред.*

[2] Ученые считают, что время квантовых компьютеров придет около 2020 года, когда согласно закону Мура размер электронных схем достигнет физического предела (будет сравним с размерами атомов и молекул). Благодаря присущему ему по природе параллелелизму вычислений квантовый компьютер сверхпроизводителен. В 2000 году (после выхода этой книги) исследователи из IBM провели вычисления на модели, состоящей из 5 атомов, работающих одновременно в режиме процессора и памяти. Экспериментальный образец использовался для решения некоторых математических задач криптографии (нахождения периода функций) и продемонстрировал скорость, заметно превышающую производительность традиционных ЭВМ. Сегодня АНБ и Министерство обороны США щедро финансируют разработки квантового компьютера в Стэнфордском университете. — *Примеч. ред.*

**Аппаратные средства сопротивления вторжению.** Многие проблемы безопасности устраняются, если использовать для защиты информации аппаратные средства сопротивления вторжению. Бурное развитие технологий привело к значительному уменьшению стоимости средств защиты от несанкционированного вмешательства[1].

**Искусственный интеллект.** Работа многих защитных средств может быть сведена к выполнению простой задачи: пропускать полезное содержимое, задерживая при этом вредоносное. Так работают брандмауэры, системы обнаружения вторжения, антивирусное программное обеспечение, VPN, системы защиты от мошенничества с кредитными картами и множество других вещей. Они подразумевают два подхода: неосмысленный (если есть любой из десятков тысяч образцов поиска в файле, следует, что файл заражен вирусом) или интеллектуальный (если программа начинает делать подозрительные вещи, скорее всего, завелся вирус, и вы должны все исследовать). Последнее кажется внушающим ужас, вызывает пугающие ассоциации с искусственным интеллектом. Что-то подобное было испытано как антивирусный механизм и в конце концов оказалось менее эффективным, чем примитивная модель проверочного сканера. Подобные идеи моделируются в некоторых продуктах обнаружения вторжения, и все еще неясно, делают ли они что-нибудь лучше, чем методичный поиск сигнатур, которые свидетельствуют о внедрении[2]. Однако когда-нибудь это может стать большим делом: если в области искусственного интеллекта со временем произойдет существенный прогресс, мы получим революцию в компьютерной безопасности.

**Средства автоматической проверки.** Многие дефекты системы безопасности, такие как ошибки переполнения буфера, — результат небрежного программирования. Хорошие автоматические средства, которые могут просматривать коды ком-

---

[1] В новом тысячелетии ученые Калифорнийского университета в Сан-Диего нашли способ взрывать кремний, пропустив через него электрический ток. Электрический сигнал посылается в часть микросхемы, которая содержит небольшое количество нитрата гадолиния, для инициирования процесса окисления кремния. Предполагается, что изобретение будет использоваться для защиты украденных компьютеров от несанкционированного доступа, при этом изменения в производстве микросхем не потребуют принципиальных технических новшеств. — *Примеч. ред.*

[2] Позиция Шнайера слишком негативна. Как бы быстро ни работали антивирусные полифаги и сканеры и как бы ни были велики их базы данных, сложилась ситуация, когда разработчики уже просто не успевают обработать и внести в базу данных все сигнатуры. Это осложняется наличием генераторов вирусов с доступным любому интерфейсом и способных видоизменяться вирусов (полиморфных). Теоретически доказано, что множество всех вирусов не поддается перечислению и что нельзя создать универсальный детектор, способный отличить «чистую» программу от зараженной (Л. Адлеман и Ф. Коэн соответственно). Именно поэтому интеллектуальные средства — будущее антивирусных систем, как бы это ни было «ужасно». Шнайер говорит об *эвристических анализаторах,* позволяющих выявлять вирусы, сигнатура которых до того была неизвестна. Они просматривают не сигнатуры, а программный код в поисках подозрительных действий и выставляют итоговую оценку. При этом вероятность пропуска и ложного срабатывания велика, также зараженную программу нельзя «вылечить». Поэтому, несмотря на всю полезность, интеллектуальные средства вызывают недовольство пользователей. Может, ради комфорта последних или в силу иных причин, но только малая капля зарубежных антивирусных систем имеет в своем арсенале эвристические анализаторы. Поэтому лидеры здесь — российские программы, из которых пионером (1994 год) и наиболее мощной является Dr. Web, умеющая обнаруживать неизвестные полиморфные и макровирусы. Но ни Dr. Web, ни «Антивирус Касперского» не умеют самообучаться, пополняя собственные вирусные базы (адаптивные средства), и многое другое. Они страшно далеки от искусственного интеллекта, но при этом перспективны для своей области. — *Примеч. ред.*

пьютера с целью обнаружения изъянов защиты, должны пройти длинный путь, чтобы они сумели обеспечить хорошую безопасность. Хорошие компиляторы и хорошая программа синтаксического контроля также проходят долгий путь, и когда задачи безопасности выйдут в них на первый план, только тогда они действительно смогут помочь программистам избежать ошибок. Кроме того, нужно убедить программистов использовать эти новшества, однако это уже другой вопрос. (Имеется ряд хороших средств, но они почти никем не затребованы.) И это отдельная проблема.

**Безопасные сетевые инфраструктуры.** Интернет не безопасен, безопасность никогда не стояла в планах для этой системы. Люди, которые будут работать в Интернете II, должны в первую очередь подумать о безопасности. При создании этой новой сети следует допустить, что кто-то будет перехватывать и пытаться похищать сетевые соединения и что заголовок пакета данных, возможно, будет подделан[1]. Нужно учесть, что пользователи в большинстве случаев не могут доверять друг другу, и рассматривать с этой точки зрения все виды приложений. Существует множество проблем, которые нельзя разрешить с помощью самых лучших протоколов сети, но некоторые можно.

Анализ трафика. Технология анализа трафика все еще находится в периоде становления, и я думаю, что новые интересные достижения в этой области появятся в следующем десятилетии[2]. Методы защиты трафика должны еще долго совершенствоваться, чтобы они могли обеспечить должный уровень защиты любой сети.

Обеспечение надежности. Надежность означает, что система делает то, что ей положено делать, и ничего лишнего. Технология, которая смогла бы обеспечить надежность программ, была бы незаменимой для компьютерной безопасности.

Большинство подобных технологий все еще находятся в стадии разработки. Я не буду делать скидок ни для одной из них. Практические результаты если и появятся, то, скорее всего, только в далеком будущем. Хотя если чему-то и научило нас двадцатое столетие, то это — как можно реже употреблять слово «невозможно».

[1] Проект Internet II был организован ведущими университетами США (сейчас около 200) из-за сложности проведения совместных научных исследований, возникших вследствие передачи университетских сетей в частные руки и недостаточной пропускной способности каналов связи. Его основная цель — увеличение доступности научной и образовательной информации. Финансирование проекта государством минимальное, Internet 2 «живет» за счет университетских вложений общей суммой 80 миллионов долларов в год. В контексте безопасности актуальнее говорить о параллельной федеральной организации «Интернет следующего поколения» (Next Generation Internet, NGI), в бюджет которой заложены большие инвестиции и которая в большей степени предусматривает разработку и тщательное тестирование новых сетевых услуг и технологий, в отличие от Internet 2, развивающего Ipv6 (где контроль заголовков пакетов заложен изначально) и QoS. Проект не является физической сетью и не может сам по себе подменить Интернет. — *Примеч. ред.*

[2] Системы интеллектуального распознавания Data Mining («добычи знаний») не новость. Однако специалисты Директората науки и техники ЦРУ после многолетних исследований сумели применить их в глобальном масштабе, и в начале 2001 года представили специализированное программное обеспечение технологии Text Data Mining, позволяющее распознавать живую речь, аудио и видео, графическую информацию и текст на 35 основных языках. Программы позволяют обрабатывать огромный трафик информационных потоков в Интернете, базы данных и тексты, автоматически анализировать разговорную речь, реагируя на ключевые слова и выражения (система Fluent), переводя ее в текст при помощи технологии Oasis. Возможно определение принадлежности голоса по половому признаку (на английском), его идентификация, определение темы разговора. Oasis уже применяется ЦРУ в странах Центральной Азии. — *Примеч. ред.*

# Научимся ли мы когда-нибудь?

Рассмотрим нападения, приводящие к переполнению буфера. Впервые о них заговорили в сообществе безопасности в начале 1960-х (система с разделением времени пострадала от такой атаки), и, вероятно, они описывались в литературе по безопасности еще раньше. В 70-е годы первые компьютерные сети имели много недостатков, которые часто использовались для нападения. В 1988 году компьютерный червь Морриса вызывал переполнение буфера, используя команду fingerd UNIX (распространенный способ нападения этого типа). Теперь, через 10 лет после появления червя Морриса и примерно через 35 лет после первоначального обнаружения такого способа нападения, можно было бы предполагать, что сообщество безопасности наконец-то решило проблему защиты буфера. Но это не так. В 1998 году более чем две трети всех обращений в CERT были связаны с проблемами, вызванными переполнением буфера. В 1999 году в течение двух особенно неудачных для Windows NT недель, в NT-приложениях было обнаружено 18 различных изъянов, открывающих дорогу для такой атаки. Когда в первую неделю марта 2000 года я стал собирать факты, с описания которых начал эту книгу, среди них мне встретились три случая ошибки переполнения буфера. Пример с переполнением буфера — это низко висящий плод. Но если мы когда-либо и научимся решать эту проблему, тут же появятся другие, возможно, еще более сложные.

Рассмотрим алгоритмы шифрования. Закрытые секретные алгоритмы появляются постоянно и затем вскрываются самым банальным образом. Каждый раз мы убеждаемся, что закрытые секретные алгоритмы — плохая идея. Тем не менее многие компании в различных отраслях промышленности продолжают выбирать частные, закрытые алгоритмы взамен открытых.

Обратимся к проблемам исправления ошибок. В начале 2000 года всего одна прореха в защите Microsoft Internet Information Server помогла хакерам украсть тысячи номеров кредитных карточек с разных сайтов электронной торговли. Microsoft выпустила «заплату», в которой эта ошибка была исправлена еще в июле 1998, и еще раз напомнила о необходимости установить эту «заплату» в июле 1999, когда стало ясно, что многие пользователи и не беспокоятся по поводу установки исправления.

Кто-нибудь обращает внимание на подобные вещи?

На самом деле никто. Или, по крайней мере, совсем немногие. Поскольку потребность в средствах защиты программных продуктов огромна, их появляется все больше. А постоянная нехватка действительно компетентных экспертов приводит к тому, что все меньшее количество людей уделяют внимание проблемам безопасности.

Можно описать сценарий, по которому происходит проектирование большинства программных продуктов, имеющих средства защиты «внутри». Менеджер находит какого-нибудь парня, который считает, что безопасность — «это круто», и назначает его ответственным за эту часть системы. Этот человек может кое-что знать о безопасности, а может и не знать. Может, он прочитал одну-две книги по этой теме, а может быть, и нет. Проектирование системы защиты — вообще забавная вещь. Это как игра в «кошки-мышки». Или как в кино: агент против агента.

Проверка системы защиты происходит так же, как и проверка любой другой части системы: вы смотрите, как это работает, и делаете выводы. Все работает великолепно — в конце концов, безопасность не имеет никакого отношения к функциональным возможностям, — и менеджер счастлив.

Однако, поскольку уровень экспертизы систем безопасности вообще низок, оказывается, средства защиты совершенно неэффективны. Но никто не догадывается об этом.

Ситуация несколько лучше, если программный продукт является средством защиты. Более вероятно, что проектировщики что-то смыслят в безопасности. Но они не могут делать все. Однажды разработчик брандмауэра рассказал мне о недостатках в его коде, приводящих к переполнению буфера. Он сказал, что сделал все возможное, чтобы гарантировать отсутствие подобных вещей, но он не мог проверять всех остальных программистов в команде. Он пробовал, но не смог. Несколько серьезных недостатков в коде, связанных с переполнением буфера, были обнаружены и исправлены за последующие годы. И нет оснований предполагать, что необнаруженных ошибок больше не осталось.

Меня бесконечно удивляют всяческие дырки в защите, используя которые, нетрудно ее взломать. Я видел продукт, осуществляющий шифрование файла, который случайно сохранил ключ в незашифрованном виде. Я видел VPN, где файл конфигурации, отвечающий за телефонную связь, случайно позволяет лицам, не обладающим соответствующими правами, пройти процедуру аутентификации, или такие, где один клиент VPN может видеть файлы всех других клиентов. Существует огромное количество способов сделать продукт ненадежным, и производители ухитряются натыкаться на одни и те же грабли снова и снова.

Они ничему не учатся, потому что и не должны учиться.

Средства защиты компьютера, так же как и программное обеспечение в общем, обычно соответствуют довольно странным представлениям о качестве. Здесь ситуация другая, чем в случае с автомобилем, небоскребом или упаковкой с жареным цыпленком. Если вы покупаете какую-либо вещь и терпите урон по вине изготовителя, вы можете предъявить иск и выиграть дело. Производителю автомобилей не избежать неприятностей, если автомобиль взорвется при столкновении. Буфетчик не останется без проблем после продажи земляничного пирога с крысой внутри. Чего никогда не скажут подрядчики-строители, так это: «Ну что ж. Мы скоро построим второй небоскреб, и уж он не упадет на все 100 %». Эти компании ответственны за свои действия.

Программное обеспечение бывает разное. Оно продается без какой-либо ответственности вообще. Например, в лицензионном соглашении Windows 98 есть следующие слова: «Изготовитель и его поставщики отказываются нести материальную ответственность за какие-либо убытки, вытекающие из использования или невозможности использования данного изделия, даже в том случае, если Изготовитель был предупрежден о возможности этих убытков».

Ваша база данных взаиморасчетов с партнерами может рухнуть, увлекая за собой вашу компанию, а вы не сможете предъявить никаких претензий к производителю программного обеспечения. Ваш текстовый процессор может испортить законченную рукопись вашей книги (из-за чего я часто беспокоюсь), уничтожив несколько лет вашей работы, и вам некуда будет обратиться за помощью. Если

ваш брандмауэр вдруг окажется совершенно бесполезным, в этом будете виноваты только вы сами. Microsoft в состоянии пропустить дефект в Hotmail, который будет использован кем-нибудь для получения сорока миллионов или около того учетных записей, защищенных паролем или нет, и даже не потрудится принести извинения.

Производители программного обеспечения могут и не беспокоиться о качестве своих продуктов, так как не несут за это никакой ответственности. (То есть ответственность существует, но она ограничивается заменой гибкого диска или компакт-диска с физическими дефектами.) Поэтому средства защиты могут и вовсе не обеспечивать безопасность, так как никто не сумеет предъявить иск их производителям, обманувшим покупателя лживыми заверениями.

В конце концов, эта ситуация приводит к тому, что рынок не поощряет создание действительно безопасных продуктов. Их разработка требует больше труда, времени и средств. Покупатели же не знают, как отличить надежный продукт от негодного. На этом рынке можно добиться успеха, выпустив программное обеспечение, о надежности которого производитель позаботился лишь постольку, поскольку это ему необходимо, чтобы избежать неприятностей.

Крупные производители знают это, как и то, что создавать надежное программное обеспечение не выгодно. Согласно исследованиям, 90-95 % всех программных изъянов безобидны, они не сказываются на работе программ, и пользователи их не замечают. Намного дешевле выпускать программное обеспечение с дефектами и затем исправлять обнаруженные 5-10 % ошибок.

Они также знают, что невыгодно заниматься обеспечением реальной безопасности. По несколько раз в неделю они сталкиваются с новыми уязвимыми местами в своих продуктах. Они устраняют те, что им под силу, а о тех, с которыми справиться не могут, делают лживые заявления для печати, и ждут, когда бум в прессе утихнет. Затем они выпускают новую версию программного обеспечения с новыми функциональными возможностями, которые сопровождаются полным набором новых погрешностей, потому что пользователи предпочитают «наворо-ченность» безопасности.

И пользователи будут их покупать. До тех пор пока закон не будет побуждать производителей к созданию безопасных продуктов, они не станут беспокоиться об этом.

# Глава 24. Процессы безопасности

В 1996 году была клонирована шотландская овца по имени Долли. В последовавшей за этим событием дискуссии в прессе «Тайм» и «Ньюсуик» высказывали мнение, что клонирование людей безнравственно, и необходимо законодательными мерами пресечь возможность проведения подобных экспериментов. Но все это бессмысленно. Кто-нибудь все равно попытается клонировать людей независимо от того, законно это или нет. Нужно принять это как неизбежность и попытаться понять, что с этим делать.

В компьютерном мире также невозможно обеспечить какие-либо гарантии. Технологические приемы могут отразить большинство случайных атак. Угроза уголовного наказания может удержать от преступления многих, но не всех. Попытки нарушить чьи-либо права все равно будут продолжаться. Сети будут взламываться. Мошенничество будет процветать. Деньги будут красть. Люди будут гибнуть.

Технология не является панацеей. Программные продукты всегда имеют недостатки, число которых постоянно увеличивается. Единственный выход состоит в том, чтобы смириться с реальностью и научиться жить в этих условиях. Точно так же, как и в любой другой сфере общественной жизни. Никакие технические приемы не могут защитить нас от террористических актов. Мы стараемся обеспечить свою безопасность с помощью доступных нам средств, таких как таможенный контроль, сбор информации об известных террористических группах, неотвратимое уголовное преследование.

## Принципы

### Разделяйте

Бывалые путешественники держат лишь небольшую сумму денег в бумажнике, а остальное — в мешочке, скрытом под одеждой. Таким образом, если их обворуют, они не потеряют все деньги. Структура шпионских или террористических организаций предполагает разделение на маленькие группы, члены которых знают только друг друга и никого более. Таким образом, если кто-нибудь оказался схвачен или сдался сам, он может выдать только тех людей, которые входят в его ближайшее окружение. Разделение — эффективный способ защиты, так как оно ограничивает последствия действий противника, приводящих к успеху. Это — пример здравого смысла, которым часто пользуются. Пользователи имеют индивидуаль¬

ные учетные записи; двери помещений офиса запираются разными ключами; права доступа предоставляются соответственно степени доверия конкретному лицу, а также реальной необходимости предоставления ему определенных сведений; личные файлы шифруют уникальными ключами. Система безопасности не строится по принципу «все или ничего», но она должна предотвращать возможность причинения значительного ущерба.

Схожий принцип — минимум привилегий. В основном это означает, что нужно давать кому-нибудь (пользователю или некоторым процессам) только те привилегии, которые необходимы для выполнения задачи. Мы постоянно сталкиваемся с этим в повседневной жизни. Ваш ключ, скорее всего, подходит только к вашему, а не любому помещению в здании. Доступ к банкоматам и хранящимся в них деньгам имеет только обслуживающий их персонал. Даже если вы пользуетесь особым доверием на службе, вы сможете выболтать только те секреты, которые вам позволено знать.

Еще больше примеров можно найти в компьютерном мире. Пользователи имеют доступ только к тем серверам, которые нужны им для работы. Только системный администратор имеет доступ к системе в целом, пользователи имеют доступ лишь к своим файлам. Иногда отдельные файлы бывают защищены групповым паролем, известным только тем, кому необходим доступ к этим файлам. Конечно, легче предоставить каждому полный доступ, но безопаснее давать людям только те привилегии, в которых они действительно нуждаются. Системы предоставления прав доступа в UNIX и NT основаны на этом принципе.

Многие нападающие пользовались нарушением принципа минимума привилегий. Как только нападающий получает доступ к учетной записи пользователя, вскрывая пароль или как-нибудь иначе, он предпринимает несколько попыток получить высшую привилегию. Например, многие нападающие пытаются вскрыть «песочницу» Java (sandbox), получить таким образом минимальные привилегии, а затем перейти в режим, позволяющий получить привилегированный статус. Взлом защиты цифровых дисков, проездных карточек и систем платного телевидения и других, имеющих одно общее свойство — все секреты хранятся в устройстве, находящемся в распоряжении пользователя, — также можно назвать получением высших привилегий.

Разделение также важно, потому что чем больше людей пользуются системой, тем меньше ее надежность. Чем крупнее компьютер, чем шире круг задач, решаемых с его помощью, тем он менее безопасен.

Это одна из причин, почему Интернет, наиболее широко используемая сеть, таит столько опасностей. Сравните веб-сервер и компьютер, работающий в режиме пониженного потребления мощности, находящийся в запертом бомбоубежище и окруженный охраной. Использование разделения делает систему более похожей на второй вариант.

## Укрепите самое слабое звено

Прежде всего следует защитить самое слабое звено. Это очевидно, но снова и снова я встречаю системы, в которых это правило игнорируется. Было бы наивностью просто вкопать огромный кол перед воротами замка и надеяться, что враг побежит

прямо на него. Защита должна быть со всех сторон, поэтому придется выкопать ров и построить частокол. Точно так же при использовании алгоритма шифрования с 256-битовым ключом не стоит надеяться, что вы в безопасности; враг, вероятно, найдет такой способ нападения, который никак не связан с алгоритмами шифрования.

Меня постоянно удивляет, как много зияющих дыр остается в коммерческих системах безопасности. Разработчики не замечают их, так как поглощены созданием защиты тех частей системы, в которых они хорошо разбираются. Изучите ландшафт уязвимых мест в целом, сконструируйте схему нападений, найдите в ней самое слабое звено и обезопасьте его. Затем переходите к следующему слабому звену. Таким образом вы, вероятно, сможете построить наиболее безопасную систему.

## Используйте пропускные пункты

Пропускной пункт представляет собой узкий коридор, в котором легко контролировать пользователей. Вспомните, как устроены и для чего предназначены турникеты на вокзале, контрольно-кассовые пункты в супермаркете и двери вашего дома. Для этих целей используются брандмауэры, маршрутизаторы, регистрация при входе в систему, а некоторые веб-сайты направляют пользователей сперва на домашнюю страницу. Так же устроена система обнаружения мошенничества с кредитными картами. Всегда имеет смысл использовать пропускные пункты в целях безопасности.

Но эти пункты полезны только в том случае, если нет никакого способа обойти их. Один из обычных способов преодоления брандмауэра состоит в том, чтобы обойти его: можно найти, например, незащищенное удаленное соединение. Случается, что люди оставляют удаленное соединение включенным. Иногда маршрутизаторы, запоминающие устройства большой емкости и даже принтеры могут иметь незащищенные порты. Все это позволяет нападающим обходить пропускные пункты.

В сетях возможны более тонкие способы нападений такого типа. Одна компания может иметь надежную систему сетевой безопасности. А другая компания может не обеспечивать ее должный уровень. Если они взаимодействуют по сети, это означает, что сеть имеет слабое звено, которое нуждается в защите.

## Обеспечьте глубинную защиту

Глубинная (многоуровневая) защита — другой универсальный принцип безопасности, который применяется в области компьютерных технологий, так же как и в других областях.

Защита территории (дверные замки и сигнализация на окнах) более эффективна, если она используется совместно с системой слежения внутри дома. Система защиты кредитных карточек от мошенничества работает лучше, если она дополняется проверкой подлинности и системой отслеживания подозрительных расходов. Брандмауэр в сочетании с системой обнаружения вторжения и сильной криптографической защитой приложений будет намного более надежным, чем просто брандмауэр.

На протяжении всей книги я пытался донести до читателя мысль, что защита настолько надежна, насколько прочно ее самое слабое звено, и предыдущее утверждение, кажется, противоречит этому принципу. В действительности все зависит от исполнения. Вспомните схемы нападений: безопасность группы узлов ИЛИ определяется надежностью защиты самого слабого узла, в то время как в случае конечных узлов И защита каждого из них усиливает защиту их совокупности.

Два брандмауэра, каждый из которых защищает отдельную точку входа в сеть, — это не глубинная защита. Для успешного нападения достаточно преодолеть любой из этих брандмауэров. Защита по глубине будет реализована в том случае, если брандмауэры установлены последовательно один за другим: тогда нападающему придется по очереди иметь дело с двумя уровнями защиты. Меня всегда поражает, когда я вижу сложные сети, точки входа в которые порознь защищают брандмауэры различных марок или даже различные конфигурации одного и того же брандмауэра. Это совершенно лишено смысла.

## Подстрахуйтесь на случай отказа

Многие системы устроены таким образом, что если система выходит из строя, пользователь обращается к резервной системе, хоть и менее безопасной. Например, в Соединенных Штатах система VeriFone используется при совершении сделок по кредитной карточке. Когда клерк проверяет вашу карту, VeriFone обращается к базе данных и проверяет, не украдена ли карта, достаточно ли денег у вас на счете и т. д. Вспомните случаи, когда терминал не работал по какой-либо причине: или он был сломан, или прерывалась телефонная связь. Разве торговец отказывался обслуживать вас? Конечно, нет. Он вытаскивал бумажные бланки и оформлял сделку по старинке.

Многие нападающие пытаются взломать защиту «кавалерийской атакой»: атаки, приводящие к отказу в обслуживании, могут повторяться снова и снова. Я уже говорил о грабителях, которые заставляют постоянно срабатывать сигнализацию для того, чтобы ее в конце концов отключили. Есть и более тонкие способы нападения. Немногие люди дисциплинированы настолько, чтобы не воспользоваться обычной связью, когда они не могут пользоваться безопасной. Даже военные, которые, как принято считать, подходят к этому серьезно, постоянно повторяют подобные ошибки.

Нужно, чтобы системы в случае отказа становились более защищенными, а не менее. Если в банкомате перестанет работать система проверки личного идентификационного номера, произойдет отказ, но при этом не посыплются деньги из щели. Если сломается брандмауэр, то это приведет к тому, что он перестанет пропускать любые пакеты данных. Когда повреждается слот-машина, она не должна проливать монетный дождь на поднос выплаты.

Тот же самый принцип используется в технике безопасности и называется *отказоустойчивостью* (безаварийный отказ). Если выйдет из строя микропроцессор в автомобиле, это не должно привести к разгону до максимальной скорости. Если произойдет отказ в системе управления ядерной ракетой, это не должно привести к ее запуску. Безаварийный отказ — хорошее правило проектирования.

## Используйте непредсказуемость

Снова и снова в этой книге я выступаю против безопасности, основанной на секретности: закрытая криптография, закрытый исходный код, секретные операционные системы. Секретность так же может иметь место, но не в продуктах, а в том, как они используются. Я называю это *непредсказуемостью*.

Одно из преимуществ защищающихся перед нападающими состоит в знании местности. Армия не передает врагу сведений о расположении своих танков, установок противовоздушной обороны и батальонов. Нет никакой причины сообщать каждому интересующемуся сведения о топологической схеме вашей сети. Слишком многие компьютеры в ответ на любой запрос выдают информацию об используемой операционной системе и номере версии. Этому нет разумного обоснования. Будет намного лучше, если при регистрации пользователя в системе на экране появится сообщение: «Предупреждение: частный компьютер. Использование этой системы предусматривает контроль безопасности. Все действия пользователя и сведения о нем регистрируются, включая IP-адрес и имя хоста». Предоставьте нападающим самим разбираться, можете ли вы следить за ними.

Если вы строите частную систему безопасности (например, электронных банковских платежей), то важно использовать сильный, открытый, надежный алгоритм шифрования. Когда вы сделаете свой выбор, незачем объявлять о нем.

Если вы используете брандмауэр, нет никакого смысла сообщать всему миру настоящее имя хоста и имя пользователя. Это также принцип построения системы мер безопасности (сетевой сигнализации, приманок и других). Администратор сети знает, как она работает и что означают те или иные действия. Когда кто-то копается в фиктивном счете, администратор знает, что это — злоумышленник. Нападающий не должен знать, где и какое оборудование работает, когда и какой протокол можно использовать, какие порты и при каких условиях будут открыты. Поразительно, как часто серверы, приложения и протоколы объявляют на весь мир: «Привет! Я — служба V2.05». Многие автоматические средства взлома в поисках определенных версий программного обеспечения просматривают машины, о которых известно, что у них имеются уязвимые места. Если сети будут непредсказуемы, нападающие не смогут так свободно их огуливать. Не располагая нужной информацией, намного труднее обозначить цель и определить возможные виды атак. Разница такая же, как между прогулкой по солнечном лугу в полдень и по зарослям колючего кустарника в полночь.

Столь же непредсказуемы должны быть и меры реагирования. Ракета «Патриот» недостаточно эффективно поражала в небе иракские «Скады», но вы никогда не узнали бы об этом из официальных сообщений Пентагона. Если Соединенные Штаты знали, насколько была неэффективна противовоздушная оборона, это не означало, что нужно сообщать об этом врагу.

Непредсказуемость — мощное средство, используемое террористами, специалистами по «промыванию мозгов» при авторитарных режимах и теми, кто еще только рвется к власти. Но оно также хорошо работает в сфере компьютерной безопасности.

## Стремитесь к простоте

Я уже говорил в предыдущей главе: сложность — злейший враг безопасности. Система настолько защищена, насколько защищено ее самое слабое звено. Поэтому систему с меньшим количеством связей легче обезопасить. Эйнштейн говорил: «Все должно быть настолько просто, насколько это возможно, но не более».

Народная мудрость «Не кладите все яйца в одну корзину», казалось бы, противоречит этому утверждению. Так оно и есть, и на этом принципе строится защита по глубине. Но помните, что охранять несколько корзин сложнее, чем одну. Когда речь идет о безопасности, лучше следовать совету одного героя Марка Твена: «Сложите все яйца в одну корзину и *присматривайте за ней*».

## Заручитесь поддержкой пользователей

Обеспечить безопасность намного легче, если приходится иметь дело с надежными и разумными пользователям, и намного тяжелее — с невежественными и имеющими злой умысел. Меры безопасности, не понятые и не согласованные к каждым, не будут работать. Помните: самыми трудными проблемами безопасности являются те, которые связаны с людьми, а самыми легкими — те, что связаны с битами. Несомненно, должна существовать защита от нападений изнутри, но в основном ваши сотрудники — это ваши союзники. Заручитесь их поддержкой, насколько это возможно.

## Обеспечьте гарантию

В чем мы действительно нуждаемся — так это в уверенности, что наши системы работают должным образом, что они обладают требуемыми свойствами и только ими. Большинство нападений в реальном мире приводит к тому, что системы либо перестают делать то, что им положено, либо начинают вести себя непредвиденным образом.

## Сомневайтесь

Сомнение в надежности защиты должно присутствовать постоянно. Подвергните сомнению ваши предположения, ваши решения. Поставьте под вопрос ваши модели безопасности и модели угроз. Продолжайте анализировать деревья атак. Не доверяйте никому, особенно самому себе. Вы будете сильно удивлены тем, что вам удастся обнаружить.

## Обнаружение и реагирование

Обнаружение атак намного важнее, чем предотвращение. Я неоднократно уже упоминал о том, что полностью предотвратить нападения невозможно. Конечно, стремиться к совершенству нужно, но все, что нам известно относительно сложных систем, говорит о том, что невозможно выявить и устранить все уязвимые точки. Нападающие найдутся всегда, и их нужно ловить и наказывать.

Меня поражает, насколько часто производители средств защиты компьютерных систем забывают об этом. Вы никогда не увидите дверной замок с рекламным лозунгом: «С этим замком вы можете не бояться воров». Но те, кто продает средства защиты компьютеров, постоянно делают заявления подобного рода: «Брандмауэры исключают возможность проникновения в вашу внутреннюю сеть», «Процедура аутентификации не допустит нежелательных посетителей в вашу систему», «Шифрование не позволит посторонним прочитать содержимое ваших файлов». Все эти заявления безосновательны. Механизмы предотвращения нужны, но предотвращение — только часть решения проблемы безопасности и притом наиболее непрочная часть. Эффективная система безопасности также включает обнаружение нападения и реакцию на него.

В реальном мире люди понимают это. Банки не заявляют: «У нас есть надежное хранилище, так что мы не нуждаемся в системе сигнализации». Никто не увольняет ночную охрану музеев только потому, что в них есть замки на окнах и дверях. В лучшем из миров все, что вы выигрываете за счет применения защитных средств, — это время. В реальном мире средства, предотвращающие нападения, часто не используются вообще.

В некоторых случаях средства защиты — это все, на что вы можете положиться. Шифрование в качестве защитного средства должно предотвращать подслушивание переговоров. Но нет никакого способа обнаружить подслушивание, так что невозможно и никакое реагирование. Тем не менее чаще всего обнаружение и реагирование возможны.

И они обеспечивают большую надежность. Систему безопасности, защищающую ваш дом (дверной замок) легко обойти, если разбить кирпичом окно. Тогда почему большинство домов еще не ограблены? Почему нет публичных призывов использовать окна из поликарбоната? Потому что есть средства обнаружения и реагирования.

## Обнаруживайте нападения

Современное общество предотвращает преступления. Это — миф. Если Алиса захочет убить Боба, она сможет это сделать. Полиции не удастся остановить ее, если, конечно, она не полная идиотка. Они не смогут защитить каждого Боба. Боб должен позаботиться о своей безопасности сам. Он вправе нанять телохранителя, если это ему по карману, но это также ничего не гарантирует.

Обычно преступление обнаруживается уже после его совершения. «Офицер, мы только что откопали на задворках Стадиона Гигантов тело Боба, нашпигованное пулями. Я думаю, что здесь пахнет преступлением». Мы расследуем обнаруженные преступления, собираем свидетельства, которые помогут убедить присяжных в виновности обвиняемого. Предполагается, что весь процесс поиска и наказания преступника должен воздействовать на общество в целом и отбивать у других охоту следовать его примеру. Конечно, приговор выносится с целью наказать виновного, но реальная польза обществу состоит в предотвращении новых преступлений, то есть наказание производит профилактический эффект.

Хорошо, что вся эта сложная система более или менее работает, потому что предотвратить преступление намного труднее, чем обнаружить. Для цифрового мира

верно то же самое. Компании, выпускающие кредитные карточки, делают все возможное для предотвращения мошенничества, но главным образом они полагаются на средства обнаружения и, в чрезвычайных случаях, на судебное преследование. Сотовые телефоны могут быть клонированы, но механизмы обнаружения ограничивают финансовые потери.

Как борются с воровством в магазинах? Можно существенно ограничить возможность кражи, если прикрепить товары к прилавкам, поместить их в застекленные витрины или позади прилавка. Эти методы работают, но уменьшают продажи, потому что покупателю нравится брать товар в руки. Поэтому появилось множество технологий обнаружения воровства: ярлыки, прикрепленные к товарам, включают сигнализацию при попытке вынести их из магазина. (Существует другой интересный способ борьбы с кражами предметов одежды: к ним прикрепляются ярлыки, которые распыляют краску при попытке удалить их ненадлежащим образом. Этот способ называется *лишением выгоды*.)

В Интернете обнаружение может оказаться сложным делом. Недостаточно установить брандмауэр и этим ограничиться. Вы должны обнаруживать нападения на сеть. Это требует чтения, понимания и интерпретации огромного количества записей в контрольных журналах, которые делает брандмауэр, а также маршрутизаторы, серверы и другие сетевые устройства, так как вполне возможно, что некоторые атаки обойдут брандмауэр. Такие нападения всегда где-нибудь оставляют след.

Обнаружение должно быть своевременным. (Поздно реагировать на нападение, о котором вы узнаете из утренних газет.) Это означает, что требуется система контроля в реальном времени. Чем скорее вы обнаружите что-то подозрительное, тем скорее вы сможете отреагировать.

## Анализируйте нападения

Простого обнаружения недостаточно, необходимо понять — что это за нападение и что оно означает. Традиционно, военные разделяют этот процесс на четыре этапа.

**Обнаружение.** Осознание того, что произошло нападение. Три ключевых сервера вашей сети одновременно вышли из строя. Что это — нападение или просто проблемы программного обеспечения сети? Или, может быть, случайное совпадение? Если вы даже не знаете, нападение это или нет, вы не можете адекватно отреагировать.

**Локализация.** Определение точки, в которой произошло нападение. Даже если вы знаете, что сеть атакована, вы можете не иметь сведений о том, какие компьютеры или порты подвергаются нападению. Вы можете знать, что поломка серверов — результат нападения, но не иметь никакого понятия, как нападающему это удалось сделать и чем он занят в настоящее время.

Идентификация. Определение, кем является нападающий и откуда он работает. Это поможет составить представление о его сильных и слабых сторонах. Например, с нападающим из Соединенных Штатов можно бороться иными методами, нежели с нападающим из Молдавии. (Этот шаг более важен в традиционном военном процессе, чем в сетевой безопасности.)

**Оценка.** Понимание целей и мотивов атакующего, его стратегии и тактики, его возможностей и, желательно, его уязвимых мест. Эта информация имеет чрезвы¬

чайно важное значение для выбора средств реагирования. Реакция на шалости ребенка будет совершенно иной, нежели на действия промышленного шпиона. Ребенок, вероятно, сразу отключится от соединения, если вы хоть как-то прореагируете на него. Более упорного нападающего так легко остановить не удастся.

Каждый последующий шаг труднее, чем предыдущий, и требует более детальной информации. Часто ее анализ нуждается в участии квалифицированных специалистов, компьютер, предоставленный сам себе, рано или поздно не сможет справиться с этой задачей (хотя автоматическая программа может хорошо работать довольно долго).

На каждом шаге вы получаете все больше информации о ситуации. Чем больше информации вы получаете (и чем скорее), тем лучше вы вооружены. К сожалению, большинство сетевых администраторов никогда не знают, что они подверглись нападению, а если и знают, то не понимают, откуда оно исходит. Идентификацию и оценку особенно трудно осуществить в Интернете, где нападающему легко маскировать свое местоположение.

Скорость существенна. Чем быстрее вы проанализируете нападение, тем быстрее ответите на него.

## Ответьте на нападение

Поговорим о способах реагирования. Беспрерывно звонящая сигнализация, на которую никто не реагирует, ничем не лучше, чем полное ее отсутствие. Ответ — это то, для чего, собственно, и нужны средства обнаружения.

Иногда реагировать легко: если украден номер телефонный карточки, значит, его следует аннулировать. Иногда бывает сложнее: кто-то проник на сервер электронной торговли. Можно закрыть сервер, но потери составят 10 миллионов долларов в час. И что теперь?

Ответить непросто, хотя часто бывает, что люди принимают разумные решения за доли секунды. «Кто-то забрался на стену и приближается к застекленной крыше. Что нам сейчас делать?» Это во многом зависит от ситуации. Но вы не можете бездействовать. Можно просто прогнать его. Можно прогнать его и удостовериться, что он больше не вернется. Можно прогнать его, определить, как ему удалось туда попасть, и закрыть уязвимое место.

Пресечь нападение — это только половина дела. Не менее важно выследить и найти нападающего. Это бывает очень трудно в некоторых случаях: например, в Интернете нападающий может переходить с одного компьютера на другой, чтобы «замести следы». Полиция не может тратить много времени на расследование таких случаев до тех пор, пока не будут привлечены дополнительные людские или финансовые ресурсы, и я думаю, что частные компании могут оказать помощь правосудию, например, собирая досье на некоторых взломщиков.

Столкнувшись с судебным разбирательством, большинство людей из компьютерного мира, чуждых правовой сфере, обнаруживают, какая это помойная яма. Недостаточно идентифицировать нападающего, необходимо еще доказать это в суде. В Англии предпринимались попытки привлечь к ответственности обвиняемых в мошенничестве с кредитными картами. Как проходит рассмотрение дела в суде?

Защитник требует представить подробные сведения о средствах безопасности, используемых банком: описание технологии, записи в контрольных журналах; обо всем, что может прийти ему в голову. Представитель банка обращается к судье: «Мы не можем представить эти сведения, поскольку это причинит ущерб нашей безопасности». Судья прекращает дело. Система безопасности могла бы быть идеальным образцом обнаружения, она могла бы правильно указать преступника, но если она не переносит процесса раскрытия информации, значит, она недостаточно полезна.

Когда Джон Уокер предстал перед судом за шпионаж, Агентство национальной безопасности тщательно взвешивало, что лучше: раскрыть информацию о том, как Уокер взламывал механизмы шифрования, или сохранить в тайне истинные размеры причиненного им ущерба. Хорошие способы обнаружения должны выдерживать судебный процесс, включая перекрестные допросы с привлечением экспертов, не теряя при этом своей эффективности. Хорошие средства обнаружения и контроля обязаны делать записи в контрольных журналах, которые могут быть использованы в суде в качестве доказательств. Должна быть возможность демонстрировать эти записи, не раскрывая секретов безопасности (это называется *разделением знаний,* knowledge partitioning).

## Будьте бдительны

Бдительность нельзя терять никогда. Чтобы обнаружение и реагирование были эффективны, необходимо работать все время: 24 часа в сутки, 365 дней в году. Службы охраны работают круглосуточно. Компании, обслуживающие системы охранной сигнализации, работают без выходных. В компьютерном мире не может быть по-другому. Вы не можете повесить объявление при входе в вашу сеть: «Пожалуйста, взламывайте нашу сеть только с девяти часов утра до пяти вечера, с понедельника по пятницу, исключая отпускное время». У хакеров свои планы.

Нападения часто происходят в то время, когда к нему не готовы. Преступные хакерские нападения связаны с определенными периодами академического года. Все виды мошенничества в торговле — с банкоматами, кредитными карточками, — учащаются в Рождественские праздники, когда люди тратят много денег. Банковские системы взламываются чаще всего после полудня в пятницу, когда банки закрываются на выходные. В 1973 году арабы напали на Израиль во время празднования Йом Кипур, самого священного еврейского праздника. Если кто-нибудь готовит серьезное нападение, он выберет столь же неудобное время.

Бдительность означает своевременность обнаружения и реагирования. Обнаружить нападение, когда оно происходит, намного важнее, чем через неделю или даже час после случившегося. Гораздо лучше закрыть уязвимые места сразу после их обнаружения, а не в следующем месяце. Запоздалая реакция часто не лучше, чем ее полное отсутствие.

Бдительность также означает подготовленность. Необходимо ясно представлять, что делать в случае нападения. Когда в 2000 году Yahoo! подверглась атаке, приводящей к отказу в обслуживании, потребовалось целых три часа, чтобы возобновить работу. Частично это произошло потому, что Yahoo! никогда раньше не сталкивалась с нападениями этого типа. Когда все идет гладко, люди забывают,

как следует реагировать в исключительных ситуациях. Службы слежения и реагирования полезны только в том случае, если они регулярно обнаруживают нападения и реагируют на них.

## Контролируйте контролеров

В банковском деле давно известно, что хорошую безопасность можно обеспечить только с помощью нескольких уровней контроля. Управляющие контролируют кассиров. Бухгалтеры контролируют управляющих. Аудиторы также проверяют, все ли в порядке, но уже другими способами, и их проверки являются эффективным средством контроля деятельности бухгалтеров. Внешние аудиторы действуют как доверенные третьи лица, им платят за проведение проверки независимо от того, найдут они что-нибудь или нет. В игорном бизнесе также используется многоуровневый контроль. Крупье наблюдают за игроками, администраторы наблюдают за крупье, а специальные контролеры наблюдают за администраторами.

В банковском деле эта схема усилена за счет обязательных отпусков. Идея состоит в том, что если кто-нибудь другой займется вашими делами, он, возможно, обнаружит ваши незаконные действия. Некто Ллойд Бенджамин Льюис, младший сотрудник расчетного отдела крупного банка, более двух лет подготавливал крупную аферу. За все это время он ни разу не уходил в отпуск, не болел и не опаздывал на работу. Он должен был всегда быть на рабочем месте, иначе кто-нибудь мог обнаружить мошенничество.

Недостаточно иметь в штате хороших системных администраторов, которые знают все о компьютерах и проблемах безопасности сети, следят за работой системы и отвечают на нападения 24 часа в сутки. Кто-то должен наблюдать за ними. И не только потому, что они могут иметь злой умысел, хотя это и бывает. (Множество преступлений было совершено банковскими руководителями высшего звена, так как у них гораздо больше возможностей уйти от ответственности. Некий контролер игровых автоматов, убедившись, что они не находятся под наблюдением, пытался изменить работу их запоминающих устройств так, чтобы сорвать банк.) Причина в том, что люди есть люди и они совершают ошибки. Даже компьютеры могут ошибаться, и необходимо предусмотреть средства обнаружения и устранения ошибок.

## Устраните последствия нападения

В 2000 году была взломана французская смарт-карта. Ничего нельзя было сделать, кроме как прекратить всякие операции с ней. Подобные проблемы возникали и с проездными карточками в Нью-Йорке, канадской кредитной карточкой и со средствами защиты DVD. Если все ваше время уходит на обдумывание предупредительных мер, вы можете совершенно упустить из виду план действий в случае неудачи.

Предупредительные меры постоянно оказываются неэффективными. Устранение проблемы и выслеживание злоумышленников — очень важные этапы, но не менее важно привести все в порядок после нападения. При разработке систем следует ориентироваться на возможность их модернизации в процессе эксплуатации.

Также желательно предусмотреть специальные средства шифрования, протоколы и процедуры, которые будут использованы в чрезвычайной ситуации. Так можно сократить потери и быстрее восстановить работоспособность системы.

## Контратака

Борьба за безопасность бывает иногда довольно жестокой. Нападающим легче, они могут вести нечестную игру, использовать новую, неизвестную технику нападения. Они могут использовать методы, мысль о которых даже не приходит в голову защищающимся. Они не ограничены моделью угрозы, используемой в защите.

Преимущество нападающего состоит в том, что ему достаточно найти одну незначительную брешь в обороне. Защитник, с другой стороны, должен позаботиться об отражении любого возможного нападения. Он должен все предусмотреть, он не может позволить себе упустить что-либо из виду.

И защитники пребывают в смятении, допуская глупые ошибки. Они пишут код, содержащий множество изъянов. Они не устанавливают «заплаты» и не обновляют средства безопасности. Их вера в абсолютно безопасное изделие сродни теологической. Они не понимают, каковы реальные угрозы, и, соответственно, не принимают необходимых мер.

Время работает на нападающих. Системы должны работать изо дня в день. Нападающие могут ждать, выискивая тем временем слабые места системы, до того момента, когда защитники потеряют бдительность. Они могут менять стратегию и тактику в зависимости от ситуации.

Единственный выход — перейти в наступление.

Мы не боремся с преступностью, когда делаем наши банки невосприимчивыми к нападениям; с преступностью мы боремся, когда ловим преступников. К счастью, преступники довольно глупы. Хороший эксперт по безопасности имеет достаточно высокие доходы, которые и не снились хакеру, поэтому он может господствовать на поле боя.

Когда существовала угроза ядерного нападения СССР на Соединенные Штаты, предполагалось нанести ответный удар в случае ее реализации. Взаимное уничтожение столь же абсурдно, сколь нереально создание неуязвимой системы обороны, но это работало.

Детективное агентство Пинкертона было основано в 1852 году. В самом начале своей деятельности люди Пинкертона защищали от грабителей поезда на американском Западе. Они сразу поняли, что сопровождать каждый поезд — слишком дорогое удовольствие. Они также хорошо понимали, что ограбление — сложная операция: если вы собираетесь ограбить поезд, вам понадобится свой человек в управлении железной дороги, знающий график движения поездов, дюжина людей, лошади и т. д. Немногие способны справиться с такой задачей. Так что люди Пинкертона решили следовать непосредственно за грабителями. Для них было не так уж важно, платит ли железная дорога за расследование; они ловили преступников, чтобы защитить своих клиентов.

Люди Пинкертона были непреклонны. Если бы вы ограбили охраняемый ими поезд, они обязательно бы выследили вас. Это были серьезные ребята; проводи¬

лись целые вооруженные сражения против банды Hole in the Wall Gang, в которые вовлекались сотни людей Пинкертона. Имеется сцена в «Butch Cassidy and the Sundance Kid» («Грубиян Кэссиди и Малыш»), где гангстеры преследуются после грабежа поезда группой, которая не собиралась отступать. «Кто эти парни?» — спрашивает Грубиян у Малыша. Это были люди Пинкертона.

В киберпространстве нужны такие же хорошие контратаки. Сегодняшняя ситуация такова, что, если вам не грозит расплата, вас ничто не удержит от взлома. Вторжение в сеть — не игра, это — преступление. Кража денег с помощью взлома системы электронных платежей — преступление. Распространение в Интернете материала, защищенного авторским правом, — тоже преступление. И преступники должны преследоваться по закону. Судебное преследование, помимо наказания преступника, приносит дополнительную пользу. Во-первых, осужденный вряд ли пойдет на новое преступление. И во-вторых, вероятно, другие люди не станут рисковать, пытаясь сделать то же самое.

Это не призыв к «охоте на ведьм», чем уже пытались заниматься ФБР и некоторые другие организации в течение прошлого десятилетия. В восьмидесятые годы они знали немного о компьютерах и сетях, а также о компьютерных преступлениях. Опасность виделась повсюду. В 1989 году, когда NuPrometheus League похитила и опубликовала в Интернете исходный код ROM Macintosh, ФБР проводило проверку множества совершенно случайных людей. В 1990 году контрразведка провела облаву в штаб-квартире компании Steve Jackson Games, потому что компания работала над созданием ролевой игры (даже не компьютерной программы), в которой упоминались некие «киберпанки» и хакеры, а также потому, что одного из сотрудников, Лойда Бланкеншипа, подозревали в принадлежности к хакерской группировке «Легион Смерти» (Legion of Doom). В 1999 году Ассоциация контроля за копированием DVD попыталась заткнуть рот 500 веб-сайтам, чье преступление состояло только в публикации сообщения о взломе DVD-криптографии. А в 2000 году Microsoft потребовала, чтобы Slashdot удалил сообщения о закрытых расширениях к протоколу Kerberos.

Это не призыв к крайним мерам, к которым прибегали в девяностые годы. Дэвид Смит, создатель вируса Melissa, может получить от пяти до десяти лет тюрьмы[1]. Кевин Митник получил и отсидел почти пять лет, и ему запрещено пользоваться компьютером еще три года. (Все его навыки связаны с компьютерами, но ему запретили читать лекции по предмету. Чиновник, занимавшийся его досрочным освобождением, предлагал получить работу в Arby.) Кевину Поулсону был вынесен аналогичный приговор. Китайское правительство приговорило хакера к смерти за взлом банковского компьютера и похищение 87 000 долларов. (По правде говоря, в Китае казнят всех грабителей банков.) Помнится, на американском Западе в начале XIX века конокрадов часто отправляли на виселицу. Общество хотело этим показать, что не потерпит конокрадства. Правительства европейских стран продемонстрировали такое же отношение к террору в начале 70-х, когда террорис¬

[1] Несмотря на то что ущерб от «Мелиссы» был оценен в 80 миллионов долларов, правосудие оказалось гуманным, и Дэвид Смит получил всего 20 месяцев тюрьмы, учитывая его активное сотрудничество со следствием. Но дело «Мелиссы» живет и побеждает — придуманный хакером способ самораспространения используют множество новых вирусов. — *Примеч. ред.*

тов начали расстреливать на улицах. Тем самым они предельно ясно объявили: «Мы больше не играем в игры». Чрезмерно суровые приговоры, выносимые хакерам, связаны с панической реакцией общества на новую угрозу.

Это также не воззвание к тому, чтобы объявить вне закона любых исследователей или хакеров, практику «полного раскрытия» или отменить право оценивать надежность средств безопасности. Законы Соединенных Штатов запрещают «обратное проектирование» (reverse engineering) систем защиты от копирования. Индустрия развлечений вовсю лоббирует эти драконовские законы, чтобы с их помощью скрыть недостатки защиты своих продуктов. Ни в одной другой отрасли промышленности производители не пытаются запретить покупателю исследовать товар, чтобы понять, как он работает. Никакая другая промышленность не пытается помешать оценивать качество ее изделий по схеме «Отзывы потребителя». Глупо казнить гонца, принесшего дурные вести.

К чему я призываю, так это к ужесточению судебного преследования людей, вовлеченных в преступную деятельность, и к вынесению справедливых приговоров. Широко распространена такая точка зрения: «Если я сижу спокойно и не поднимаю шума, то никто не побеспокоит меня». Компании неохотно преследуют компьютерных преступников, потому что боятся мести. Действительность же такова, что до тех пор, пока мы не начнем преследовать преступников, они будут и далее распространять средства нападения и взламывать сети. Как только мы начнем наказывать их, хакерство перестанет быть столь привлекательным делом. Это не идеальное решение, так как хакеры, скорее всего, уйдут в подполье, но ситуация все же изменится. Применение в семидесятые годы жестких мер против терроризма имело два положительных эффекта: неисправимые террористы были вынуждены действовать с большей осторожностью, а другие просто сложили оружие.

## Управляйте риском

Абсолютной безопасности не существует, но не всегда это является проблемой. Только в Соединенных Штатах индустрия кредитных карточек теряет из-за мошенничества 10 миллиардов долларов в год, но ни Visa, ни MasterCard не собираются сворачивать бизнес. Ущерб от краж в магазинах Соединенных Штатов составляет от 10 до 26 миллиардов долларов в год, но при этом «утруска» (как это называется) редко становится причиной закрытия магазинов. Недавно мне понадобилось заверить документ у нотариуса: эта процедура использует самый слабый «протокол безопасности», который я видел когда-либо. Однако она прекрасно служит тем целям, ради которых была разработана.

После того как вы определили угрозы, вам предстоит сделать выбор: смириться с риском, постараться снизить его или застраховаться. Если нельзя полностью обезопасить себя, нужно управлять риском. Компании, выпускающие кредитные карточки, понимают это. Ущерб от мошенничества известен. Известно также, что потери, связанные с оплатой по телефону, приблизительно в пять раз больше потерь при непосредственных расчетах по сделке с использованием кредитной карточки и что потери от сделок в Интернете еще вдвое больше. (Многие издержки от фальшивок типа «карта не настоящая» несут торговцы, которые неактивно обращались

за помощью при предъявлении счета.) Они выдвигают альтернативы Интернету, такие как SET, именно ввиду повышенного риска.

Закрытая система, подобная этой, — исключение. Мое первичное опасение относительно киберпространства состоит в том, что люди не понимают опасностей и слишком верят в способность технологии устранить их. Киберпространство где-то копирует физический мир, а где-то разительно от него отличается (см. главу 2). И продукты безопасности не могут в одиночку решить ее проблемы.

Необходимо изменить отношение к средствам безопасности. Сейчас средства защиты выдаются производителями за профилактические меры, способные полностью исключить возможность нападения. Хорошее шифрование якобы предотвращает подслушивание. Хороший брандмауэр якобы предотвращает нападения на сеть. И так далее.

Принцип предотвращения угроз более подходит для формирования политики национальной безопасности, чем для организации защиты в коммерческом мире. Бизнес почти всегда связан с риском, поэтому в реальном мире большее внимание уделяется средствам обнаружения и реагирования. Веб-сайты не нуждаются в абсолютно не поддающихся взлому паролях, достаточно, если они будут защищать от нападений большую часть времени. Нет необходимости создавать абсолютно надежные смарт-карты, достаточно, чтобы механизмы обнаружения и реагирования успевали сработать вовремя. (На самом деле и это не так уж важно: в системе кредитных карточек, где вращаются многие миллиарды, используются очень слабо защищенные карты с магнитной полосой и управляемые продавцом терминалы.)

Как только вы начинаете думать о безопасности подобным образом, все остальное разваливается на кусочки. Если безопасность служит для ухода от угроз, тогда она должна оправдывать вложенные деньги. Если безопасность — управление рисками, это становится способом повысить доход. Если компания способна вычислить, как управлять опасностью, которая может возникнуть при подключении к сети их системы, то она сможет захватить большую часть рынка. Если компания кредитных карточек сумеет спрогнозировать, как управлять рисками некоторого класса клиентов, то она сможет продавать большее количество кредитных карточек. Бизнес — это всегда риск, и более востребованы те люди, которые лучше умеют управлять рисками.

Безопасность старее компьютера И промышленность, выпускающая защитные средства, думает о контрмерах как способах управлять риском. Различие огромно. Уход от угроз — как черное и белое: или вы избегаете угрозы, или нет. Управление риском неоднозначно: вы или принимаете риск, или уменьшаете его, или страхуетесь.

Безопасный компьютер — тот, который вы застраховали.

Я считаю, что будущее компьютерной безопасности — за страхованием. Можно застраховаться от чего угодно: от кражи или вандализма, от того, что какой-нибудь выродок расстреляет ваших сотрудников, и т. д. Почему в таком случае не застраховаться от нарушения цифровой безопасности?

Крупные страховые компании не упускают такую возможность. Они разрабатывают различные аспекты страхования рисков, связанных с компьютерной безопасностью: страхование внутренних сетей, страхование от нападений, приводящих к отказу в обслуживании, от подмены веб-сайта при взломе. Это трудная задача, поскольку никто не знает, с какими рисками придется иметь дело.

Излюбленная шутка страховщиков: к ним обращается некая компания, желающая застраховаться от какой-то неслыханной опасности. Между страховщиками и представителями компании происходит следующий диалог:

— Как велики возможные потери?
— Мы не знаем.
— Как может наступить страховой случай?
— Мы не знаем.
— Сколько стоит ваша компания?
— Столько-то.
— Это и будет страховой взнос, можете внести его.

Уже сейчас страховые компании предлагают страхование от хакерства, но я не думаю, что они достаточно хорошо представляют, на что идут. Большинство стратегий сложны и неуправляемы и содержат так много условий, что я не уверен, окупятся ли они когда-нибудь. Преимущество стандартизации процессов безопасности — возможность количественной оценки рисков. Если тысяча компаний используют схожие безопасности, то страховые фирмы могут выработать соответствующую политику. Так работает ADT. Компании покупают обслуживание не потому, что это делает их товарные склады более безопасными, а потому, что имеют возможность оценивать риски.

Со временем появятся два типа страхования сети. Первый тип очевиден: если кто-то вламывается в вашу сеть и причиняет вам ущерб, страховая компания компенсирует его. Но второй тип страхования даже более важен. Представьте, что кто-то проникает в вашу сеть и разоряет ваших клиентов, пользуется их конфиденциальной информацией и наносит непоправимый урон их репутации. Размер компенсации в этом случае может быть огромен. Страхование от угроз подобного типа оказывается рискованным.

Управление риском — будущее цифровой безопасности. Кто научится лучше управлять рисками, тот — победитель. Страхование — важная составляющая управления рисками. С помощью технических решений можно уменьшить риски до такого уровня, что страхование становится оправданным.

# Аутсорсинг процессов безопасности

Процессы безопасности — это способ уменьшения рисков. Средства защиты сети всегда имеют недостатки: необходимы некоторые процедуры, как для того, чтобы поймать нарушителей, которые используют в преступных целях эти недостатки, так и для исправления найденных ошибок. Если атакующий является посвященным лицом, то необходим процесс, который позволит обнаружить нападение, восстановить систему после нанесенных повреждений и преследовать преступника по закону. В сложных системах обычно присутствует множество уязвимых мест, и это может поставить под угрозу функционирование программ и оказание многих услуг (вспомните сотовые телефоны и DVD); тогда процесс необходим для восстановления системы и укрепления безопасности. Контрразведка — единственный способ быть в курсе того, что действительно происходит. Чтобы управлять остаточным риском, существует страхование.

Все эти процессы сложны, и, чтобы их обеспечить, нужны специалисты. И поскольку уже многие аспекты нашей жизни тем или иным образом связаны с киберпространством, растут требования к компьютерной безопасности, а следовательно, и спрос на специалистов. Единственно правильное решение — привлекать экспертов как можно чаще.

Представим центр, контролирующий безопасность большой сети. Требуется как минимум пять обученных аналитиков для замены одного, работающего в режиме 24 x 7 (часов и дней), а согласованные атаки могут потребовать внимания полдюжины аналитиков. Отдельной организации не выгодно нанимать стольких людей на постоянную работу, поскольку нападения происходят довольно редко, удобнее привлекать этих людей в необходимых случаях. Независимая служба способна обучать своих аналитиков теории и практике. Эта служба может активно испытывать меры противодействия, анализировать способы вторжения, разрабатывать новые способы отыскания уязвимых точек и быть в курсе новых методов взлома. Она также может наблюдать процессы, происходящие в различных зонах Интернета, а не только в сети одной организации.

Я думаю, что в ближайшее время возрастет число действующих в киберпространстве независимых служб безопасности, подобно тому как в физическом мире растет число частных служб охраны, таких как Allied Security и компаний по обслуживанию сигнализации, подобных ADT. Слишком много специальных знаний требуется для обеспечения безопасности киберпространства, и только сотрудники специализированной службы могут в полной мере обладать этими знаниями. Моя консультационная компания Counterpane Systems предлагает разработку и анализ систем безопасности с привлечением средств криптографии. Другие компании предлагают оценку риска, разработку политики безопасности, установку, испытание, обновление оборудования и т. д.

Такие компании оказывают и услуги по защите систем на предприятиях. Кто-то должен постоянно контролировать работу средств защиты и реагировать на все, что происходит. Они (один человек не может находиться на рабочем месте все 24 часа) должны разбираться в нападающих и их методах. Они должны поддерживать средства защиты, приспосабливаясь к постоянным изменениям в сетях и сетевых службах. Компании не могут самостоятельно справиться с этим. Они производят автомобили, продают книги или занимаются чем-либо еще, но не безопасностью сетей. Они привлекают независимые, специализирующиеся в этом компании для управления их сетями, а также для обеспечения безопасности. Конечно, всегда будут специализированные сети (банковские, сети сотовой связи, системы кредитных карточек), которые должны быть закрытыми, и всегда найдутся консультанты безопасности, специализирующиеся в этом. Этим занимается моя новая компания Counterpane Internet Security, Inc.

Это нормальная эволюция служб безопасности. Никто не нанимает свою собственную охрану; ее привлекают. Никто не нанимает своих собственных аудиторов; их привлекают. Даже такая обыденная вещь, как нарезка бумаги для документов, лучше будет сделана в привлеченной компании, специализирующейся в этом.

Кроме выгоды от профессионального подхода и эффективности, практика привлечения специализированных служб способствует тому, что аналитические данные по проблемам безопасности накапливаются. Привлекаемые компании могут

заниматься активным сбором информации о новых видах нападений и, возможно, даже вести разведывательную деятельность среди хакеров для предотвращения преступлений. Они сталкиваются с различными моделями нападения и могут их распознавать. И они способны защитить от атак своих многочисленных клиентов: могут обнаружить нападение в Нью Дели и защитить своих клиентов в Нью-Йорке.

В реальном мире организации привлекают службы безопасности. Никакая компания не нанимает собственных охранников, каждая обращается в охранную компанию. Банки платят за транспортировку автомобильным компаниям, обеспечивающим вооруженную охрану. Компании приглашают аудиторов для проверки правильности ведения дел. Обеспечение безопасности компьютера от обеспечения безопасности сети ничем не отличается. Оно в той же степени сложно и важно. Безопасность требует бдительности. В цифровом мире привлеченные услуги — единственное, что может обеспечить необходимую бдительность.

# Глава 25. Заключение

Марк Лойзокс, президент фирмы «Управляемые Разрушения» (Controlled Demolitions), взрывает старые здания для постройки новых. Жалуясь на «непрофессионализм» современных террористов, он утверждал в журнале Harpers Magazine (июль 1997): «Мы можем взорвать все мосты в Соединенных Штатах за пару дней... Я берусь заехать на грузовике на мост Верразано[1], взорвать мост и уехать обратно на велосипеде. И никто мне не помешает в этом».

Так как технологии усложняются, социологи приобретают более узкую специализацию. Почти для каждой области деятельности характерно, что люди, имеющие знание ее социальной инфраструктуры, также обладают знаниями по ее уничтожению.

Спросите любого врача, как кого-нибудь отравить, он сможет рассказать вам. Поинтересуйтесь у какого-нибудь аэродромного служащего, можно ли безнаказанно вывести из строя «Боинг-747», и не сомневайтесь, он знает, как это сделать. Теперь справьтесь у любого профессионала по безопасности Интернета, как «сломать» Интернет. Мне поведали примерно с полдюжины различных способов, и я знаю, что это не предел[2].

Перед обладателем таких знаний система беззащитна. Все, что для этого нужно, — это сочетание умений и моральных качеств. Иногда даже не требуется большого навыка. Тимоти Маквей[3] натворил вполне достаточно в правительственном здании Оклахома-Сити, даже несмотря на утилитарное использование взрывчатых веществ, что вызвало бы отвращение у такого профессионала, как Лойзокс.

---

[1] Verrazano Narrows Bridge — одно из «семи чудес» Нью-Йорка, наряду со статуей Свободы и башнями Всемирного торгового центра (не очень удачная аналогия). Соединяет остров (район) Стэйтен Айленд с Лонг Айлендом (Бруклином) и отличается грандиозностью: пролеты длиной в 4260 футов, одни из самых длинных в мире, поддерживаются опорами высотой с 70-этажное здание. — *Примеч. ред.*

[2] Вообще говоря, подвержены взлому не только отдельные узлы. Несмотря на расхожее мнение о неуязвимости глобальной распределенной сети «сломать» Интернет можно. Принципы иерархического пространства имен DNS подразумевают магистральные каналы и основные серверы для хранения имен доменов. Обычный террористический акт или военные действия могут вывести из строя узлы и перерубить каналы. Писали, что когда в 2000 году под Клином ковшом экскаватора был перерублен оптоволоконный кабель на Финляндию, доступ к зарубежным сайтам в южной части Центральной России оказался невозможен. — *Примеч. ред.*

[3] Сержант США, ветеран войны в Персидском Заливе. 19 апреля 1995 года взорвал здание федеральной администрации в Оклахоме (штат Индиана), что повлекло гибель 168 человек, в том числе 19 детей, ранено более 500. Будучи анархистом, придерживался крайне правых взглядов, протестовал против контроля за оружием. Бомба (мазут, нитроглицерин и аммиачная селитра) была помещена в автомобиль и оснащена дистанционным взрывателем. Маквей и его сообщник Терри Николс также подозревались в ряде ограблений банков. В 1997 году Маквей был приговорен к смертной казни и казнен 11 июня 2001 года. Он съел заказанный накануне предсмертный завтрак — около килограмма шоколадного мороженого — и в качестве последнего слова прочел поэму английского поэта Уильяма Эрнеста Хенли (XIX век). «Не важно, сколь узки ворота, Сколь моя кара тяжела, Хозяин я своей судьбы, Своей души я полководец». Маквей не раскаивался, говорил, что и сейчас бы сделал все так же, как в 1995 году. «Если я попаду в ад, у меня будет большая компания». После 1963 года это была первая федеральная казнь (не под юрисдикцией штатов). Правительство организовало закрытую телетрансляцию казни (ввод инъекции) для родственников погибших и людей, выживших во время взрыва, запрещенную для записи. — *Примеч. ред.*

Доктор Гарольд Шипман убил 150 своих пациентов, используя такие безыскусные методы, как впрыскивание морфия.

На первый взгляд киберпространство никак не отличается от любой другой инфраструктуры нашего общества, оно настолько же непрочно и уязвимо. Но, как я говорил в главе 2, характер нападений различен. Маквей должен был приобрести знание, прийти на частную ферму и попрактиковаться во взрывах, арендовать грузовик, нагрузить его взрывчаткой, приехать к месту теракта, зажечь запал и уйти. Доктор Шипман ничего бы не добился без медицинской практики, не обзаведясь пациентами. Наш гипотетический борец с «Боингами» должен уметь обслуживать самолеты. Они все обязаны были подобраться близко к цели, подвергнуть себя риску, осуществить задуманное, совершить ошибки. И они должны были знать, что делают.

Или подумайте о гонке ядерных вооружений. Еще не было никакой крупномасштабной эскалации вооружений, а знания по производству ядерных бомб стали достоянием публики. Почему так случилось? Потому что эти знания не являлись опасными; осуществить громоздкие технические программы, вложив в них гигантские средства, могли себе позволить только единичные государства.

Киберпространства бывают разные. Вы можете находиться в другом месте, далеко от узла, на который вы нападаете. Вы можете не иметь никаких навыков и вообще ничего, кроме программы, которую вы загрузили с первого попавшегося веб-сайта. И вы даже не подвергаетесь риску. Порядочный хакер распространит сведения о найденной уязвимости в Интернете, а преступник напишет программу-имитатор нападения, которая продемонстрирует уязвимость, и затем любой неумеха без комплексов сможет воспользоваться ею для нападения. Например, как червем филиппинского студента, инфицировавшим десять миллионов компьютеров, что повлекло 10 миллиардов долларов ущерба. Или, возможно, в некотором царстве-государстве с неразвитой правовой системой приютилась веб-страница, которая содержит приложение Java: «Щелкните здесь, чтобы „сломать" Интернет». Не очень привлекательно, так ведь?

В девятнадцатом столетии французский социолог Эмиль Дурхейм постулировал, что анонимность делает людей преступниками. Вы можете дополнить его доводы, приведя в пример психологию современного хакера: невозможность ни с кем связаться, анонимность действий и отсутствие последствий за содеянное приводят некоторых людей к антиобщественным поступкам. Миазмы, вредные влияния виртуального пространства — гарантия этого.

Технология — не панацея, она не смогла помешать Маквею или Шипману. Оба были схвачены с помощью процессов безопасности: обнаружения и реагирования. В случае Шипмана обнаружение и реагирование абсурдно не соответствовали, и он избежал должной за свою бойню кары на десятки лет заточения. Правоохранение выявило случившееся, расследование установило, кто это сделал, а закон не наказал виновного по заслугам[1].

---

[1] Суд вынес вердикт, что Шипман унес жизни 15 пациентов во время своей практики в Гайде, вспрыскивая им морфий, и в январе 2000 года 55-летний английский «Доктор Смерть» был заключен в тюрьму Frankland в County Durham. Известно, что за 24 года своей практики в Манчестере с именем Гарольда Шипмана было связано 297 смертей, но доказать виновность Шипмана в этих случаях было уже невозможно. Дознание выявило причастность «ангела смерти» еще к 25 насильственным смертям, но суд не смог предъявить обвинение. — *Примеч. ред.*

Для социальных проблем нет технических решений. Законы жизненно важны для безопасности.

Если кто-нибудь изобретет дверь, которую невозможно вскрыть, такой же оконный замок или совершенную систему сигнализации, общество не изменится и не скажет: «Нам не нужны полиция или устаревшие законы о взломе и вторжении». Если история преступной деятельности что-нибудь и показала, так это ограниченность технологий. Мы нуждаемся в охране для наблюдения за продуктами и в полиции для расследования преступлений. Нам требуются законы, чтобы преследовать по суду мошенников в сфере электронной торговли, нарушителей компьютерной безопасности и людей, которые создают средства, облегчающие эти преступления. Мы можем внедрить наилучшие технологии для предотвращения преступлений, которые совершаются в первый раз, или для их обнаружения уже после свершившегося факта. Но мы все равно окажемся перед необходимостью полагаться на охрану для поимки преступника и на судебную систему для вынесения приговора. Мы можем сделать все возможное, чтобы помешать проведению маркетинговых исследований фирмы, включающих нелегальный сбор информации на людей, но мы также нуждаемся в законах, чтобы преследовать по суду эти нарушения.

Короче говоря, мы должны заручиться гарантией, что люди будут подвергать себя опасности при совершении преступлений в киберпространстве.

Мы также должны учиться на своих ошибках.

Когда самолет DC-10 терпит крушение, что-либо узнать об этом легко. Есть расследования и отчеты, и в конечном счете люди учатся на таких несчастных случаях. Вы можете подключиться к Air Safety Reporting System и получить детальные описания десятков тысяч аварий и несчастных случаев, начиная с 1975 года.

Крах безопасности происходит по разным причинам, но обычно не из-за шаровой молнии или внезапного удара. Наиболее успешные нападения (на банки, корпорации и правительства) не упоминаются в средствах массовой информации. Некоторые из них проходят незамеченными даже самими жертвами. Мы знаем все относительно MD-80[1], вплоть до металлургии винтового домкрата, и лишь немного о том, как нападавшие крали номера кредитных карточек с веб-узлов. Это подобно Аэрофлоту Советского Союза: официально никогда не происходило никаких аварий, но каждый знал, что иногда самолеты не достигают пункта назначения по загадочным обстоятельствам.

Предание гласности не оценивается по достоинству. Когда в 1995 году Ситибанк потерял 12 миллионов долларов из-за российского хакера, банк объявил о факте внедрения и уверил клиентов, что предпринял новые, более серьезные меры безопасности для предотвращения таких нападений в будущем. Даже в этом случае миллионы долларов были отозваны людьми, которые полагали, что их счета стали уязвимы сразу после объявления Ситибанка. В конечном счете, Ситибанк возместил убытки, но получил ясный и однозначный урок: не разглашать.

---

[1] MD-80 — реактивный самолет, используемый для полетов на средние расстояния, в основном на внутренних авиалиниях (дальность полета со 155 пассажирами 3800 км). С 1977 года до перехода к новой системе обозначений в компании «Макдонелл-Дуглас» в 1983 году назывался DC-9 Super. В 1992 году количество заказов на самолеты серии перевалило за 1000 экземпляров. — *Примеч. ред.*

Мы обязаны предавать гласности нападения. Должны открыто анализировать, почему системы выходят из строя. Мы должны разделить информацию о нарушениях безопасности на причины, слабые места, результаты и методики. Секретность только на руку нападающим.

Недальновидная политика тех, кто стремится запретить перепроектирование, только ухудшает положение. Почему люди, покупающие программное обеспечение, не должны знать, как оно работает, в отличие от покупателей, например, автомобилей? Почему программное обеспечение должно быть освобождено от «Отзывов потребителя» (способа анализа и испытания)? Опять тайна только помогает нападающим.

И мы нуждаемся в реальной ответственности поставщиков за продукты. Это очевидно: поставщики не будут предоставлять безопасное программное обеспечение, пока это не в их интересах.

Сочетание безответственности и невозможности перепроектирования особенно разрушительно. Если исследователям запрещен анализ безопасности продукта, имеет ли смысл ограждать поставщиков от ответственности? И если поставщики не несут никакой ответственности за некачественные продукты, то как указание на недостатки может быть незаконным?

Всюду в этой книге я доказывал, что технологии безопасности имеют ограничения. Я не хочу сказать, что они бесполезны. Такие контрмеры, как криптография, защита от несанкционированного вмешательства и обнаружение вторжения, делают систему более безопасной. Технологии задерживают незрелые сценарии и случайных нападающих, которые действительно не ведают, что творят. Но эти технологии подобны рентгеновской установке и датчикам металла в аэропортах: они не делают ничего, чтобы остановить профессионалов, но препятствуют любителям.

Средний человек не способен отличить хорошую безопасность от плохой. Она так же работает и столько же стоит. (Плохая безопасность могла бы даже лучше смотреться и меньше стоить; компания, которая не слишком волнуется о безопасности, может уделять больше ресурсов для изготовления модных «примочек».) Реклама и специальная литература в обоих случаях одинаковы. Разницы не заметить, не заглянув внутрь исходного кода или «железа». И к тому же вы должны быть специалистом. Средний человек все еще не может отличить продукт высокого качества от подделки.

Мир заполнен вещами, угрожающими общественной безопасности, суть которых находится вне понимания среднего человека. Люди не могут отличить безопасную авиалинию от опасной, но 1,6 миллиона человек в Соединенных Штатах летают каждый день. Люди не всегда способны отличить качественное лекарство от бесполезного, и все же объем американского фармацевтического рынка составляет 60 миллиардов долларов в год. Люди пользуются «заезженными» каботажными судами, доверяют свои деньги не тем, кому следует, и едят обработанное мясо — не проявляя реального беспокойства по поводу своей безопасности.

В коммерции все то же самое. Когда в последний раз вы лично проверяли точность насосов бензоколонки, или счетчик такси, или информацию о весе и объеме пищевых продуктов в пакетах? Когда в последний раз вы вошли в офис здания и потребовали показать свидетельство последнего осмотра лифта? Или проверяли лицензию фармацевта?

Мы часто полагаемся на правительство для защиты потребителя в областях, где большинство людей не имеют навыков или знаний, чтобы оценить риски должным образом и сделать разумный выбор покупки. Федеральное авиационное агентство (FAA) регулирует безопасность авиатранспорта; Министерство транспорта (DOT) отвечает за безопасность автомобилей. Администрации штатов регулируют веса и меры в торговле. Вы не можете ожидать от семейства, движущегося в направлении к Диснейленду, принятия разумного решения относительно того, является ли их самолет безопасным для полета или арендованный автомобиль безопасным в движении. Мы не ждем, что балкон второго этажа гостиницы упадет на портик ниже. Вы можете спорить о том, действительно ли правительство хорошо справляется со своей ролью (так как избиратели не понимают, как оценивать риски, то они и не вознаграждают правительство за хорошую оценку угрозы), но все же благоразумнее оставить эту роль именно ему.

Но если Управлению по контролю за продуктами и лекарствами США (FDA) доверить роль управления Интернетом, то в результате государственного регулирования из Интернета, вероятно, будет вычищено все, что делает его Сетью, сама его сущность. Регулирование часто выбирает неправильные решения (сколько денег было потрачено, чтобы оснастить посадочные места в самолетах спасательными поясами, и какое количество людей реально воспользовались ими при крушении?), и оно медлительно. FDA потребовалось три с половиной года на утверждение Interleukin-2[1], что соответствует вечности в мире Интернета. С другой стороны, неспешность FDA бывает иногда на пользу: из-за нее Соединенные Штаты не имели национальной катастрофы из-за «Талиломида» (Thalidomide)[2] в масштабе Британии. И благодаря ей же продажа «Лаэтрила» (Laetrile)[3] на американском рынке так и не была санкционирована.

---

[1] «Интерлейкин-2» («Ронколейкин») — иммунное средство нового поколения (клеточный рецептор, иммуномодулятор). Препарат используют в лечении сепсиса, инфекционных (туберкулез, гепатит С и др.) и онкологических заболевай : - *Примеч. ред.*

[2] Иммунодепрессант, оказавший гигантское влияние на законодательную систему Европы в области фармацевтики — фактически сформировавший ее: законы и контролирующие органы. Первая пробная партия была выпущена в 1954 году, но массовое применение пришлось на 1957-1961 годы. Тогда же оказалось, что прием талидомида во время беременности (в качестве успокаивающего или средства от бессонницы) приводит к врожденным уродствам детей, у ряда взрослых пациентов наблюдался вызванный им полиневрит. Наибольшее количество жертв пришлось на Западную Германию и Англию. В 1962 году препарат был изъят во всех странах. Против изготовителя и продавца талидомида в Англии, компании «Дистиллерс», возбудили процесс родители новорожденных с пороками развития. Реклама талидомида особо подчеркивала его нетоксичность, хотя этот вывод был сделан из-за отсутствия острой токсичности при приеме разовой дозы препарата. Кутерьма вокруг «Дистиллерс» и талидомида в Англии завершилась в 1981 году принятием «Закона о неуважении к суду», заменившего прецедентные нормы. Теперь, спустя 40 лет, талиломид снова возвращается. Выяснилось его противоопухолевое действие и многие надеются, что свидетельства его эффективности при лечении рака, СПИДа, проказы и ревматоидного артрита станут достаточным основанием для реабилитации лекарства. — *Примеч. ред.*

[3] Цианидсодержащее соединение, получаемое из персиковых косточек. Также называется витамином В17. «Лаэтрил» не обладает биологической активностью витамина, имея химически сходный состав. Биохимики давно признали несостоятельность снадобья, но оно по-прежнему продолжает применяться в альтернативной медицине для «лечения» рака и продается в аптеках. В частности, и в богатой шаманами, целителями и колдунами России. — *Примеч. ред.*

Или представим, что Underwriters Laboratory отвечает за безопасность кибер¬пространства. Это частная лаборатория, которая испытывает и сертифицирует электрическое оборудование. (Они также оценивают надежность сейфов.) «Отзывы потребителей» предоставляют подобный сервис для других продуктов. Частная компания может обеспечить компьютерную и сетевую безопасность, но ценой огромных затрат. И поэтому правительство отворачивается от такой модели, а новые законы в Соединенных Штатах не признают законность оценки безопасности продуктов частыми компаниями и лицами.

Другой пример. Medical Doctors и Registered Nurses лицензированы. Инженеры, имеющие сертификаты, могут помещать буквы PE (зарегистрированный инженер) после своего имени. Но свидетельства значимы на местах, а Интернет глобален. И все еще нет никакой гарантии.

Все эти модели не выводят нас из затруднительного положения. Мы нуждаемся в технологических решениях, но они пока не совершенны. Мы нуждаемся в специалистах для управления этими технологическими решениями, но имеющихся специалистов мало. Мы нуждаемся в сильных законах, чтобы преследовать по суду преступников, и готовы следовать установленному порядку, но большинство атакованных компаний не хотят обнародовать случившееся.

В главе 24 я доказывал, что способов обеспечить безопасность при ограничениях технологии, кроме как задействовать процессы безопасности, нет. И что эти процессы неразумно осуществлять силами организации, правильнее привлечь профессионалов безопасности киберпространства. Это представляется единственным выходом из обрисованного выше положения.

Предположим, что решение привлечь стороннюю организацию вас устраивает.

В моей первой книге «Прикладная криптография» я писал: «Шифрование слишком важно, чтобы оставить его исключительно правительству». Я все еще верю этому, но в более общем смысле. Безопасность слишком значительна, чтобы разрешить ею заниматься любой организации. Доверие нельзя доверить случаю.

Доверие индивидуально. Один человек всецело доверяет правительству, в то время как другой может не доверять ему вообще. Различные люди могут доверять различным правительствам. Некоторые люди доверяют корпорациям, но не правительству. Невозможно проектировать систему безопасности (продукт или процесс), которая будет лишена доверия; даже человек, пишущий собственное программное обеспечение безопасности, должен вверить его компилятору и компьютеру.

К сожалению, большинство организаций не понимают, кому они доверяют. Кто-то может вслепую положиться на какую-то особую компанию без особой на то причины. (Слепая вера некоторых людей заставляет их иметь особенную операционную систему, брандмауэр или алгоритм шифрования.) Иная администрация безоговорочно доверяет своим служащим. (Я слышал, что некоторые оценивают безопасность не по тому, сколько истратили на брандмауэр, а по тому, сколько стоит системный администратор.)

Из того, что безопасность по своей сути обязана ограничивать риски, вытекает, что организации должны доверять объектам, которые ограничивают их риск. Это значит, что объекты имеют гарантию. Доверенные объекты обладают такими свойствами, как проверенный послужной список, хорошая репутация, независимые

сертификаты и аудиторы. Каждое по отдельности не считается доказательством, но все вместе являются основанием для доверия.

Выбор заключается не в том, чтобы доверять организации, а в том, какой организации доверять. Для меня отдел MIS (управленческой информационной структуры) компании, в которой я работаю, вероятно, заслуживает меньшего доверия, чем независимая привлеченная организация, которая всерьез заботится о безопасности.

Безопасность — это не продукт, а процесс. Вы не можете ее просто добавить к системе уже после нападения. Жизненно важно понять реальные угрозы для системы, спроектировать политику безопасности, соразмерную серьезности угроз, и реализовать соответствующие контрмеры. Помните, что не требуются идеальные решения, но также недопустимы системы, которые можно полностью разрушить. Хорошие процессы безопасности также существенны, они помогают продукту работать.

Благоразумнее готовиться к наихудшему. Нападения и нападающие со временем только совершенствуются, а системы, установленные сегодня, могли быть к месту на 20 лет раньше. Реальным уроком Y2K[1] стала замена устаревшего кода компьютера. Мы все еще подвержены ошибкам, допущенным в аналоговых телефонных системах десятилетия назад и в цифровых сотовых системах годы назад. Мы все еще работаем с опасным Интернетом и ненадежными системами защиты пароля.

Мы также до сих пор еще имеем дело с ненадежными дверными замками, уязвимыми финансовыми системами и несовершенной юридической системой. Но все же ничто из перечисленного не является причиной крушения цивилизации и маловероятно, что станет. И мы не добьемся должной цифровой безопасности, пока не сосредоточим внимание на процессах безопасности, а не на технологиях.

[1] Проблема-2000. — *Примеч. ред.*

# Послесловие

Эта книга постоянно менялась. Я написал две трети книги и осознал, что мне нечем обнадежить читателя. Казалось, возможности технологий безопасности исчерпаны. Мне пришлось отложить рукопись более чем на год, было слишком тягостно работать над ней.

В начале 1999 года я разочаровался в своей консультационной деятельности. Компания Counterpane Systems давала консультации по криптографии и компьютерной безопасности на протяжении нескольких лет, и бизнес быстро развивался. В основном наша работа заключалась в проектировании и анализе. Например, заказчик приходил к нам со своей проблемой, и мы разрабатывали систему, которая была способна противостоять определенным угрозам. Или же заказчик приходил к нам с уже разработанной системой, которая должна была противостоять целому списку угроз, а мы искали просчеты в решении и устраняли их. Мы могли работать до изнеможения. Единственная проблема состояла в том, что наши изящные разработки не выдерживали столкновения с реальным миром. Хорошая криптография постоянно оказывалась неэффективной из-за плохого исполнения. Прошедшие тщательные испытания средства защиты отказывали из-за ошибок, допущенных людьми. Мы делали все, что могли, но системы все же оставались ненадежными.

От криптографии я обратился к безопасности и представлял себе проблему примерно так же, как и военные. Большинство публикаций по вопросам безопасности были проникнуты той же идеей, которую можно кратко сформулировать так: профилактические меры могут обеспечить безопасность.

Для чего используется шифрование? Существует угроза прослушивания, и шифрование должно помешать этому. Представьте себе, что Алиса общается с Бобом, а Ева подслушивает их. Единственный способ помешать Еве узнать, о чем говорят Алиса и Боб, — использовать такое профилактическое средство защиты, как шифрование. В этом случае нет ни обнаружения, ни реагирования. Нет также возможности управлять риском. Поэтому следует заранее отвести угрозу.

В течение нескольких десятилетий мы использовали этот подход к безопасности. Мы рисовали различные схемы. Мы классифицировали различных нападающих и определяли, на что они способны. Мы использовали профилактические меры, такие как шифрование и контроль доступа. Если мы можем отвести угрозы, то мы победили. Если нет — мы пропали.

Вообразите мое удивление, когда я узнал, что в реальном мире такой подход не работает. В апреле 1999 года я прозрел: безопасность связана с управлением рисками, процессы безопасности имеют первостепенное значение, а обнаружение и реагирование — реальные способы улучшить безопасность, и все это могут обеспечить лишь привлеченные независимые компании. Внезапно все стало на свои мес¬

та. Так что я переписал книгу и преобразовал свою компанию Counterpane Systems в Counterpane Internet Security, Inc. Теперь мы предоставляем услуги по контролю безопасности сетей (обнаружение и реагирование).

Раньше, когда использовалась связь только по радио или телеграфу, такой подход не имел бы смысла. Обнаружение и реагирование были невозможны. Сегодняшний электронный мир сложнее. Нападающий не просто подключается к линии связи. Он взламывает брандмауэр. Он пытается украсть деньги, используя подделанную смарт-карту. Он хозяйничает в сети. Сегодняшний виртуальный мир больше похож на физический со всеми его возможностями.

Но это не ситуация «все или ничего». Раньше, если уж Ева умела подслушивать, то она могла бы подслушать всех. А если бы у нее не было такой возможности, то ей не удалось бы подслушать никого. Сегодня все иначе. Можно украсть деньги, но не слишком много. Пират-одиночка может сделать несколько копий DVD, но не десятки тысяч. Нападающий может проникнуть в сеть и поковыряться в ней минут десять, после чего он будет обнаружен и выдворен.

Реальный мир полон опасностей. Это не значит, что их нужно избегать, невозможно делать деньги, ничем не рискуя. В выигрыше оказывается не та компания, которая лучше всех отводит угрозы, а та, которая лучше всех управляет рисками. (Вспомните систему кредитных карт.)

В Counterpane Internet Security мы считаем, что одни только технические средства не могут защитить от нападений, осуществляемых человеком, поэтому основная наша деятельность заключается в предоставлении услуг квалифицированных специалистов по безопасности. Мы собираем информацию с разных устройств в сетях клиентов и тщательно выискиваем следы нападений. Все сколько-нибудь подозрительное исследуют наши аналитики, которые знают все о нападениях, могут определить, действительно ли происходит нападение, и знают, как на него реагировать.

Я понял, что проблема не в технологиях, а в их использовании. В своей фирме мы используем симбиоз человеческих способностей и возможностей машины. Люди — наиболее уязвимое звено любой системы безопасности, в том числе компьютерной.

И если читателю покажется, что эта книга написана в рекламных целях, то он будет отчасти прав. И книга, и новая компания появились на свет благодаря открытию того, что самую надежную защиту можно обеспечить только с помощью опытных специалистов, занимающихся обнаружением и реагированием. Эта книга отражает ход моих мыслей, связанных с преобразованием нашей компании, и поясняет, чем мы занимаемся.

Вы можете узнать больше о нас на сайте www.counterpane.com.

Спасибо за внимание.

# Источники

Многие идеи этой книги были почерпнуты автором в работах других людей. Я намеренно не прерывал повествование сносками или цитатами. В конце книги я решил поместить список некоторых наиболее полезных источников.

Некоторые специалисты открыто осуждают Интернет за то, что не поддается систематизации как архив, поскольку URL постоянно меняются. Так что вы можете использовать этот список, чтобы попытаться подтвердить или опровергнуть этот взгляд.

Работы Росса Андерсона (Ross Anderson) всегда интересны и заслуживают внимания. Его веб-сайт: www.cl.cam.ac.uk/users/rja14/. Ищите его книгу *Security Engineering: A Comprehensive Guide to Building Dependable Distributed Systems* (John Wiley & Sons, 2000).

Дороти Деннинг (Dorothy Denning) писала о криптографии, компьютерной безопасности и защите баз данных, а позже об информационной войне. Я использовал ее самую последнюю книгу *Information Warfare and Security* (Addison-Wesley, 1999), а также ее классическую работу *Cryptography and Data Security* (Addison-Wesley, 1982).

Работы и речи Вита Диффи (Whit Diffie) также повлияли на мои размышления. Я рекомендую книгу, написанную им в соавторстве со Сьюзен Ландау (Susan Landau) *Privacy on the Line* (MIT Press, 1998).

Карл Эллисон (Carl Ellison) продолжает писать эссе и статьи об инфраструктуре открытого ключа. Многие его работы можно найти на веб-сайте: world.std.com / ~cme/.

От Эда Фелтона (Ed Felton) я узнал много нового о ненадежности модульного программного обеспечения и о средствах безопасности Java. Именно он показал мне однажды рисунки, воспроизведенные в этой книге под номерами 10.1 и 10.2.

Речи Дэна Гира (Dan Geer) были столь же содержательны.

Превосходная работа Дитера Голлманна (Dieter Gollmann) *Computer Security* (John Wiley & Sons, 1999) содержит много полезных сведений.

Классическая книга Дэвида Кана (David Kahn) *The Codebreakers* изобилует интересными историческими фактами из области криптографии.

Стюарт МакКлур (Stuart McClure), Джоел Скрамбрей (Joel Scrambray) и Джордж Курц (George Kurtz) написали книгу *Hacking Exposed (Osborne/McGraw-Hill,* 1999), которую я настоятельно рекомендую. Я написал предисловие ко второму ее изданию, которое уже должно выйти к моменту публикации этой книги.

Гэри Макграу (Gary McGraw) подробно описал разработку надежного программного обеспечения, а также все плюсы и минусы открытого исходного кода. Я использовал его книгу *SecuringJava* (John Wiley & Sons, 1999), написанную в соавторстве с Эдом Фелтоном (Ed Felton).

Наблюдения Питера Неймана (Peter Neumann) настолько глубоки и очевидны, что я часто забываю, что не всегда верил ему. Его колонка *Inside Risks* на последней странице журнала Communications of the ACM всегда интересна. Я также настоятельно рекомендую его книгу *Computer-Related Risks* (Addison-Wesley, 1995).

Эссе, речи и застольные шутки Маркуса Ранума (Marcus Ranum) долго были для меня источником вдохновения и здравого смысла. Я настоятельно рекомендую прочитать все, что он написал. Его веб-сайт: http://pubweb.nfi-.net/~mjr/.

Эви Рувин (Avi Ruvin), Дэн Гир (Dan Geer) и Маркус Ранум (Marcus Ranum) совместно написали книгу *Web Security Sourcebook* (John Wiley & Sons, 1997), которую я тоже рекомендую.

Книга Винна Швартау (Winn Schwartau) *Time Based Security* (Interpact Press, 1999) содержит схожие с моими идеи о важности обнаружения и реагирования.

Диомидис Спинеллис (Diomidis Spinellis) приводит данные о сложности операционных систем и языков программирования в своей статье *Software Reliability: Modern Challenges* (in G. I. Schueller and P. Kafka, editors, *Proceedings ESREL'99—The Tenth European Conference on Safety and Reliability,* pages 589-592, Munich-Garching Germany, September 1999).

Размышления Ричарда Тиема о хакерстве и эпистемологии Интернета были для меня источником вдохновения. Вы можете найти его работы на сайте: www.thiemeworks.com.

Сотни эссе и статей о компьютерной безопасности публикуются каждый год. Если бы я все их прочитал, то, несомненно, все мысли, идеи, размышления, нюансы и умные остроты, собранные там, попали бы в эту книгу. Я приношу свои извинения за то, что не могу выразить благодарность каждому, кто заслуживает этого.

# Алфавитный указатель

## Б



## В



## Г



## Д



## З



## И

## К



## Л



## М



## Н



## О

## П



## Р



## С

## Т



## У



## Ф



## Х



## Ц



## Ч



## Ш



## Э

*Брюс Шнайер*

**Секреты и ложь. Безопасность данных в цифровом мире**

*Перевела с английского Н. Дубнова*


| | |
|---|---|
| Главный редактор | *Е. Строганова* |
| Заведующий редакцией | *И. Корнеев* |
| Руководитель проекта | *А. Васильев* |
| Научный редактор | *С. Беззатеев* |
| Литературные редакторы | *Е. Васильев* |
| Художник | *Н. Биржаков* |
| Иллюстрации | *В. Шендерова* |
| Корректоры | *С. Беляева, И. Тимофеева* |
| Верстка | *А. Келле-Пелле* |

Лицензия ИД № 05784 от 07.09.01.
Подписано в печать 20.07.03. Формат 70X100/16. Усл. п. л. 29,67.
Тираж 4000 экз. Заказ № 276.
ООО «Питер Принт». 196105, Санкт-Петербург, ул. Благодатная, д. 67в.
Налоговая льгота - общероссийский классификатор продукции
ОК 005-93, том 2; 953005 - литература учебная.
Отпечатано с готовых диапозитивов в ФГУП «Печатный двор» им. А. М. Горького
Министерства РФ по делам печати, телерадиовещания и средств массовых коммуникаций.
197110, Санкт-Петербург, Чкаловский пр., 15.

Б. ШНАЙЕР

# СЕКРЕТЫ И ЛОЖЬ

## БЕЗОПАСНОСТЬ ДАННЫХ В ЦИФРОВОМ МИРЕ

**КНИГИ, КОТОРЫЕ НЕ СТАРЕЮТ!**

КЛАССИКА COMPUTER SCIENCE

**Брюс Шнайер** — президент криптографической компании Counterpane Systems (США), член совета директоров Международной ассоциации криптологических исследований (IACR) и член консультативного совета Информационного центра электронной приватности (EPIC). Шнайер известен как автор нескольких бестселлеров и признанный эксперт в области прикладной криптографии и компьютерной безопасности. Его книги давно стали настольными энциклопедиями для множества людей, интересующихся вопросами защиты информации.

В своей новой книге Брюс Шнайер развеивает миф о том, что информация в цифровом мире может быть абсолютно конфиденциальной. Он проповедует собственный подход к защите информации. Безопасность данных для него — не только систематизация, обнаружение и отражение угроз, главное — управление рисками, своевременные превентивные меры для снижения риска угроз, каждодневная кропотливая работа по системному обеспечению безопасности предприятия.

> «...Наконец-то я могу предложить решения для обозначенного круга проблем, показать путь из тьмы, дать надежду на будущее компьютерной безопасности...»
>
> **Брюс Шнайер**

ISBN 5-318-00193-9

9 785318 001932



ПИТЕР®
WWW.PITER.COM

WILEY